IEEE STANDARDS ASSOCIATION

IEEE Standard for Distributed Interactive Simulation— Application Protocols

IEEE Computer Society

Sponsored by the
SISO Standards Activity Committee

IEEE
3 Park Avenue
New York, NY 10016-5997
USA

IEEE Std 1278.1™-2012
(Revision of
IEEE Std 1278.1-1995)

19 December 2012

IEEE Std 1278.1™-2012
(Revision of
IEEE Std 1278.1-1995)

IEEE Standard for Distributed Interactive Simulation—Application Protocols

Sponsor
SISO Standards Activity Committee
of the
IEEE Computer Society

Abstract: Data messages, known as Protocol Data Units (PDUs), that are exchanged on a network among simulation applications are defined. These PDUs are for interactions that take place within specified domains called protocol families, which include Entity Information/ Interaction, Warfare, Logistics, Simulation Management, Distributed Emission Regeneration, Radio Communications, Entity Management, Minefield, Synthetic Environment, Simulation Management with Reliability, Information Operations, Live Entity Information/Interaction, and Non-Real-Time protocol.

Keywords: data messages, Distributed Interactive Simulation, IEEE 1278.1™, protocol data units (PDUs), simulation network

---

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Print: ISBN 978-0-7381-7310-8 STD97285
PDF: ISBN 978-0-7381-7362-7 STDPD97285

Notice and Disclaimer of Liability Concerning the Use of IEEE Documents: IEEE Standards documents are developed within the IEEE Societies and the Standards Coordinating Committees of the IEEE Standards Association (IEEE-SA) Standards Board. IEEE develops its standards through a consensus development process, approved by the American National Standards Institute, which brings together volunteers representing varied viewpoints and interests to achieve the final product. Volunteers are not necessarily members of the Institute and serve without compensation. While IEEE administers the process and establishes rules to promote fairness in the consensus development process, IEEE does not independently evaluate, test, or verify the accuracy of any of the information or the soundness of any judgments contained in its standards.

Use of an IEEE Standard is wholly voluntary. IEEE disclaims liability for any personal injury, property or other damage, of any nature whatsoever, whether special, indirect, consequential, or compensatory, directly or indirectly resulting from the publication, use of, or reliance upon any IEEE Standard document.

IEEE does not warrant or represent the accuracy or content of the material contained in its standards, and expressly disclaims any express or implied warranty, including any implied warranty of merchantability or fitness for a specific purpose, or that the use of the material contained in its standards is free from patent infringement. IEEE Standards documents are supplied "**AS IS**."

The existence of an IEEE Standard does not imply that there are no other ways to produce, test, measure, purchase, market, or provide other goods and services related to the scope of the IEEE standard. Furthermore, the viewpoint expressed at the time a standard is approved and issued is subject to change brought about through developments in the state of the art and comments received from users of the standard. Every IEEE standard is subjected to review at least every ten years. When a document is more than ten years old and has not undergone a revision process, it is reasonable to conclude that its contents, although still of some value, do not wholly reflect the present state of the art. Users are cautioned to check to determine that they have the latest edition of any IEEE standard.

In publishing and making its standards available, IEEE is not suggesting or rendering professional or other services for, or on behalf of, any person or entity. Nor is IEEE undertaking to perform any duty owed by any other person or entity to another. Any person utilizing any IEEE Standards document, should rely upon his or her own independent judgment in the exercise of reasonable care in any given circumstances or, as appropriate, seek the advice of a competent professional in determining the appropriateness of a given IEEE standard.

Translations: The IEEE consensus development process involves the review of documents in English only. In the event that an IEEE standard is translated, only the English version published by IEEE should be considered the approved IEEE standard.

Official Statements: A statement, written or oral, that is not processed in accordance with the IEEE-SA Standards Board Operations Manual shall not be considered the official position of IEEE or any of its committees and shall not be considered to be, nor be relied upon as, a formal position of IEEE. At lectures, symposia, seminars, or educational courses, an individual presenting information on IEEE standards shall make it clear that his or her views should be considered the personal views of that individual rather than the formal position of IEEE.

Comments on Standards: Comments for revision of IEEE Standards documents are welcome from any interested party, regardless of membership affiliation with IEEE. However, IEEE does not provide consulting information or advice pertaining to IEEE Standards documents. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments. Since IEEE standards represent a consensus of concerned interests, it is important to ensure that any responses to comments and questions also receive the concurrence of a balance of interests. For this reason, IEEE and the members of its societies and Standards Coordinating Committees are not able to provide an instant response to comments or questions except in those cases where the matter has previously been addressed. Any person who would like to participate in evaluating comments or revisions to an IEEE standard is welcome to join the relevant IEEE working group at http://standards.ieee.org/develop/wg/.

Comments on standards should be submitted to the following address:

Secretary, IEEE-SA Standards Board
445 Hoes Lane
Piscataway, NJ 08854-4141
USA

Photocopies: Authorization to photocopy portions of any individual standard for internal or personal use is granted by The Institute of Electrical and Electronics Engineers, Inc., provided that the appropriate fee is paid to Copyright Clearance Center. To arrange for payment of licensing fee, please contact Copyright Clearance Center, Customer Service, 222 Rosewood Drive, Danvers, MA 01923 USA; +1 978 750 8400. Permission to photocopy portions of any individual standard for educational classroom use can also be obtained through the Copyright Clearance Center.

Notice to users

Laws and regulations

Users of IEEE Standards documents should consult all applicable laws and regulations. Compliance with the provisions of any IEEE Standards document does not imply compliance to any applicable regulatory requirements. Implementers of the standard are responsible for observing or referring to the applicable regulatory requirements. IEEE does not, by the publication of its standards, intend to urge action that is not in compliance with applicable laws, and these documents may not be construed as doing so.

Copyrights

This document is copyrighted by the IEEE. It is made available for a wide variety of both public and private uses. These include both use, by reference, in laws and regulations, and use in private self-regulation, standardization, and the promotion of engineering practices and methods. By making this document available for use and adoption by public authorities and private users, the IEEE does not waive any rights in copyright to this document.

Updating of IEEE documents

Users of IEEE Standards documents should be aware that these documents may be superseded at any time by the issuance of new editions or may be amended from time to time through the issuance of amendments, corrigenda, or errata. An official IEEE document at any point in time consists of the current edition of the document together with any amendments, corrigenda, or errata then in effect. In order to determine whether a given document is the current edition and whether it has been amended through the issuance of amendments, corrigenda, or errata, visit the IEEE-SA Website or contact the IEEE at the address listed previously. For more information about the IEEE Standards Association or the IEEE standards development process, visit the IEEE-SA Website.

Errata

Errata, if any, for this and all other standards can be accessed at the following URL: http://standards.ieee.org/findstds/errata/index.html. Users are encouraged to check this URL for errata periodically.

Patents

Attention is called to the possibility that implementation of this standard may require use of subject matter covered by patent rights. By publication of this standard, no position is taken by the IEEE with respect to the existence or validity of any patent rights in connection therewith. If a patent holder or patent applicant has filed a statement of assurance via an Accepted Letter of Assurance, then the statement is listed on the IEEE-SA Website at http://standards.ieee.org/about/sasb/patcom/patents.html. Letters of Assurance may indicate whether the Submitter is willing or unwilling to grant licenses under patent rights without compensation or under reasonable rates, with reasonable terms and conditions that are demonstrably free of any unfair discrimination to applicants desiring to obtain such licenses.

Essential Patent Claims may exist for which a Letter of Assurance has not been received. The IEEE is not responsible for identifying Essential Patent Claims for which a license may be required, for conducting inquiries into the legal validity or scope of Patents Claims, or determining whether any licensing terms or conditions provided in connection with submission of a Letter of Assurance, if any, or in any licensing agreements are reasonable or non-discriminatory. Users of this standard are expressly advised that determination of the validity of any patent rights, and the risk of infringement of such rights, is entirely their own responsibility. Further information may be obtained from the IEEE Standards Association.

Participants

At the time this draft recommended practice was submitted to the IEEE-SA Standards Board for approval, the Distributed Interactive Simulation (DIS) Product Development Group had the following membership:

James M. McCall, *Chair*
Robert Murray, *Vice Chair*
Robert Byers, *Secretary*
Graham Shanks, *Drafting Group Editor*

Ron Arlund
Joanne Atherton
Jane Bachman
Michael Bates
Curtis Blais
Mary Christopher
Vincent Cloutier
Mark Crnarich
Jonathan Davis
Gina Densler
Timothy DiVecchia
Steven Dix
Uwe Dobrindt
William Dobson
Jason Esteve
Joey Fann
Grayden Figart
Francois Gagnon
Damian Gallegos
Jerry Gibson
Kevin Heffner
Michelle Herman
Frank Hill
Susan Hilt
William Hinkle
Paul Houldsworth
Mike Iannnicelli
Kyle Isakson
Vann Jackson
John Jinkerson
Karl Jones
Stephen Jones
Rob Karl
James Kolgler
Phil Landweer
Jennifer Lewis
Lance Marrou
Michael McCaslin
Don McGregor
Patrick Merlet
Jerry Merritt
Sara Meyer
Duncan Miller
Katherine Morse
Thomas Mullins
John Nicol
Cynthia Nikolai
Kevin O'Brien
Michael O'Connor
William Oates
William Oliver
Willard Phelps
Ed Powell
Riley Rainey
Susanne Riehemann
Bill Rivera
Peter Ross
Christopher Rouget
Peter Ryan
Randy Saunders
Randy Schuetz
Steven Sheasby
James Shiflett
Per-Phillip Sollin
Richard Sorg
Joseph Sorroche
Kevin Springer
Nelson Stephens
Steve Stephens
Eugene Stoudenmire
Jeff Troung
Grant Tudor
Eric Watz
Ralph Weber
Brian Wharry
Doug Wood
Douglas Wood
Lucien Zalcman
Michael Zanghi

The DIS Product Development Group was assisted by the following past members of the DIS Product Development group and the members of the other working groups as listed:

Cory Adams
Michael Aldinger
Rick Altilio
Richard Andrade
Bill Andrews
Matthew Archer
Joseph Armbruster
Stephanie Banaag
Brendan Barsness
Timothy Bartolo
William Batchelor
William Belue
Alan Berry
Christie Best
Aaron Birenboim
Bjarni Birgisson
Kymberlee Boisvert
Edward Bosanquet
Tom Bourne
James Bowman
Douglas Brown
John Burkley
Robert Butterfield
Lance Call
James Campbell
Pat Cannon
Pascal Cantot
Diana Carl
Tyrone Carter
Michael Caruso
James Chambers
Peter Clark
Timothy Clarke
Bruce Clay
Sandy Cook
Lloyd Copeland
Jon Covin
Molly Curl
Timothy Daigle
Michael Davenport
James Dennig
Hoang Doan
Michael Donabella
Arren Dorman
Russell Downing
Jennifer Dyson
Chris Earley
Angela Eiben
Gary England
Christoper Evey
Gary Farmer
Mark Faulk
Michael Faulk
Richard Fedorowicz
Carston Felise
Hurley Fontenot
Michael Gagliano
Jean-Francois Gamache
Joshua George
Cedric Gilmore
Joyce Ginsburg
Susan Goodchild
Craig Goodyear
Chris Graham
Mike Gray
Cynthia Green
Kenneth Greenwell
James Gregg
Mark Gregory
Russell Grieb
Tami Griffith
David Haber
Joseph Hainline
Paul Halliday
Roger Hammond
Johannes Hattingh
Carrie Hardin
Troy Havener
Thomas Hayes
Brian Herndon
Douglas Hodson
Scott Holben
Robert Holcomb
Paul Hoshall
Eric Hougland
Brian Hulbert
Richard Jessop
Caroline Johnson
Kimberly Johnson
Scott Johnson
Douglas Jones
Mike Kassel
Loren Keck
Greg Keene
Ly Kim
Ken Kingston
Mark Knapke
Sam Knight
Jeffery Kraisser
Randy Kubic
Douglas LaFont
Benoit Lalumiere
Lance Legan
Mattias Lindblad
Anthony Lipscomb
Steven Lovelady
Frank Luongo
Mario Marin
Issac Marquez
Sam Marshall
Josh Mason
Tammie McClellan
Terrence McDermott
Bruce McGregor
Sandy McPherson
John Micheletti
Mike Mora
Neil Morris
Patrick Moy
Chau Ngo
Alex Nichols
Robert Nitti
Roger Noreiga
John O'Reilly
Steve Padilla
Jim Parker
Stan Patterson
Wilfredo Perez
Tuan Pham
James Pollock
Jeff Poor
Art Pope
Monte Porter
Douglas Pouk
Mary Pridgen
Barry Prins
Zhou Renxian
Terrance Reynolds
Andrew Robbie
Bruce Robbins
Lawrence Root
David Ropte
Andrew Rutan
Carol Rydzak
John Saicawalo
Lindsay Samora
Joseph Sardella
Geoff Sauerborn
Wesley Saults
Steve Schwalm
John Scimeca
David Senterfitt
Rick Severinghaus
Ken Sewell
Dale Sexton
Robert Shankle
Rick Sharp
Yong Shin
Mark Sladen
Jeffrey Sloan
Clement Smartt
Heather Smith
Mark Speed
Matthew Stafford
Frank Steves
Jerry Szulinski
Armando Terrazas
Joseph Theis
Clay Thomsen
Mark Torpey
Kyle Tygret
Terry Tyson
Tony Valle
Clinton Valverde
Mike Van Boening
David Wade
Steven Walsh
Brian Waltersdorf
Mike Weber
Trey Westmorland
Scott Wilson
James Wing
Donald Wright
John Woodyard
Joseph Zehnle
Lin Zheng

Directed Energy

Riley Rainey, *Subgroup Lead until 2006,* **and Joseph Sorroche***, Subgroup Lead from 2006*

Alan Berry
Aaron Birenboim
Bill Burckel
Chris Burns
Pat Cannon
Timothy Clarke
Ana Cooksey
Michelle Creedon
Ryan Gray
Frank Hill
Friel Joseph
Todd Kellett
Bill Klein
Tony Lashley
Stan Patterson
Tuan Pham
Bob Praus
Barry Prins
Jan Roubidoux
David Schafer
Charles T. Vrahnos

Emissions

Alan Berry, *Subgroup Lead*

Tom Bourne
Larry Boyce
Robert Byers
Steve Casto
Mary Christopher
Glenn Cicero
Timothy Daigle
Timothy DiVecchia
Hoang Doan
Gary England
Grayden Figart
Jeff Fu
Damian Gallegos
Russell Grieb
David Haber
Joseph Hainline
Frank Hill
William Hinkle
Todd Hoeman
Paul Hoshall
Douglas Jones
Stephen Jones
Michael McCaslin
Patrick Merlet
Thomas L. Meyer
Steve Michels
Steve Padilla
Scott Radke
Kevin Richardson
Bill Rivera
Peter Ross
Randy Schuetz
Ed Scott
Brian T. Smith
Joseph Sorroche
Mark Speed
Jerry Szulinski
Steven Westerhouse
Scott Wilson

IFF

Frank Hill, *Subgroup Lead and Mode 5/S IFF Study Group Chair*

Bob Baker
Shelby Barrett
Alan Berry
Larry Boyce
Gary Burkholder
James Chambers
Louis Chan
Bang Choon
Mary Christopher
Mark Crnarich
Timothy Daigle
Lisa Danger
Jonathan Davis
Em Delahostria
Jennifer Dyson
Winston Fairbrother
Daniel Falbe
Jeffrey Fu
Michael Gagliano
Juan Garcia
Jerry Gibson
James Globe
David Hansen
Carrie Hardin
Dave Hershman
Jim Hollenbach
Jae Jun Hwang
Douglas Jones
Jeffery Kraisser
Kimberly Larsen
Larry Lawrence
Rick Levin
Wayne Lindo
David Long
Lee Marden
Lance Marrou
Edward McCall
Sandra McPherson
Michael Montgomery
Scott Moore
Michael Myjak
Tillman Oxford
Donald Neal
Steve Padilla
Adam Parkinson
Ken Peplow
Wilfredo Perez
Mikel Petty
Guillaume Radde
Bill Rivera
Peter Ross
John Seereiter
Ray Shellman
Van Sickle
Robert Steffes
David Taylor
Jan Tegner
Joseph Theis
Allen Turco
Emerson Vallo
Mike Van Boening
Rene Verhage
Brian Wharry
Scott Wilson
Joseph Zehnle
Albert Zhong

Information Operations
Lee Anthony, *Subgroup Lead*

Robert Adams
Richard Anderson
Alan Berry
Larry Boyce
David Cates
Mary Christopher
Donna Clapp
Jeff S. Dean
Hurley Fontenot
Patrick Getchel
Robert B. Hartlage
Frank Hill
Gary Jett
Ray Larrumbide
Sara Meyer
Donald Neal
Elizabeth Park
Mike Patterson
Kevin Powell
Kevin P. Richardson
Richard Salinas
Randy J. Schuetz
Clement D. Smartt
Scott Waldron

Radio Communications
Stephen Jones, *Subgroup Lead*

Brian Ambroiso
Timothy Bartolo
Alan Berry
John Burkley
Robert Butterfield
Robert Byers
Edytka Christner
Mary Christopher
Grayden Figart
Paul Halliday
Brian Herdon
Frank Hill
William Hinkle
Paul Houldsworth
Lance Legan
Lance Marrou
Patrick Merlet
Robert Murray
Nat Napetano
James Robinson
Peter Ross
Peter Ryan
Mark Sladen
Joseph Sorroche
Steven Weiss
Joseph Zehnle

Time
Randy Saunders, *Subgroup Lead*

Mary Christopher
Frank Hill
Douglas Jones
Robert Murray
Steve Padilla
Graham Shanks
Mark Sladen
Tony Valle

Transfer Ownership
Frank Hill, *Subgroup Lead and Transfer Control Study Group Chair*
Patrick Merlet, *Transfer Control Study Group Vice Chair*

Bob Baker
Larry Boyce
Robert Boyles
Robert Byers
Gary Burkholder
Jim Chaney
Louis Chan
Steve Ciallella
Mark Crnarich
Lisa Danger
E.G. Delahostria
Eric Dumais
Winston Fairbrother
Daniel Falbe
Miles Fidelman
Michael Gagliano
Jerry Gibson
James Gill
Scott Hansen
Carrie Hardin
Dave Hershman
Don Johnson
Stephen Jones
Jeffery Kraisser
Rick Levin
Wayne Lindo
Dave Long
Lee Marden
Edward McCall
Bonnie G. McDaniel
Scott Moore
John Morrison
Donald Neal
Scott Newman
Huat Ng
Jonathan Prescott
Wilfredo Perez
Bill Rivera
Ray Shellman
Robert Steffes
Jan Tegner
Joseph Theis
Allen Turco
Emerson R. Vallo
Mike Van Boening
Garth Van Sickle
Gary Vincent
Scott Wilson

Variable Records
Robert Murray, *Subgroup Lead*

Robert Butterfield
Robert Byers
Mary Christopher
Ed Colunga
Grayden Figart
Damian Gallegos
Brian Herndon
Frank Hill
Stephen Jones
Douglas LaFont
Lance Marrou
Mark McCall
William Oates
Steve Padilla
Bill Rivera
John Saicawalo
Chad Simkins
Joseph Sorroche
Jeff Wakefield

Warfare
Frank Hill, Graham Shanks, *Subgroup Leads*

Robert Byers
Troy Havener
William Hinkle
Terrence McDermott
Randy Saunders
Terry Tyson

The following members of the individual balloting committee voted on this standard. Balloters may have voted for approval, disapproval, or abstention.

Keith Chow
Mary Christopher
Thomas Dineen
Uwe Dobrindt
Sourav Dutta
David Fuschi
Randall Groves
Frank Hill
Susan Hilt
Werner Hoelzl
Robert Holibaugh
John Jinkerson
Rameshchandra Ketharaju
James Kogler
Greg Luri
Lance Marrou
Edward McCall
James M. McCall
Sara Meyer
Steven Monson
Thomas Mullins
Robert Murray
Christopher Rouget
Peter Ryan
Randy Saunders
Bartien Sayogo
Randy Schuetz
Graham Shanks
Steven Sheasby
Gil Shultz
James Smith
Joseph Sorroche
Thomas Starai
Eugene Stoudenmire
Walter Struppler
Gerald Stueve
Marcy Stutzman
Ralph Weber
M. Karen Woolf
Janusz Zalewski

When the IEEE-SA Standards Board approved this standard on 30 August 2012, it had the following membership:

Also included are the following nonvoting IEEE-SA Standards Board liaisons:

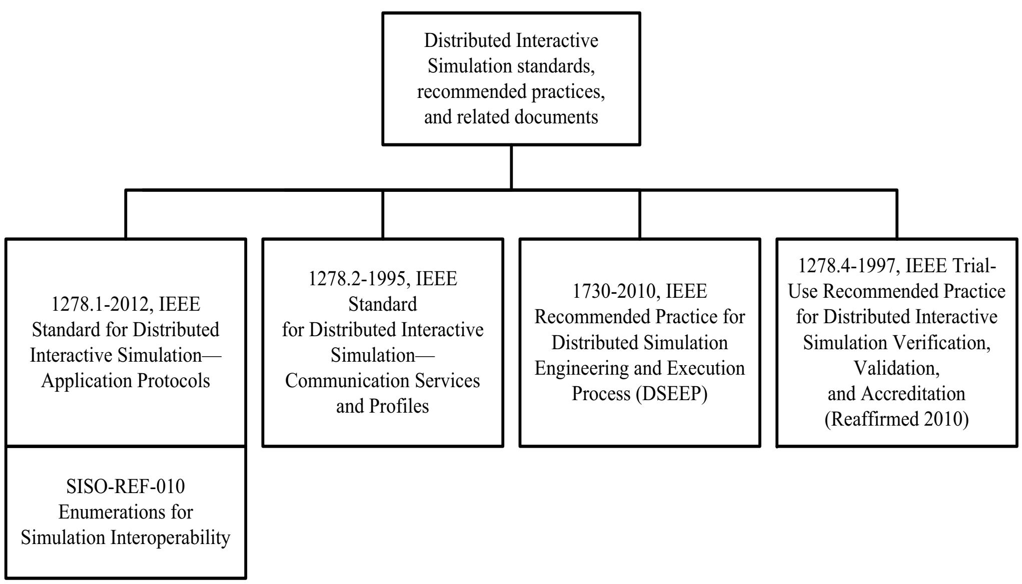
Introduction

This introduction is not part of IEEE Std 1278.1-2012, IEEE Standard for Distributed Interactive Simulation—Application Protocols.

This edition of IEEE Std 1278.1 supersedes both IEEE Std 1278.1-1995 and the material included in the IEEE Std 1278.1a-1998 amendment. Distributed Interactive Simulation (DIS) is a government/industry initiative to define an infrastructure for linking simulations of various types at multiple locations to create realistic, complex, virtual worlds for the simulation of highly interactive activities. This infrastructure brings together systems built for separate purposes, technologies from different eras, products from various vendors, and platforms from various services and permits them to interoperate. DIS exercises are intended to support a mixture of virtual entities with computer-controlled behavior (computer-generated forces), virtual entities with live operators (human-in-the-loop simulators), live entities (operational platforms and test and evaluation systems), and constructive entities (wargames and other automated simulations). DIS draws heavily on experience derived from the Simulation Networking (SIMNET) program developed by the Advanced Research Projects Agency (ARPA), adopting many of SIMNET's basic concepts and heeding lessons learned.

For DIS to take advantage of currently installed and future simulations developed by different organizations, a means had to be found for assuring interoperability between dissimilar simulations. These means were developed in the form of industry consensus standards. The open forum (including government, industry, and academia) chosen for developing these standards was a series of semiannual Workshops on Standards for the Interoperability of Distributed Simulations that began in 1989. The workshops resulted in several IEEE standards and recommended practices.

The relationship between the component documents constituting the set of IEEE DIS documents is shown in the following figure. Used together, these standards and recommended practices will help produce an interoperable simulated environment.



Documentation relationships

The interoperability components addressed by these standards and recommended practices are as follows:

— Application protocols
— Communication services and profiles
— Distributed simulation engineering and execution
— Verification, validation, and accreditation

IEEE Std 1278.1-2012 defines the format and semantics of data messages, also known as Protocol Data Units (PDUs), that are exchanged among simulation applications and simulation management. The PDUs provide information concerning simulated entity states, types of entity interactions that take place in a DIS exercise, data for management and control of a DIS exercise, simulated environment states, aggregation of entities, and the transfer of ownership of entities. This standard also specifies the communication services to be used with each of the PDUs.

An additional, non-IEEE document is required for use with IEEE Std 1278.1-2012. This document is titled Enumerations for Simulation Interoperability and is available from the Simulation Interoperability Standards Organization, Orlando, Florida.

IEEE Std 1278.2™-1995[a] defines the communication services required to support the message exchange described in IEEE Std 1278.1-2012. In addition, IEEE Std 1278.2-1995 provides several communication profiles that meet the specified communications requirements.

Together IEEE Std 1278.1-2012 and IEEE Std 1278.2-1995 provide the necessary information exchange for the communications element of DIS.

IEEE Std 1730™-2010 [B5][b] is a recommended practice defining the processes and procedures that should be followed by users of distributed simulations to develop and execute their simulations; it is intended as a higher level framework into which low-level management and systems engineering practices native to user organizations can be integrated and tailored for specific uses. This recommended practice is intended to replace IEEE Std 1278.3™-1996 [B4]. This recommended practice is used in conjunction with IEEE Std 1278.1-2012 and IEEE Std 1278.2-1995.

IEEE Std 1278.4™-1997 provides guidelines for verifying, validating, and accrediting a DIS exercise. This recommended practice, used in conjunction with IEEE Std 1730-2010 [B5], presents data flow and connectivity for all proposed verification and validation activities and provides rationale and justification for each step.

The principal changes between IEEE Std 1278.1™-1995 and IEEE Std 1278.1a™-1998 and the present standard are as follows:

a) Extensive clarification of requirements throughout the standard.
b) The general requirements have been expanded to cover detailed requirements related to simulations, enumerations, objects, heartbeats, timeouts, thresholds, gateways, and communication services.
c) All identifiers used in the standard have been clarified, and consistent, simplified terminology has been adopted.
d) To provide flexibility and reduce the number of heartbeats, entity heartbeats are now defined by entity kind, domain, and whether the entity is moving or stationary.
e) A new Information Operations (IO) family has been added along with two new PDUs, the IO Action PDU and IO Report PDU, to support information warfare.
f) The Electromagnetic Emission PDU has been clarified, and a new jammer field has been added using an existing padding field to better support a wider range of multiresolution simulations.
g) A new Directed Energy Fire PDU has been added to support high-fidelity directed energy engagements.
h) A new Entity Damage Status PDU has been added to reflect high-fidelity damage to an entity.

[a]Information on references can be found in Clause 2.
[b]The numbers in brackets correspond to those of the bibliography in Annex J.

i) The Transfer Control function has been renamed the Transfer Ownership function, and the Transfer Control Request PDU has been retitled the Transfer Ownership PDU. The entire Transfer Ownership function has been revised to improve its functionality.

j) Transponder and Interrogator requirements have been updated to support high-fidelity Mode 5 Identification Friend or Foe (IFF) and Mode S systems.

k) Time requirements have been extensively clarified and revised.

l) Dead reckoning requirements have been updated including the addition of a new quaternion equation. Annex E Dead Reckoning has been completely revised to clarify requirements, although all the existing formulas have been retained.

m) The Articulation Parameter record found in the Entity State and other PDUs has been renamed the Variable Parameter record to denote that its original design supports more than just its use for articulated and attached parts records. This now provides a way for additional attribute data to be added to entities and detonation characteristics to be added to the Detonation PDU.

n) A new Attribute PDU has been added to support DIS extensibility.

o) The Warfare—General requirements subclause (5.4.2) has been rewritten to incorporate the use of the Fire and Detonation PDUs for expendables and the use of the Detonation PDU for non-munition explosions.

p) The Transmitter PDU was revised to add the capability to have variable Transmitter Parameters records in addition to having a single Modulation Parameters record and multiple Antenna pattern records.

q) Entity separations have been addressed by clarifying how it is to be done for various situations including for multistage missiles and submunition portrayal.

r) Seven new annexes have been added as follows:

    1) Annex A. Warfare (normative). Provides additional requirements related to PDUs used to support the warfare functional area.

    2) Annex B. Specific transponder and interrogator systems (normative). Contains detailed requirements applicable to specific transponder and interrogator systems.

    3) Annex C. Radio systems (normative). Contains detailed requirements applicable to specific radio systems.

    4) Annex D. Objects (normative). Contains detailed requirements related to object types and primary and secondary identifiers.

    5) Annex F. Heartbeats, timeouts, and thresholds (informative). Provides guidance on how to maintain interoperability when some simulations have implemented the new entity timeout requirements and some have not.

    6) Annex G. Time calculations and uses (informative). Provides additional information on time and its uses in a distributed simulation environment.

    7) Annex H. Transfer ownership function (normative). Contains detailed requirements for transfer ownership.

Contents

IEEE Standard for Distributed Interactive Simulation—Application Protocols

1 Overview

1.1 General

This standard explains the information technology protocols required for Distributed Interactive Simulation (DIS) applications. This standard is divided into nine clauses. Clause 1 provides the scope of the standard and details key DIS concepts that will help in understanding the context of this standard. Clause 2 lists references to other standards that are useful in applying this standard. Clause 3 provides definitions of terms, acronyms, and abbreviations that are used in the standard. It is imperative for the user of this standard to thoroughly review these definitions before proceeding on to the other clauses. Clause 4 contains requirements concerning the content and use of Protocol Data Units (PDUs) in DIS exercises. Clause 5 defines the various PDUs and their fields. Clause 6 contains requirements concerning the representation of data within the PDUs. Clause 7 defines the layout and contents of the PDUs. Clause 8 contains the definition of a protocol specifically for applications operating in non-real time. Clause 9 is a stand-alone, self-contained clause that contains both the requirements and the PDU definitions for use by live entities participating in a DIS exercise.

1.2 Scope

This standard is part of a set of standards and recommended practices for DIS applications. Each standard and recommended practice in the set describes one or more of the elements that constitute the DIS environment. As a whole, the set of standards and recommended practices defines an interoperable simulation environment. This particular standard addresses the application protocols.

1.3 Purpose

This standard defines the data messages, known as PDUs, that are exchanged on a network among simulation applications. The messages are organized within specified domains called protocol families. The protocol families included in this standard include Entity Information/Interaction, Warfare, Logistics, Simulation Management, Distributed Emission Regeneration, Radio Communications, Entity Management, Minefield, Synthetic Environment, Simulation Management with Reliability, Information Operations, Live Entity Information/Interaction, and Non-Real Time. Future versions of this standard may contain additional protocol families or PDUs to exchange information about interactions and functions not currently supported.

1.4 Terminology

The following terms, which are defined in Clause 3, form the basis for understanding the key concepts stated in 1.6:

a) DIS
b) Host computer
c) Simulation application
d) Simulation entity
e) Simulation exercise
f) Simulation environment

1.5 Conventions used in this document

The following conventions are used in this document:

a) All numeric values specified in this standard are listed in base ten unless otherwise noted by the suffix (H), which denotes a hexadecimal number.
b) Terms written in all capital letters are symbolic names. The values assigned to the symbolic names are given in 6.1.8.
c) If a term in italics precedes a PDU name, it is used to more clearly and easily define the use of the PDU.
d) The names of fields in PDUs and the names of records are capitalized.
e) All units within PDUs use the International System of Units (SI) (see IEEE/ASTM SI 10™) except when modeling a system that complies with another standard that requires different units.
f) Enumeration fields have values and descriptions defined in SISO-REF-010. The table for each enumeration is identified by a Unique Identifier (UID) in the form "[UID nnn]". In this standard, references to SISO-REF-010 tables use the same UID syntax; for example, "see [UID 874]".
g) World coordinate axes are denoted with uppercase X, Y, and Z. Entity (body) and object coordinate axes are denoted with lowercase x, y, and z.

1.6 Key concepts

1.6.1 Introduction

This subclause contains details of key DIS concepts that will help the reader understand the context of this standard. This information is of a general or an explanatory nature that may be helpful, but it is not mandatory.

1.6.2 Basic architecture concepts

The basic architecture concepts of DIS are an extension of the Simulator Networking (SIMNET) program developed by the Advanced Research Projects Agency (ARPA). The basic architecture concepts for DIS, and their implications as they apply to DIS, are as follows:

a) No central computer controls the entire simulation exercise. Some simulation systems have a central computer that maintains the world state and calculates the effects of each entity's actions on other entities and the environment. These computer systems have to be sized with resources to handle the worst-case load for a maximum number of simulated entities. DIS uses a distributed simulation approach in which the responsibility for simulating the state of each entity rests with separate simulation applications residing in host computers connected via a network. As new host computers are added to the network, each new host computer brings its own resources.

b) Autonomous simulation applications are responsible for maintaining the state of one or more simulation entities. Simulation applications (or simulations) are autonomous and generally responsible for maintaining the state of at least one entity. In some cases, a simulation application will be responsible for maintaining the state of several entities. As the user operates controls in the simulated or actual equipment, the simulation is responsible for modeling the resulting actions of the entity using a simulation model. That simulation is responsible for sending messages to others, as necessary, to inform them of any observable actions. All simulations are responsible for interpreting and responding to messages of interest from other simulations and for maintaining a model of the state of entities represented in the simulation exercise. Simulations may also maintain a model of the state of the environment and nondynamic entities, such as bridges and buildings, that may be intact or destroyed.

c) A standard protocol is used for communicating ground truth data. Each simulation application communicates the state (which is herein called ground truth) of the entity it controls/measures (location, orientation, velocity, articulated parts position, etc.) to other simulations on the network. The receiving simulation is responsible for using this ground truth data to calculate whether the entity represented by the sending simulation is detectable by visual or electronic means. This perceived state of the entity is then presented to the user as required by the individual simulation.

d) Changes in the state of an entity are communicated by its controlling simulation application.

e) Perception of events or other entities is determined by the receiving application.

f) Dead reckoning algorithms are used to reduce communications processing. A method of position/orientation estimation, called dead reckoning, is used to limit the rate at which simulations have to issue state updates for an entity. Each simulation maintains an internal model of the entity it represents. In addition, the simulation maintains a dead reckoning model of its entity. The dead reckoning model represents the view of that entity by other simulation applications on the network and is an extrapolation of position and orientation state using a specified dead reckoning algorithm. On a regular basis, the simulation compares the internal model of its entity with the dead reckoning model of the entity. If the difference between the two exceeds a predetermined threshold, the simulation will update the dead reckoning model using the information from the internal model. The simulation then also sends updated information to other simulations on the network so that they can update their dead reckoning model of the entity. By using dead reckoning, simulations are not required to report the status of their entities as often.

1.6.3 Coordinate systems

1.6.3.1 World coordinate system

Locations in the simulated world are identified using a right-handed, geocentric Cartesian coordinate system called the world coordinate system. The shape of the world is described in NIMA TR 8350.2. The origin of the coordinate system is the centroid of the World Geodetic System 1984 (WGS 84) reference frame (ellipsoid) as defined in NIMA TR 8350.2. The axes of this system are labeled *X*, *Y*, and *Z*, with the positive *X*-axis passing through the prime meridian at the equator, with the positive *Y*-axis passing through 90° east longitude at the Equator and the positive *Z*-axis passing through the north pole as shown in Figure 1. A distance of one unit measured in world coordinates corresponds to a distance of 1 m in the simulated world. A straight line in the world coordinate system is a straight line in the simulated world. This is a rotating reference frame that rotates on a daily period as the Earth rotates.

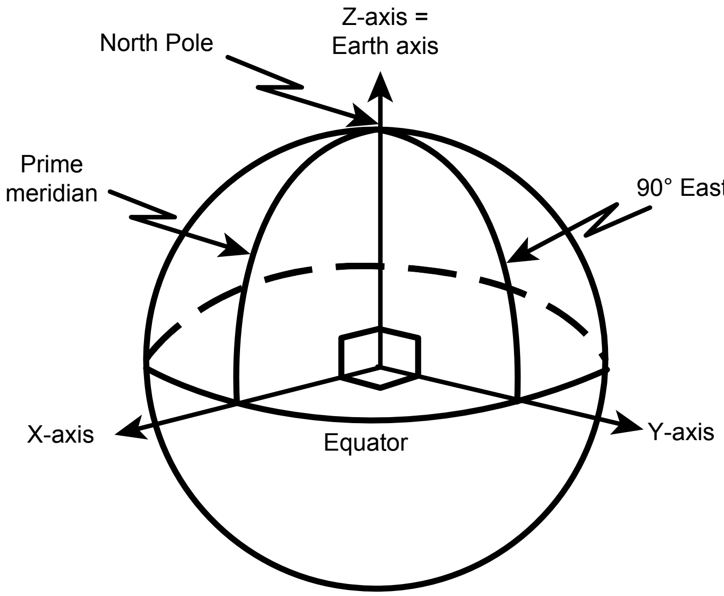


Figure 1—World coordinate system

1.6.3.2 Entity coordinate system

To describe the location and orientation of any particular entity, an entity coordinate system is associated with the entity. This is also a right-handed Cartesian coordinate system with the distance of one unit corresponding to 1 m as in the world coordinate system. The origin of the entity coordinate system is the center of the entity's bounding volume. The bounding volume of an entity does not include its real-world articulated and attached parts, regardless of whether these parts appear in the Entity State PDU. The axes are labeled *x*, *y*, and *z* with the positive *x*-axis pointing to the front of the entity, the positive *y*-axis pointing to the right side of the entity, and the positive *z*-axis pointing out the bottom of the entity as shown in Figure 2.

.

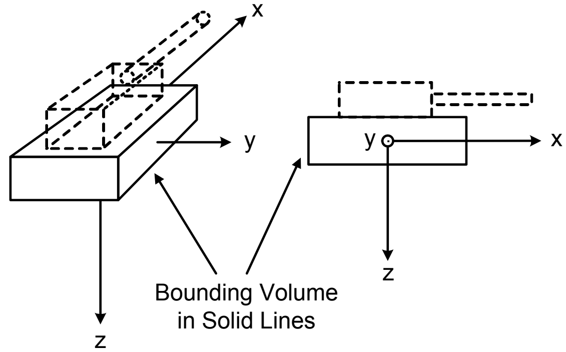


Figure 2—Entity coordinate system

The location of an entity is specified as the position of the origin of its entity coordinate system, expressed in world coordinates. The entity's orientation is specified using three angles that describe the successive rotations needed to transform from the world coordinate system into the entity coordinate system. These angles are called Euler angles and specify a set of three successive rotations about three different orthogonal axes as shown in Figure 3. The order of rotation is first, rotate about z by the angle psi (ψ), then about the new y (y') by angle theta (θ), and then about the newest x (x'') by the angle phi (ϕ). The positive direction of rotation about an axis is defined as clockwise when viewed toward the positive direction along the axis of rotation.

1.6.3.3 Entity velocity and acceleration vectors

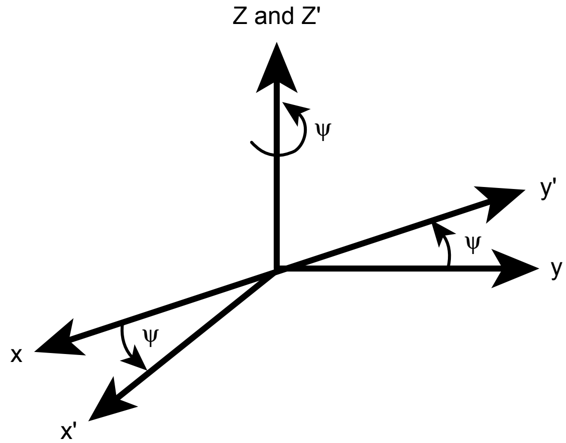
If an entity is moving, its linear velocity is required for dead reckoning. Linear acceleration and/or angular velocity can also be used, if available, to perform a more accurate extrapolation. Velocity and acceleration describe the motion of the entity relative to the rotating Earth, not relative to local effects such as wind or sea currents that can be modeled. For example, the velocity of an aircraft corresponds to ground speed, not to air speed.

Linear velocity and acceleration vectors are transmitted in either world or entity coordinates, depending on the dead reckoning algorithm in use. For linear velocity, the same vector (i.e., the same magnitude and direction from a world view) is represented with either method, only the coordinate axes for defining it are different. For acceleration, however, the centripetal component is removed when converting from world coordinates into body coordinates. For example, an entity turning in a circular path with constant speed has a nonzero, purely centripetal acceleration in world coordinates, but zero acceleration in entity coordinates.

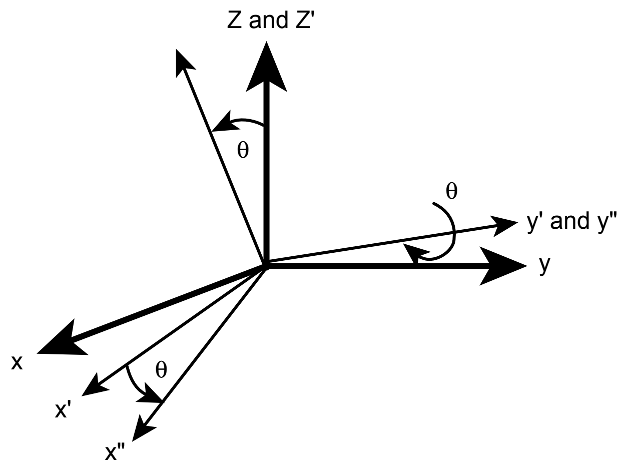
Angular velocity is represented as rotation rates about the entity axes. The rotation rates are generally not the same as Euler angle rates. Instead, angular velocity is a vector $[\omega_x\ \omega_y\ \omega_z]$ in entity coordinates. The vector's direction represents the axis of rotation, and its magnitude $|\omega|$ represents the rate of rotation about that axis. Extrapolation of orientation can be visualized as rotation about the axis at a rate defined by the vector magnitude over a given amount of time.

1.6.3.4 Object coordinate system

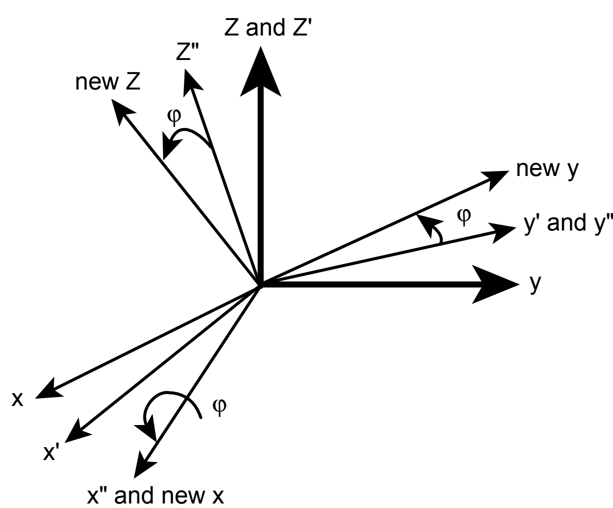
All objects, except as noted below, use the same local coordinate system as described for entities in 1.6.3.2.



(a) First, rotate about *z* by angle psi (ψ)



(b) Second, rotate about *y* (*y*') by theta (θ)



(c) Third, rotate about newest *x* (*x*'') by phi (ϕ)

Figure 3—Definition of Euler angles

a) *Linear Objects*

Linear objects as conveyed in the Linear Object State PDU use the same local coordinate system as with entities but have a different origin. There is one origin per segment as specified by the corresponding Segment Location in the Linear Segment Parameter record. As shown in Figure 4, the location of the origin is on one end of the bounding volume and where the object intersects with the terrain, water, or surface upon which it is placed. If a single object origin is required, then the single origin is the origin of the first segment.

b) *Areal Objects*

The location of the points specified for an areal object are where the object intersects with the terrain, water, or surface upon which it is placed.

c) *Ribbon Bridges (Point Objects)*

Ribbon bridges that are point objects and have an Object Type with a Domain of Land (1), Kind of Passageway (4), Category of Ribbon Bridge (3), and any Subcategory, use an origin identical to a linear segment of a linear object.

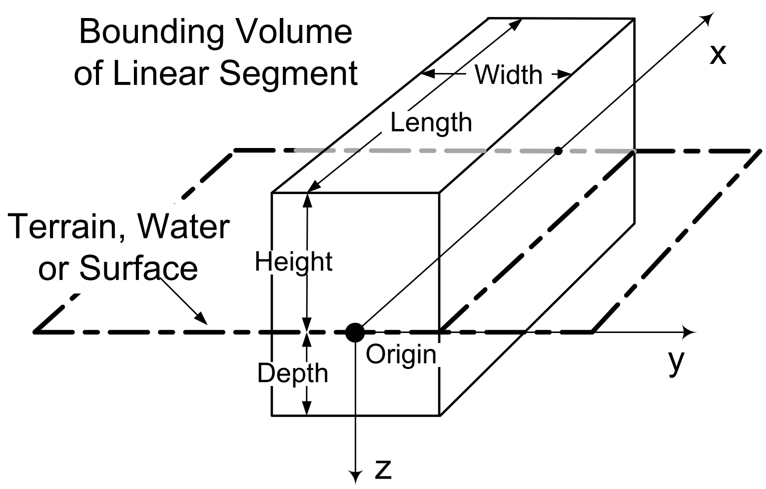


Figure 4—Linear object origin

1.6.4 Communication services

The communication services required by each DIS PDU are described in detail in IEEE Std 1278.2™.

1.6.5 Functional areas for DIS

DIS supports the following functional areas:

- Entity Information/Interaction
- Warfare
- Logistics
- Simulation Management
- Distributed Emission Regeneration
- Radio Communications
- Entity Management
- Minefields

— Synthetic Environment
— Simulation Management with Reliability
— Information Operations
— Non-Real-Time protocol
— Live Entity Information/Interaction

A brief description of each functional area follows:

a) *Entity Information/Interaction*. The PDUs that provide basic entity and entity collision information are listed under this functional area. An entity is a physical object in the synthetic environment that is created and controlled by a simulation that is affected by the exchange of DIS PDUs. An entity may be a live, virtual, or constructive entity. Examples of entities are as follows: tanks, submarines, ships, aircraft, missiles, buildings, bridges, spacecraft, and life forms such as humans and animals. Information on an entity is sent initially, upon change, and at heartbeat intervals. Such information is designed to support a mixed environment of lower and higher fidelity simulations and visual, aural, and sensor models. A variety of data records allows a specific entity to convey additional attributes needed to support higher fidelity simulations. PDUs can be extended using an attribute message.

   Collision interactions that are currently supported include elastic and in-elastic collisions. If two entities collide, the simulations controlling the entities have to be informed of the collision. A message about the collision is sent by each simulation when it detects that its entity has collided with another entity. Each simulation determines the damage to its own entity based on information in the collision message. (See 5.3 and 7.2.)

b) *Warfare*. The PDUs that provide basic warfare information are contained in this functional area. These PDUs support the firing or launch of weapons, including directed energy weapons, the detonation of munitions, the simulation of non-munition explosions, the release of expendables, and the calculation and dissemination of damage effects. When an entity fires a weapon or releases an expendable, the simulation controlling the entity communicates information regarding the fire event that may be needed by other simulations. The detonation of munitions, non-munitions (e.g., fuel tanks), and expendables is also communicated by the simulation controlling the munition, explosion, or expendable. Using the information in the detonation message, all simulation applications controlling affected entities assess damage to their entities. Directed energy weapons are supported by conveying detailed characteristics of the energy deposition such as the type of weapon, duration, and beam shape. The effects of weapons fire, collisions, or other sources of damage are communicated in entity damage status messages. (See 5.4, 7.3, and Annex A.)

c) *Logistics*. Repair and resupply logistic services are modeled in a simulation exercise by means of the logistics PDUs. Messages representing requests for services and the transfer of supplies are exchanged between simulations that are providers of the repair or resupply service and those simulated entities in need of such services. (See 5.5 and 7.4.)

d) *Simulation Management*. PDUs used to manage an exercise and facilitate the operation of the exercise network are contained in the simulation management function. DIS management functions are divided into network management and simulation management. Basic network management functions such as load management, monitoring of nodes and gateways, and error reporting are covered by standard network protocols and tools and are not addressed by the Simulation Management function. Functions of simulation management covered by the Simulation Management function include starting, restarting, pausing, and stopping an exercise; exchanging initialization data; ordering the instantiation and removal of entities; and supporting data collection and dissemination. (See 5.6 and 7.5.)

e) *Distributed Emission Regeneration*. This functional area supports the simulation of designator lasers; active electromagnetic emitters such as radars and electronic identification and surveillance systems; and active acoustic emissions such as from sonar systems. The messages representing these emission sources are designed to provide sufficient data to allow a receiving sensor simulation to properly detect and interact with the emitter source. The local receiving model recreates the interaction of the system being simulated by using operational parameters in the received PDU along with information from stored databases that describe the system's capabilities. (See 5.7, 7.6, and Annex B.)

f) *Radio and Intercom Communications*. Audio and digital message communications play an important role in DIS exercises. The sending (transmitting) entity sends a message defining the details of the communicating device and then the communicated message (voice or digital data). Entities receiving the message can determine their capability to receive the transmitted data and subsequently how to process the received data. Audio communication includes both radio and intercom communications. Tactical data link messages may be conveyed using PDUs in this functional area regardless of which medium is used to send the messages (i.e., a radio, satellite link, a land-based cable, wide area networks, or any other form of communications). (See 5.8, 7.7, and Annex C.)

g) *Entity Management*. This functional area supports larger DIS exercises by providing mechanisms to allow the aggregation or grouping of entities during an exercise and for the reporting of the state of the group or aggregate in place of the states of the individual entities. Specific protocols are contained in this standard that support these capabilities and the capability to transfer the ownership of an entity from one simulation to another. (See 5.9 and 7.8.)

h) *Minefields*. The simulation of minefields and individual mines is supported by the PDUs that belong to this functional area. The exchange of information about mines and minefields can be executed in either a heartbeat or a query response mode. (See 5.10 and 7.9.)

i) *Synthetic Environment*. PDUs within this functional area support the simulation of nonentity synthetic environment objects such as weather, diurnal effects, natural and human-made disturbances (e.g., volcano explosions, earthquakes, and dust and smoke clouds from vehicles or explosions), and terrain-, space-, and water-related environments. This information may include changes to the synthetic environment objects, for instance, the dispersion of chemical clouds over time or changes in terrain by either engineering effects, such as the construction of a bridge or a berm or the destruction of buildings, or by natural effects such as flooding. (See 5.11 and 7.10.)

j) *Simulation Management with Reliability*. PDUs within this functional area perform the same tasks as the Simulation Management family [see item d)]. In addition, this family specifies mechanisms for reliable communication so that critical management tasks are completed even if individual PDUs are dropped. (See 5.12 and 7.11.)

k) *Information Operations*. Information Operations (IO) supports the interoperability of simulated electronic warfare, computer network operations, military deception, and similar operations used to influence or disrupt enemy decision making. DIS conveys the details of IO attacks in IO action messages. IO Action messages can also contain the predicted effects of an attack. The actual effects of an attack are communicated in IO report messages. (See 5.13 and 7.12.)

l) *Non-Real-Time Protocol*. Most DIS exercises operate with a human in the loop or have other real-time requirements; therefore, simulation time has to advance at the same rate as real-world time. However, this standard supports other time methods to allow simulation time to advance at a rate other than real-world time. This is accomplished by describing how existing DIS messages can be used to support non-real-time exercises and experiments. (See Clause 8.)

m) *Live Entity Information/Interaction*. PDUs have been developed with a smaller footprint to support live participants using instrumented ranges where there is limited bandwidth. Live entities include life forms such as soldiers, vehicles, aircraft, and ships. DIS is one of several protocols used on instrumented ranges. In some cases, a gateway may convert live participants represented by DIS or other protocols used on a range into regular simulation entities. In this case, DIS supports the identification of a simulated entity as representing a live participant. (See Clause 9.)

2 Normative references

The following referenced documents are indispensable for the application of this document (i.e., they must be understood and used, so each referenced document is cited in text and its relationship to this document is explained). For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments or corrigenda) applies.

AIMS 03-1000A, Technical Standard for the ATCRBS/IFF/MARK XIIA Electronic Identification System and Military Implementation of Mode S with Changes 1, 2 and 3, 1 January 2011, DoD International AIMS Program Office, Robins AFB, Georgia.[1]

IEEE/ASTM SI 10™, American National Standard for Metric Practice.[2, 3]

IEEE Std 754™, IEEE Standard for Floating-Point Arithmetic.

IEEE Std 1278.2™, IEEE Standard for Distributed Interactive Simulation—Communication Services and Profiles.

International Civil Aviation Organization (ICAO) basic Mode Select (Mode S) related publications[4]:

a) ICAO Doc 9684 AN/951, Manual on the Secondary Surveillance Radar (SSR) Systems.
b) ICAO Doc. 9688 AN/952 Manual on Mode S Specific Services.
c) International Civil Aviation Organization (ICAO) Annex 10 to the Convention on International Civil Aviation, Aeronautical Communications, Volume III Communications Systems.
d) International Civil Aviation Organization (ICAO) Annex 10 to the Convention on International Civil Aviation, Aeronautical Communications, Volume IV Surveillance Radar and Collision Avoidance Systems.

ITU-T Recommendation G.711, Pulse Code Modulation (PCM) of Voice Frequencies.[5]

ITU-T Recommendation G.726, 40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM).

MIL-STD-188-113, Interoperability and Performance Standards for Analog-to-Digital Conversion Techniques.[6]

NIMA TR 8350.2, Third Edition, Amendment 1, 3 January 2000, Department of Defense World Geodetic System 1984 (WGS 84), Its Definition and Relationships with Local Geodetic Systems.[7]

SISO-REF-010, Enumerations for Simulation Interoperability. (SISO-REF-010-00v20-0 or later is required to support this standard. This document is periodically updated.)[8]

SISO-STD-002, Standard for Link 16 Simulations.

---

[1]The latest revision of the AIMS 03-1000A document, and interim changes, is available to authorized U.S. and non-U.S. personnel by contacting the DoD International AIMS Program Office, Robins AFB, Georgia (https://dod-aims.com/).

[2]IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/). ASTM publications are available from the American Society for Testing and Materials (http://www.astm.org/).

[3]The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

[4]The latest edition of ICAO publications is available from the ICAO Web site (http://www.icao.int/). Note that ICAO publications may have errata published as separate documents.

[5]ITU-T publications are available from the International Telecommunications Union (http://www.itu.int/).

[6]This publication is available from DLA Document Services, Bldg 4/D, 700 Robbins Ave, Philadelphia, PA 19111. MIL publications are also available from the U.S. Department of Defense (http://www.defense.gov/).

[7]This publication is available from the National Technical Information Service (NTIS) (http://www.ntis.gov/).

[8]SISO publications are available from the Simulation Interoperability Standards Organization (SISO) (http://www.sisostds.org).

3 Definitions, special terms, acronyms, and abbreviations

3.1 Definitions

For the purposes of this document, the following terms and definitions apply. *The IEEE Standards Dictionary Online* should be consulted for terms not defined in this clause.[9]

absolute time: A reference time synchronized across the exercise, which may or may not be Coordinated Universal Time (UTC) time depending on exercise agreements.

absolute timestamp: A timestamp that contains absolute time. Absolute time is signified by the least significant bit being set to one in a timestamp field of a Protocol Data Unit (PDU). See 4.6.3.

acquiring simulation: The simulation application that will become the new owner of the entity involved in a transfer transaction.

active beam: A beam that is producing detectable electromagnetic or acoustic energy.

active emitter: An emitter system with one or more active beams.

aggregate (unit): A group of entities or a group of other aggregates. The substitution of the word "unit" is used to avoid phrases like "aggregate aggregate."

aggregation: The process of changing the resolution of two or more aggregates by replacing them with a single aggregate at a lower level of detail.

areal object: A synthetic environment object that is geometrically anchored to the terrain with a set of at least three points that come to a closure.

articulated part: A visible part of a simulated entity that is able to move relative to the entity or relative to another articulated part.

attached part: A visible part of a simulated entity that may not move relative to the entity, but that may or may not be present. For example, a bomb on an aircraft wing station.

attribute: A property or characteristic of an entity, object, or event, for example, antenna location. Also, a property inherent in an entity or object or associated with that entity or object for database purposes.

automatic mode: A simulation is in the automatic mode if the simulation application is automatically processing all messages associated with a transfer transaction. A simulation at which an operator manually initiates a transfer is considered to be in the automatic mode if the operator has no control over the completion of the transfer transaction except to manually cancel the transfer.

ballistic munition: Any munition that follows a ballistic trajectory.

beam: The focused emissions from an electromagnetic or active acoustic transmitter. The beam is defined by the main lobe of the antenna pattern.

NOTE—Beams are also referred to as emitter beams in this standard.[10]

[9] *The IEEE Standards Dictionary Online* subscription is available at http://www.ieee.org/portal/innovate/products/standard/standards_dictionary.html.

[10] Notes in text, tables, and figures of a standard are given for information only and do not contain requirements needed to implement this standard.

best effort service: A communication service in which transmitted data is not acknowledged. Such data typically arrives in order, complete and without errors. However, if an error occurs, or a packet is not delivered, nothing is done to correct it (e.g., there is no retransmission).

bit: The smallest unit of information in the binary system of notation.

bit field record: A record whose field structure is defined by explicit bit numbering, not necessarily on octet boundaries.

body coordinates: *See:* **entity coordinate system**.

constructive: Models and simulations that involve computer-generated entities, their simulated human operators, and other objects that represent life forms, crowds, vehicles, communications, environmental processes, cultural and natural features, terrain, structures, and systems whose behavior and interactions are based on rule sets or other behavioral logic including preplanned activities over a period of simulation time. Real people usually control such simulations and may override programmed actions and outcomes. This extends to real people being able to initiate or take control of an existing constructive entity or process, altering preprogrammed actions and terminating an entity. A typical constructive simulation may generate entities that number in the thousands.

dead reckoning: A method for the estimation of the position/orientation of an entity based on a previously known position/orientation and estimates of the passage of simulation time and motion.

disaggregation: The process of changing the resolution of a single aggregate by replacing it with two or more aggregates at a higher level of detail.

Distributed Interactive Simulation (DIS): A time and space coherent synthetic representation of world environments designed for linking the interactive, free-play activities of people in operational exercises. The synthetic environment is created through real-time exchange of data units between distributed, computationally autonomous simulation applications in the form of simulations, simulators, and instrumented equipment interconnected through standard computer communicative services. The computational simulation entities may be present in one location or may be distributed geographically.

Distributed Interactive Simulation (DIS) gateway: A software application that is not part of a host simulation whose purpose is to affect the transmission, receipt, forwarding, or filtering of DIS Protocol Data Units (PDUs), the modification of PDU content, or which may translate between DIS and some other protocol(s) or associated object models.

divesting simulation: The simulation application that currently owns the entity involved in a transfer transaction.

emitter: A device that is able to discharge detectable electromagnetic or acoustic energy.

NOTE—Emitters are also referred to as emitter systems in this standard.

emulation data: Data that emulates the characteristics of real-world systems or message content that uses the exact same data format, method, or resolution. This includes data that represents electromagnetic characteristics of a system that would allow emissions to be detected by a physics-based model.

enumeration: A meaning assigned to a numeric value (e.g., 0 = not specified, 1 = normal, 2 = standby, 3 = emergency mode) that is contained in a field or subfield.

entity: *See:* **simulation entity**.

entity bounding volume: The six-sided rectangular space that minimally enclosed the entity excluding its real-world articulated and attached parts, and whose axes are aligned with the entity coordinate system.

entity coordinate system: A right-handed Cartesian coordinate system, centered at the center of the entity's bounding volume excluding articulated and attached parts, used to define locations of objects with respect to the entity. Also known as body coordinates. See 1.6.3.

entity database: The database(s) used by a simulation to store local entities eligible to be transmitted using the Entity State or Entity State Update PDUs and remote entities derived from those same PDUs.

Euler angles: A set of three angles used to describe the orientation of an entity as a set of three successive rotations about three different orthogonal axes (x, y, and z). See 1.6.3.

exercise: *See:* **simulation exercise**.

exercise agreement: An agreement among exercise participants that defines the requirements that have to be met by the participants in support of exercise objectives. These requirements include those that are not covered by standards or other applicable documents, as well as any deviations allowed from those documents. An exercise agreement, as used in this standard, includes any form of agreement among participants, whether written or communicated only verbally.

fidelity: The degree to which the representation within a simulation is similar to a real-world object, feature, or condition in a measurable or perceivable manner.

field: A series of contiguous bits treated as an instance of a particular data type that may be part of a higher level data structure.

functional data: Data that represents characteristics of real-world systems or message content that does not use the exact same data format, method or resolution.

Guise: A function that provides the capability for an entity to be viewed with one appearance by one group of participants, and with another appearance by another group.

heartbeat: A Protocol Data Unit (PDU) that is issued when a predetermined length of real-world time has elapsed since the last PDU was issued for a given entity, environmental process, or supplemental data [e.g., Electromagnetic Emission (EE) PDU and Identification Friend or Foe (IFF) PDU].

heartbeat timeout: The period of real-world time after which the data associated with a Protocol Data Unit (PDU) type for a specific object is cleared from a receiving simulation's database if no new PDU is received.

heartbeat timer: A variable parameter associated with object Protocol Data Units (PDUs) or supplemental PDUs to indicate when a heartbeat PDU is required to be issued. This allows all simulations to receive a full set of data for an entity or other object within a certain period of real-world time. Entity heartbeat timer parameters are based on entity kinds and platform entity domains and whether they are moving or stationary.

host computer: A computer that supports one or more simulation applications. All host computers participating in a simulation exercise are connected by network(s) including local area networks, wide area networks, radio frequency links, and so on.

interactive mode: The Identification Friend or Foe (IFF) Simulation Mode associated with a simulation that outputs the IFF Protocol Data Unit (PDU) where the interrogator model sends out an IFF PDU at the realistic system frequency and one or more transponder models respond with an IFF PDU to the receipt of that specific interrogation IFF PDU. This IFF Simulation Mode is normally only used between a high-fidelity interrogator and transponder model, such as between a virtual flight simulator transponder and an interrogator model to more realistically portray interactions when not participating in a distributed simulation environment.

internal state data: Data held by a simulation that is not contained in the Entity State, Entity State Update, or an associated supplemental Protocol Data Unit (PDU) [e.g., Electromagnetic Emission (EE) PDU and Identification Friend or Foe (IFF) PDU].

local entity: An entity that is owned by the simulation application and for which it issues Entity State Protocol Data Units (PDUs).

linear object: A synthetic environment object that is geometrically anchored to the terrain with one point and has a segment size and orientation.

live: An entity or other data that is based on data received from real-world, as opposed to simulated, players such as actual ships, aircraft, or instrumented soldiers. If a fielded combat system is playing in an exercise, it is considered a live participant regardless of whether it is in a live, simulated, or mixed live/simulation environment. Live entities may be generated in an exercise using either the Time Space Position Information (TSPI) Protocol Data Unit (PDU) or the Entity State or Entity State Update PDUs.

manual mode: In a transfer transaction, a simulation is in the manual mode if an operator action is required to acknowledge the transfer request (able/not able to comply) or to perform other operator actions during the transfer. Manual mode transfers allow extra real-world time for operator actions.

manual pull transfer: A Pull Transfer that is initiated by an acquiring simulation that is in the manual mode. A manual Pull Transfer is indicated as such in the initial TO Protocol Data Unit (PDU), thus, informing the divesting simulation that the acquiring simulation may require extra real-world time for operator actions.

minimum heartbeat compliance: The case where a simulation does not meet all the heartbeat and timeout requirements of the standard. See F.2.3 for additional information.

multicast: A transmission mode in which a single message is sent to multiple network destinations (i.e., one-to-many).

network management: The collection of administrative structures, policies, and procedures that collectively provide for the management of the organization and operation of the network as a whole.

node: A general term denoting either a switching element in a network or a host computer attached to a network.

non-self-identifying record: A record that does not include a numeric identification field (e.g., Record Type) to uniquely identify the record.

nonsequential separation: The separation of an entity from another entity that does not occur in a fixed sequential order. An example is munitions carried by a fighter aircraft. The station a munition is fired from will vary depending on the operational situation, and there is no fixed pattern as to which munition will be fired first.

object: A representation of a real-world physical item or phenomenon in the synthetic battlespace. Objects are one of the following: an entity, environmental object (point, linear, areal), minefield, aggregate, subaggregate, environmental process, or a designated group of entities. When the term "object" is used alone, it shall mean any of the types of objects listed. Otherwise, it will be prefaced with an object type (e.g., Point object) or the specific term used for a type of object will be used (e.g., entity).

object identifier: The identifier for an object. This is a 48-bit object identifier consisting of the Site Number, Application Number, and Reference Number.

octet: A sequence of 8 bits, usually operated on as a unit.

owner: The simulation that owns the entity and is sending Protocol Data Units (PDUs) related to the entity.

ownership conflict: An ownership conflict exists when two or more simulations are transmitting an entity with the same Entity identifier (ID). See 5.9.4.2.1 g.

point object: A synthetic environment object that is geometrically anchored to the terrain with a single point.

portable weapon: A weapon that is able to be carried by a life form, such as a human entity. This includes such things as hand guns, rifles, and hand-held missile launchers.

primary entity identifer (ID): The Entity ID in a Protocol Data Unit (PDU) to which the data primarily applies when more than one Entity ID is contained in the PDU. For example, for the Fire PDU, the Firing Entity ID is the primary entity ID.

primary object identifer (ID): The primary object ID is the identifier for the main object that the Protocol Data Unit (PDU) conveys information about. For example, the Firing Entity ID is the primary object ID in the Fire PDU.

primary key: In a database, the column or set of columns that uniquely define each record in a table.

protocol: A set of rules and formats (semantic and syntactic) that determines the communication behavior of simulation applications.

Protocol Data Unit (PDU): A Distributed Interactive Simulation (DIS) data message that is passed on a network between simulation applications according to a defined protocol.

pseudo crypto code: A simulated cryptographic code used in the Identification Friend or Foe (IFF) Protocol Data Unit (PDU) to emulate the operation of cryptographic equipment where both the sender and the receiver have to have matching codes in order to be able to exchange data.

pseudo crypto key: A simulated crypto code used in the Transmitter Protocol Data Unit (PDU). *See:* **pseudo crypto code**.

Pull Transfer: A transfer where the Acquiring simulation wants to take ownership of another simulation's entity. A Pull Transfer may be an Auto Pull Transfer or a Manual Pull Transfer.

real-world time: Elapsed time as determined by a chronometer such as a wristwatch, a clock on the wall, or computer time display.

receiving entity: An entity to which a Protocol Data Unit (PDU) is intended as denoted by the inclusion of the Receiving Entity identifier (ID) field in the PDU.

receiving simulation: A term used to indicate a simulation when related to its capability to receive Protocol Data Units (PDUs). It also refers to the type of simulation associated with the Receiving Simulation identifier (ID) field included in a Simulation Management PDU.

reference time: A real-world time defined by exercise agreement for the coordination of timestamps. Any point in simulation time corresponds to at least one point in reference time. If an exercise is reset to an earlier simulation time, simulation time in the interval between the freeze instant and the earlier restart instant will correspond to two different reference times. If an exercise is frozen and resumed, reference time in the interval between the freeze and the resume will correspond to the freeze instant of simulation time. Reference time, like any real-world time, does not stop when simulation time is paused. Accuracy and precision of reference time, relative to Coordinated Universal Time (UTC), impacts the performance of the exercise as described in G.3.

regeneration mode: The Identification Friend or Foe (IFF) Simulation Mode associated with a simulation that outputs the IFF Protocol Data Unit (PDU) where the data represents the interrogator and transponder data that would be sent during normal real-world operations with the actual interactive exchange of messages. In this simulation mode, each system's model stores the received IFF PDU data and then uses it to recreate an interactive interrogation-reply environment. This is the normal IFF Simulation Mode used for the exchange of IFF data in order to substantially reduce bandwidth usage. Otherwise, IFF PDUs would have to be issued continuously to interrogate transponders and to respond to such interrogations as occurs in the real world. An interrogator model simulates this real-time interaction using received IFF data and its internal model of the operational characteristics of the interrogator.

relative time: A reference time that need not be synchronized. Simulation applications using relative time can still be synchronized to a time server. The initial reference time for a simulation application using relative time can be set to zero, to the local time value, or to some other time value.

relative timestamp: A timestamp that contains relative time. Relative time is signified by the least significant bit being set to zero in a timestamp field of a Protocol Data Unit (PDU). See 4.6.3.

remote entity: An entity that is owned by another simulation application.

right-hand rule: Positive rotation is clockwise when viewed toward the positive direction along the axis of rotation.

run rate: The rate at which an exercise is being run or replayed. *1x* means it is running in real time, that is, 1 s of simulation time in the exercise *(Absolute or Relative Time)* equals 1 s of real-world time. *2x* means that it is running at two times real-world time, that is, 1 s of simulation time in the exercise is equal to 2 s of real-world time. *Another way to express it is that 2 s in the exercise takes 1 s of real-world time to accomplish.* An exercise involving actual military systems is typically run at the *1x* rate when executed. It may be replayed at various run rates.

secondary object identifer (ID): A secondary object ID is the identifier for any object other than the primary object that is contained in a Protocol Data Unit (PDU). For example, the Target Object ID is a secondary object ID in the Fire PDU.

self-identifying record: A record that does include a numeric identification field [e.g., Record Type and Record Identifier (ID)] to uniquely identify the record.

serial simulation: A simulation whose simulation application can issue objects (e.g., entities and objects) that exceed the maximum entity or object number (i.e., 65 533) contained in the object identifier [e.g., Entity Identifier (ID) or Linear Object State Protocol Data Unit (PDU)]. In this case, the simulation application is allowed to use precoordinated additional Application Numbers for the same Site Number to generate the additional entities or objects.

sequential separation: The separation of an entity from another entity that occurs in a fixed sequential order. An example is a multistage ballistic missile where each stage falls away as a new entity in a fixed sequence.

simulation application: The executing software on a host computer that models all or part of the representation of one or more simulation entities. The simulation application represents or simulates real-world phenomena for the purpose of training or experimentation. Examples of simulation applications include manned vehicle simulators, computer-generated forces, environment simulators, and computer interfaces between a Distributed Interactive Simulation (DIS) network and real equipment. The simulation application receives and processes information concerning entities created by peer simulation applications through the exchange of DIS Protocol Data Units (PDUs). More than one simulation application may simultaneously execute on a host computer.

NOTE—This document sometimes uses the term “simulation” in place of simulation application.

simulation entity: A physical object in the synthetic environment that is created and controlled by a simulation application and affected by the exchange of Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs). Examples of types of simulated entities are platforms (tanks, submarines, aircraft), munitions, and life forms. Objects such as munitions, radios, and emitters that are attached to platforms or life forms are not entities, although they may become entities if they detach from the platform. It is possible that a simulation application may be controlling more than one simulation entity.

NOTE—Simulation entities are also referred to as *entities* in this standard.

simulation environment: The operational environment surrounding the simulation entities. This environment includes terrain, atmospheric, and oceanographic information. It is assumed that participants in the same Distributed Interactive Simulation (DIS) exercise will be using environment information that is adequately correlated for the type of exercise to be performed.

simulation exercise: An exercise that consists of one or more interacting simulation applications. Simulations participating in the same simulation exercise share a common identifying number called the exercise identifier. These simulations also utilize correlated representations of the synthetic environment in which they operate.

simulation management: A process that provides control of the simulation exercise. Functions of simulation management include start, restart, maintenance, shutdown of the exercise, and collection and distribution of certain types of data.

simulation manager: A simulation application that performs simulation management functions. See 5.6.3 for definitions of these functions.

simulation time: The shared time being simulated within a simulation exercise. This time is established by the simulation management function and is common to all participants in a particular exercise. As a simulation is running, each instant of simulation time corresponds to an instant of reference time. If the simulation uses fixed frame steps, then simulation time advances discretely from one frame to the next. In order to maintain real-time execution, when the simulation completes a frame, it may need to wait until reference time advances to correspond with the simulation time of the next frame. Simulations without fixed frames may advance simulation time by different amounts, waiting as necessary to align the next instant of simulation time with the corresponding instant of reference time. Whenever simulation time is advanced, the interval between the new instant of simulation time and the prior instant of simulation time must equal the interval of between the reference time corresponding to the prior instant of simulation and the reference time corresponding to the new instant of simulation time. This assures 1 s of simulation time passes in 1 s of reference time for real-time operation.

state Protocol Data Unit (PDU): A PDU that updates the state of an entity or other object. State PDUs have a heartbeat requirement. Examples of state PDUs are the Entity State, Entity State Update, Electromagnetic Emission (EE), Transmitter, Identification Friend or Foe (IFF), and Designator PDUs. *Contrast:* **transient Protocol Data Unit (PDU)**.

stationary heartbeat timer: The heartbeat timer that is used when an entity is stationary.

submunition: Any munition that, to perform its task, separates from a parent munition.

supplemental data: Data about an entity or object that is not contained in the basic Protocol Data Unit (PDU) that defines the entity or object (i.e., data contained in the Electromagnetic Emission (EE) PDU and Identification Friend or Foe (IFF) PDU for an entity).

supplemental Protocol Data Unit (PDU): A PDU that provides additional data about an object but does not define the object itself. (See Table D.1 for PDUs that define objects.)

synchronized time: The time in the Timestamp field when the computer running a simulation application that is issuing or receiving Protocol Data Units (PDUs) is synchronized to a time server or to Global Positioning System (GPS) time. Both Absolute and Relative Time can be synchronized to a common time source. When synchronized to an external time source, such as a network time server or GPS receiver, the simulation application keeps the internal computer clock synchronized with the external time source. This maintains the present Absolute or Relative Time value in synchronization with the internal computer clock by incrementing the value based on the internal clock once initial synchronization is attained.

synthetic environment: The integrated set of data elements that define the environment within which a given simulation application operates. The data elements include information about the initial and subsequent states of the terrain including cultural features as well as atmospheric and oceanographic environments throughout a Distributed Interactive Simulation (DIS) exercise. The data elements include databases of externally observable information about instantiable DIS entities and are adequately correlated for the type of exercise to be performed.

threshold: A threshold is used to determine when an update of specific data should be sent based on exceeding a predefined difference in values rather than the time since the last Protocol Data Unit (PDU) was sent. A threshold also includes the case where different logic may be invoked if predefined values are exceeded.

time server: A server computer that reads the reference time from a reference clock and distributes this information to its clients using a computer network. The time server may be a local network time server or an Internet time server.

tracked munition: A munition for which tracking data is required. By necessity, a tracked munition becomes a simulation entity during its flight; its flight path is represented, therefore, by Entity State Protocol Data Units (PDUs).

transfer control: The aggregate management functional capability for transferring aggregates. This is not part of the transfer ownership management functional capability.

transfer ownership: The entity management functional capability and protocol for transferring the ownership of an entity from one simulation application to another.

transient Protocol Data Unit (PDU): A PDU that is sent once per specific event. For example, a Fire PDU is a transient PDU sent to indicate the launch of a munition, expendables, and so on. A transient PDU does not have a heartbeat requirement. *Contrast:* **state Protocol Data Unit (PDU)**.

unicast: A transmission mode in which a single message is sent to a single network destination (i.e., one-to-one).

valid data: Data that conforms to the requirements of the standard and that is appropriate for a given circumstance as determined by validity checks performed by a simulation.

validity check: A check by a simulation to determine whether a Protocol Data Unit (PDU) is received in the proper sequence or when required by the standard, or in accordance with an exercise agreement, and that the data is valid. A simulation can use additional criteria to determine validity beyond that required by this standard.

virtual: An entity or data that is derived from a virtual or mock-up representation of the actual or anticipated system. Examples include flight simulators, mock-ups of Navy ship command information centers and virtual Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) systems based on fielded equipment.

world coordinate system: The right-handed geocentric Cartesian system. See 1.6.3.

3.2 Special terms

appropriate Protocol Data Unit (PDU): A PDU involved in a transfer transaction that (1) is sent or received at the correct point in the message sequences associated with a transfer transaction, and (2) contains correct data.

auto Pull Transfer: A Pull Transfer that is initiated at the originating simulation via automatic logic that does not involve an operator.

***Cancel* TO Protocol Data Unit (PDU):** A Transfer Ownership (TO) PDU with Transfer Type set to a value of Cancel Transfer (7).

Coupled Protocol Data Unit (PDU) Extension: A means of extending a PDU by immediately following it with an Attribute PDU in a PDU bundle and setting the Coupled Extension Indicator in the PDU Status field of the PDU being extended. The two PDUs are coupled so that they are not separated and are delivered together to the end consumer. Transient PDUs are required to use coupling to be extended. State PDUs may use Coupled PDU Extension or may send the Attribute PDU separately (Noncoupled PDU Extension), bundled or not.

***expendable* Detonation Protocol Data Unit (PDU):** A Detonation PDU issued for an expendable, for instance, to indicate a burst of chaff, flare or pyrotechnic ignition, or leaflet drop dispersion.

***expendable* Fire Protocol Data Unit (PDU):** A Fire PDU issued for an expendable.

***explosion* Detonation Protocol Data Unit (PDU):** A Detonation PDU issued for an explosion associated with an entity that is not a munition.

***final* Entity State Protocol Data Unit (PDU):** The last Entity State PDU issued for an entity as indicated by the Entity Appearance record State field (bit 23) set to Deactivated (1).

full heartbeat compliance: The case where a simulation meets all the heartbeat and timeout requirements of the standard. See F.2.3 for additional information.

full transfer ownership capability: The capability to initiate and respond to both Push and Pull Transfer requests.

***heartbeat* Identification Friend or Foe (IFF) Protocol Data Unit (PDU):** An IFF PDU with the Heartbeat Indicator in the Change/Options record set to Heartbeat (1).

Identification Friend or Foe (IFF) Protocol Data Unit (PDU): The present title for the PDU that was formerly the IFF/Air Traffic Control (ATC)/Navigational Aids (NAVAIDS) PDU.

initial **Entity State Protocol Data Unit (PDU):** The first Entity State PDU issued for an entity.

interrogator **Identification Friend or Foe (IFF) Protocol Data Unit (PDU):** An IFF PDU that represents an interrogator (a transmitter that emits a signal to trigger a response from an IFF system).

limited transfer ownership capability: The capability to initiate and/or respond to a Push or Pull Transfer request but not able to both initiate and respond to both Push and Pull Transfer requests. *Contrast:* **full transfer ownership capability**.

minefield Protocol Data Units (PDUs): Any of the four PDUs associated directly with a minefield: Minefield State, Minefield Data, Minefield Query, and Minefield Response Negative Acknowledgment (NACK) PDUs.

munition **Detonation Protocol Data Unit (PDU):** A Detonation PDU issued for a munition.

originating entity: An entity that initiates an interaction [e.g., using the Simulation Management Protocol Data Units (PDUs)] or is the source of information (e.g., in the Supplemental Emission/Enitity State (SEES) PDU). The originating entity is denoted by the inclusion of its entity identifier in the Originating Entity identifier (ID) field of the PDU.

originating simulation: The simulation application that initiates an interaction [e.g., using the IsPartOf Protocol Data Units (PDUs)] or is the source of information [e.g., in the Attribute PDU or the Information Operations (IO) Report PDU]. The originating simulation is denoted by the inclusion of its simulation address in the Originating Simulation ID field of the PDU.

ownership verification check: A check made to verify the present owner of an entity. This is accomplished by comparing the Originating Entity identifier (ID) from the Transfer Request with stored ownership information previously received in the *ownership* Event Report Protocol Data Unit (PDU) or an *ownership* Data PDU.

ownership **Data Protocol Data Unit (PDU):** A Data PDU with a datum record for Datum identifier (ID) = 15800, Ownership Status.

ownership **Data Query Protocol Data Unit (PDU):** A Data Query PDU with Datum identifier (ID) = 15800, Ownership Status. If the simulation that owns the entity has implemented the Transfer Ownership function, it is required to respond with an *ownership* Data PDU.

ownership **Event Report Protocol Data Unit (PDU):** An Event Report PDU with a datum record for Datum identifier (ID) = 15800, Ownership Status.

Pull Transfer request: A TO Protocol Data Unit (PDU) with Transfer Type set to an Auto or Manual Pull Transfer value.

Push Transfer: A transfer where the Divesting simulation wants to give its entity to another simulation.

Push Transfer request: A TO Protocol Data Unit (PDU) with Transfer Type set to a Push Transfer value.

transfer request: A Transfer Ownership (TO) Protocol Data Unit (PDU) with Transfer Type set to a Push or Pull Transfer value.

transfer transaction: The process associated with the transfer of an entity. This process is initiated by issuing the Transfer Ownership (TO) Protocol Data Unit (PDU) for an entity and is completed at the acquiring simulation when it issues the *initial* Entity State PDU and at the divesting simulation when it receives that PDU. A transfer transaction can be terminated without being completed by issuing and receiving a *Cancel* TO PDU between the two simulations involved in the transaction. A transfer transaction can be in effect at one simulation, or both simulations, depending on the phase of the transfer transaction and message exchange.

***transponder* Identification Friend or Foe (IFF) Protocol Data Unit (PDU):** An IFF PDU that represents a transponder (a system that produces IFF responses when it receives an appropriate signal).

3.3 Acronyms and abbreviations

3-D	Three-Dimensional
ABL	Airborne Laser
ACAS	Airborne Collision Avoidance System
ADPCM	Adaptive Differential Pulse Code Modulation
ADS	Active Denial System
ADS-B	Automatic Dependent Surveillance—Broadcast
AEDB	Active Emissions Database
AII	Active Interrogation Indicator
AIMS	Air Traffic Control Radar Beacon System (ATCRBS), Identification Friend or Foe (IFF), and Mark X/XII/XIIA Systems
AIS	Automated Identification System
ANDB	Additional Narrowband Database
ANSI	American National Standards Institute
APA	Additional Passive Activity
ARI	Aircraft Radio Installation
ARPA	Advanced Research Projects Agency
ASCII	American Standard Code for Information Interchange
ATC	Air Traffic Control
ATCRBS	Air Traffic Control Radar Beacon System
ATL	Advanced Tactical Laser
AVLB	Armored Vehicle Launched Bridge
AWACS	Airborne Warning and Control System
BRG	Bearing
BTID	Battlefield Target Identification Device
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance
CADB	Common Acoustic Database
CCTT	Close Combat Tactics Trainer
CEI	Coupled Extension Indicator
CGF	Computer-Generated Forces

CIR	Collision Intermediate Result
CIT	Combined Interrogator/Transponder
CNO	Computer Network Operations
CR	Conditionally Required
DAP	Downlink of Aircraft Parameters
DE	Directed Energy
DER	Distributed Emission Regeneration
D/E	Depression/Elevation
DIS	Distributed Interactive Simulation
DoD	Department of Defense
DRM	Dead Reckoning Model (see Annex E for further description on DRM notation)
DTI	Detonation Type Indicator
EE	Electromagnetic Emission
ELT	Emergency Locator Transmitter
EM	Electromagnetic
EOD	Explosive Ordnance Disposal
EPLRS	Enhanced Position Location Reporting System
ERP	Effective Radiated Power
EW	Electronic Warfare
FAA	Federal Aviation Administration
FOUO	For Official Use Only
FCR	Fire Control Radar
FH	Frequency Hopping
FHC	Full Heartbeat Compliance
FTI	Fire Type Indicator
GED	Group Entity Description
GICB	Ground Initiated Comm-B
GPS	Global Positioning System
HEL	High-Energy Laser
HLA	High-Level Architecture
HPM	HighPower Microwave
IAI	Intercom Attached Indicator
IC	Interrogator Code (Mode S)
ICAO	International Civil Aviation Organization
ID	Identification, Identifier
IED	Improvised Explosive Device

IFF	Identification Friend or Foe
II	Interrogator Identifier (Mode S)
IO	Information Operations
I/P	Identification of Position
IR	Infrared
IRIG-B	Inter-Range Instrumentation Group B
ISLS	Interrogator Side Lobe Suppression
ISM	IFF Simulation Mode
ISO	International Organization for Standardization
ITU	International Telecommunications Union
JTIDS	Joint Tactical Information Distribution System
LE	Live Entity
LEL	Low-Energy Laser
LRAD	Long-Range Acoustic Device
LSB	Least Significant Bit
LVC	Live-Virtual-Constructive
MHC	Minimum Heartbeat Compliance
MIDS	Multi-Information Distribution System
MILDEC	Military Deception
MP	Modulation Parameters
Mode S	Mode Select
MWOD	Multiple Word of Day
MSB	Most Significant Bit
MTHEL	Mobile Tactical High-Energy Laser
MTU	Maximum Transmission Unit
NA	Not Applicable
NACK	Negative Acknowledgment
NASA	National Aeronautics and Space Administration
NAVAIDS	Navigational Aids
NATO	North Atlantic Treaty Organization
NDA	Nozzle Deflection Angle
NIMA	National Imagery and Mapping Agency
NTP	Network Time Protocol
O	Optional
OPSEC	Operations Security
PCM	Pulse Code Modulation

PDU	Protocol Data Unit
PgRF	Pulse group Repetition Frequency
PHASR	Personnel Halting and Stimulation Response
PID	Positive Identification
PRF	Pulse Repetition FrequencyPSYOPPsychological Operations
PW	Pulse Width
QRP	Query Response Protocol
R	Required
RAI	Radio Attached Indicator
RCP	Radio Communications Protocol
RNG	Range
ROC	Rate of Change
RPM	Revolutions Per Minute
RPR FOM	Real-time Platform Reference Federation Object Model
RRB	Reply Receiver "B"
RWR	Radar Warning Receiver
SA	Simulation Application
SADL	Situational Awareness Data Link
SC	Submunition Canister
SD	Supplemental Data
SEDB	Special Event Database
SEES	Supplemental Emission/Entity State
SI	International System of Units
SI	Surveillance Identifier (Mode S)
SIMAN	Simulation Management
SIMAN-R	Simulation Management with Reliability
SIMNET	Simulator Networking
SINCGARS	Single Channel Ground and Airborne Radio System
SISO	Simulation Interoperabililty Standards Organization
SM	Simulation Manager
SPI	Special Process Indicator
SV	Standard Variable
TCAS	Traffic Collision Avoidance System
TDL	Tactical Data Link
TEI	Transferred Entity Indicator
TENA	Test and Training Enabling Architecture

TI	Transaction Initiator
T/I	Transponder/Interrogator
TMD	Tactical Munitions Dispenser
TNT	Trinitrotoluene
TO	Transfer Ownership
TOD	Time of Day
TR	Transaction Respondent
TRN	Turn
TR	Technical Report
TSPI	Time Space Position Information
TWR	Tail Warning Radar
UA	Underwater Acoustic
UAV	Unmanned Aerial Vehicle
UID	Unique Identifier
UTC	Coordinated Universal Time
VP	Variable Parameter
VRT	Vertical Rate of Climb
VTP	Variable Transmitter Parameters
WAN	Wide Area Network
WGS	World Geodetic System
WOD	Word of Day

4 General requirements

4.1 General

This clause contains requirements concerning the content and use of PDUs in DIS exercises.

4.2 DIS exercise

4.2.1 General

A DIS exercise consists of two or more simulations that are interfaced via a network to support an event where DIS is the primary protocol being used. An event may be an exercise, test, experiment, or some other event or activity that requires a distributed simulation environment.

4.2.2 General requirements

The following general requirements shall apply to a DIS exercise and to DIS simulations:

a) A single Exercise Identifier shall be used to denote those simulations that are participating in the same event, such as participating in the same training exercise.
b) A specific simulation shall be identified by a Simulation Address record (see 6.2.80).
c) If a simulation is also represented as an entity, an Entity Identifier shall be assigned and appropriate entity-related PDUs issued for it.
d) No two simulations shall have the same simulation address.
e) All requirements applicable to a single simulation shall also be applicable to a serial simulation (see 4.2.3.3).

4.2.3 Simulations

4.2.3.1 General

A simulation issues and/or receives and processes PDUs. A simulation application may consist of one or more software processes, which can be designated as either processes or applications. However, all PDUs issued under a single Simulation Identifier shall be considered to be part of a single simulation.

A simulation may be an active simulation that manages the event, a producer that issues PDUs representing one or more objects (e.g., entities, environment objects, etc.), or a gateway that processes PDUs. A gateway may perform various functions related to PDUs, such as filtering and PDU manipulation, or it may originate PDUs from a database built from received PDUs from one or more simulations that are then sent onto the exercise network.

A simulation may be a passive simulation such as a stealth viewer, an event monitor, a PDU data analyzer, or a PDU data recorder. An active simulation may also be a single simulation or a serial simulation.

4.2.3.2 Single simulation

A single simulation is the simulation address of a single site and application at that site whose output of any generated objects (e.g., entities, environmental objects, etc.) does not exceed 65 533. (See 6.2.80.) There are two formats used to represent a specific simulation in a PDU or record when referring to the simulation application itself and not to objects it may be able to produce.

a) *32-Bit Format.* This format is used when a simulation is required to be included in a PDU or record where the field is used exclusively for the purpose of denoting a specific simulation application. It will indicate that the Simulation Address record is to be used for the content of the field (see 6.2.80). The field name will usually reflect the purpose for which the simulation address is being included (e.g., Reporting Simulation).

b) *48-Bit Format.* This format is used when a field may contain either a simulation application or an entity or other object identifiers because the latter identifiers use a 48-bit format. In this case, the simulation is represented by the Simulation Identifier record (see 6.2.81). The Simulation Management and Simulation Management with Reliability PDUs are examples of PDUs that use the 48-bit format.

4.2.3.3 Serial simulations

A serial simulation is a single simulation software program that has been assigned multiple simulation addresses because it is capable of generating more than 65 533 objects (e.g., entities, environmental objects, etc.), which is the maximum value that can be contained in the Object Number field of an object identifier for a single Application Number. The presence of a serial simulation and the assigned range of Application Numbers should be coordinated prior to the start of the event or exercise. A serial simulation should be assigned a block of Application Numbers that covers the maximum number of objects expected to be generated for the duration of the exercise.

4.2.4 Enumerations

SISO-REF-010 contains the enumerations associated with the PDUs and records included in this standard. It also further defines subfields for some of the records that are specified in this standard such as the Entity Appearance record. The document also includes the formats for additional records not specified in this standard.

4.2.5 Objects

4.2.5.1 General

Objects are physical objects or phenomena that are present in the synthetic battlespace, or collections of such objects or phenomena. Individual objects may be perceived by simulations, systems, units, and personnel participating in the synthetic battlespace. Not all objects interact with each other or with the simulations, systems, units, and personnel participating in the event.

4.2.5.2 Object identifiers

The designators assigned to uniquely identify objects such as entities, aggregates, minefields, environmental processes, groups of entities and points, and linear and areal objects are referred to as object identifiers. Table 1 is a list of objects and their object identifiers. The following requirements shall apply:

a) Object identifiers are valid for the duration of an exercise.

b) An object identifier has either one or three numeric components. If an object has three numeric components, they are as follows: Site Number, Application Number and a Reference Number. The type of reference number varies depending on the identifier. Table 1 lists object identifiers, object identifier records, and their components.

c) An object identifier shall be unique and shall not duplicate the object identifier used by another active object, regardless of whether they are the same or different object types. That is, if there is an Entity Identifier of 48-2-36, there cannot also be an Object Identifier of 48-2-36 active in the exercise or event simultaneously.

d) An entity or environmental process that is transferred shall retain its original Entity or Environmental Process Identifier (see 5.9.4).

e) An object identifier assigned to an object (e.g., assigned to an entity) shall not change for the life of the object.

f) An object identifier may be reused after the object to which it is assigned has been deactivated and is no longer in the exercise or event. A simulation should sequence through all available object identifier numbers for new objects before reassigning an object identifier that has been previously used.

g) No object identifier shall be assigned a value of NO_SITE, NO_APPLIC and NO_REF_NUMBER.

h) An object identifier need not be registered or retained for future exercises.

Table 1—Object identifiers

Name	Represents	Identifier			Record reference
		Site no.	App. no.	Ref no.[b]	
Entity Identifier	Single entity	16-bit	16-bit	Entity Number	6.2.28
Aggregate Identifier	Aggregates and subaggregates	16-bit	16-bit	Aggregate Number	7.8.2
Point Object Identifier	Single point object	16-bit	16-bit	Object Number	6.2.64
Linear Object Identifier	Single linear object	16-bit	16-bit	Object Number	6.2.64
Areal Object Identifier	Single areal object	16-bit	16-bit	Object Number	6.2.64
Live Entity (LE) Identifier	Live entity represented by an LE PDU.	8-bit	8-bit	Entity Number	6.2.53
Group Identifier	Nonaggregate group of entities	16-bit	16-bit	Group Number	7.8.3
Group Entity Identifier	Entity in a group	16-bit	16-bit	Entity Number	[UID 215-223][c]
Minefield Identifier	Minefield	16-bit	16-bit	Minefield Number	6.2.56
Mine Entity Identifier	Mine entity in a minefield	16-bit[a]	16-bit[a]	Mine Entity Number	6.2.55
Environmental Process Identifier	Single Environmental Process	16-bit	16-bit	Env. Process Number	7.10.2

[a]The Entity Number in conjunction with the Site Number and Application Number values from the primary object identifier in the PDU form the complete 48-bit Entity Identifier for the entity.

[b]All reference numbers are 16-bit unsigned integers.

[c]These tables are in SISO-REF-010.

4.2.5.3 Other identifiers

The term “identifier” is also used to designate other data items that are not objects. Table 2 lists some of these other identifiers along with what they represent, their component(s), and a reference.

Table 2—Examples of other identifiers

Name	Represents	Identifier components	Reference
Simulation Identifier	Simulation.	Site Number Application Number Ref Number = NO_REF_NUMBER (0)	6.2.81
Exercise Identifier	The number assigned to an exercise.	Exercise Number	5.2.3
Event Identifier Event Identifier record	Used to designate a specific event.	Site Number Application Number Event Number	6.2.33
System Identifier System Identifier record	System information for an IFF emitting system.	Numerous fields	6.2.87
Request Identifier	A unique identifier used to identify a specific request and acknowledgment associated with Simulation Management PDUs.	32-bit unsigned integer	6.2.75
Named Location Identifier Named Location Identification record	Information about the discrete positional relationship of the part entity with respect to its host entity. Used with the IsPartOf PDU and the Separation VP record.	Station Name Station Number	6.2.62
Radio Identifier	Identifies a unique radio in an exercise.	Combined Radio Reference Identifier and Radio Number	6.2.70
Intercom Identifier	Identifies a unique intercom in an exercise.	Combined Intercom Reference Identifier and Intercom Number	6.2.47

4.2.5.4 Simulation address

A specific simulation is identified by its simulation address. The simulation address consists of the combination of the Site Number and Application Number. There are two types of simulation addresses, one for simulated objects and one for live objects. These are defined in 6.2.80 and 6.2.54, respectively. When a 48-bit object identifier field in a PDU may contain either an Object Identifier or a simulation address, the simulation is designated by a Simulation Identifier (see 6.2.81).

4.2.5.5 Reference number

A reference number (e.g., entity number and object number) is the third component of an object identifier. Table 1 lists all the reference numbers for object identifiers. The following requirements shall apply:

a) A reference number is never sufficient to uniquely identify an object such as an entity as all simulations generate reference numbers in the same range. It is the combination of the Site Number, Application Number, and reference number that uniquely identifies an object such as an entity.

b) The range of valid reference numbers associated with a specific object is from 1 to 65 533. The mechanism by which reference numbers are assigned is beyond the scope of this standard.

c) A simulation shall not assign a duplicate reference number to one of its objects that is currently active in an exercise.

d) A reference number may be reused by a simulation if it is known that no object is currently active in the exercise that uses that reference number in conjunction with the same simulation address. [See also item f) in 4.2.5.2.]

e) Each simulation shall be able to process a reference number value in the range 0 to 65 535.

f) The following reference numbers are special and shall not be assigned to represent a specific object:

   1) NO_REF_NUMBER. This value indicates that the identifier record is a simulation ID that does not use the reference number field. This value is also referred to as the NO_ENTITY parameter when it is an Entity Identifier (see 6.1.8).

   2) RQST_ASSIGN_ID. This value may be present in the Create Entity PDU to indicate to the addressed simulation that it shall assign the Entity Number for a new entity that it has been requested to create (see 6.1.8).

   3) ALL_OBJECTS. This value shall mean all objects at a simulation. It may be used in the receiving simulation ID field of various Simulation Management PDUs (see 6.1.8).

g) An Entity Identifier Entity Number field shall be set and used as follows:

   1) The Entity Number shall remain assigned to the entity as long as it remains active in the exercise and regardless of subsequent ownership by another simulation (see 5.9.4).

   2) The Entity Number should be incremented to the next available number until all possible numbers have been exhausted. In cases where a simulation application can generate many entities, they should cycle through all possible entity numbers before reusing a number to better support data collection and analysis activities.

   3) An Entity Number may only be reused if the previous entity has been deactivated; i.e., the *final* Entity State or *final* Entity State Update PDU has been issued [i.e., Entity Appearance record State field (bit 23) set to Deactivated (1)], and the applicable entity timeout period has expired. This is done so that the same entity ID when reassigned to a new entity will not be erroneously associated with the previous entity in case the *final* Entity State or *final* Entity State Update PDU was not received by a simulation.

   4) In the event that a simulation rejoins an exercise in progress, there is no requirement that it keep track of the last entity number used in order to continue with entity numbers that are greater. However, it shall not use an entity ID that it previously owned that is now owned and being issued by another simulation present in the exercise due to a transfer of ownership.

4.2.5.6 Primary and secondary object IDs

There may be more than one Object Identifier field contained in a PDU, and they may be the same type (e.g., all are entity IDs). In some cases, it is necessary to distinguish between similar object IDs contained in a PDU. The terms “primary” and “secondary” are used to distinguish between them. Table D.1 lists the primary and secondary object identifiers for PDUs.

4.2.5.7 Object interactions

Some objects may interact with each other, and some may not. The degree and conditions under which objects may interact as reflected in their presence in secondary object ID fields of specific PDUs shall be as described in Annex D.

4.2.5.8 Identifier number assignments

The valid range of object numbers, and the combination of numbers that can be assigned to the Site Number, Application Number, and Reference Number fields of an object or Simulation Identifier, shall be as defined in Table 3. See 5.6.2 for the valid combinations for identifiers in SIMAN and SIMAN-R PDUs.

Table 3—Valid range of identifier numbers

PDU	Field	Site number	Application number	Reference number
Entity State	Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Collision	Issuing Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Colliding Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
Collision-Elastic	Issuing Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Colliding Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
Entity State Update	Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Attribute	Originating Simulation Address	1 to 65 534	1 to 65 534	N/A
	Record Set 1 … S - Entity/Object ID	1 to 65 534	1 to 65 534	1 to 65 533
Fire	Firing Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Target Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
	Munition/Expendable Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
Detonation	Source Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Target Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
	Exploding Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
DE Fire	Firing Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Entity Damage Status	Damaged Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Service Request	Requesting Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Servicing Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Resupply Offer	Receiving Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Supplying Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Resupply Received	Receiving Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Supplying Entity ID	1 to 65 534	1 to 65 534	1 to 65 533

Table 3—Valid range of identifier numbers *(continued)*

PDU	Field	Site number	Application number	Reference number
Resupply Cancel	Receiving Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Supplying Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Repair Complete	Receiving Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Repairing Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Repair Response	Receiving Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Repairing Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
EE	Emitting Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Track/Jam [Entity ID]	1 to 65 534	1 to 65 534	1 to 65 533
Designator	Designating Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Designated Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
UA	Emitting Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
IFF	Emitting Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	Layer 3 Mode 5 Interrogator—Reporting Simulation	1 to 65 534	1 to 65 534	N/A
	Layer 3 Mode 5 Transponder—Reporting Simulation	1 to 65 534	1 to 65 534	N/A
	Layer 4 Mode S Interrogator—Reporting Simulation	1 to 65 534	1 to 65 534	N/A
	Layer 4 Mode S Transponder—Reporting Simulation	1 to 65 534	1 to 65 534	N/A
	Layer 5—Reporting Simulation	1 to 65 534	1 to 65 534	N/A
SEES	Originating Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Transmitter	Radio Reference ID	1 to 65 534	1 to 65 534	1 to 65 533
Signal	Radio Reference ID	1 to 65 534	1 to 65 534	1 to 65 533
Receiver	Radio Reference ID	1 to 65 534	1 to 65 534	1 to 65 533
	Transmitter Radio Reference ID	1 to 65 534	1 to 65 534	1 to 65 533
Intercom Signal	Intercom Reference ID	1 to 65 534	1 to 65 534	1 to 65 533
Intercom Control	Source Intercom Reference ID	1 to 65 534	1 to 65 534	1 to 65 533
	Master Intercom Reference ID	1 to 65 534	1 to 65 534	1 to 65 533
Aggregate State	Aggregate ID	1 to 65 534	1 to 65 534	1 to 65 533
	Aggregate ID 1 … W	1 to 65 534	1 to 65 534	1 to 65 533
	Entity ID 1 … J	1 to 65 534	1 to 65 534	1 to 65 533
IsGroupOf	Group ID	1 to 65 534	1 to 65 534	1 to 65 533

Table 3—Valid range of identifier numbers *(continued)*

PDU	Field	Site number	Application number	Reference number
Transfer Ownership	Originating ID	1 to 65 534	1 to 65 534	0
	Receiving ID	1 to 65 534	1 to 65 534	0
	Transfer Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
IsPartOf	Originating Simulation ID	1 to 65 534	1 to 65 534	1 to 65 533
	Receiving Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
Minefield State	Minefield ID	1 to 65 534	1 to 65 534	1 to 65 533
Minefield Query	Minefield ID	1 to 65 534	1 to 65 534	1 to 65 533
	Requesting Simulation ID	1 to 65 534	1 to 65 534	0
Minefield Data	Minefield ID	1 to 65 534	1 to 65 534	1 to 65 533
	Requesting Simulation ID	1 to 65 534	1 to 65 534	0
		0	0	0
Minefield Response NACK	Minefield ID	1 to 65 534	1 to 65 534	1 to 65 533
	Requesting Simulation ID	1 to 65 534	1 to 65 534	0
Environment Process	Environmental Process ID	1 to 65 534	1 to 65 534	1 to 65 533
Gridded Data	Environmental Simulation ID	1 to 65 534	1 to 65 534	0
Point Object	Object ID	1 to 65 534	1 to 65 534	1 to 65 533
	Referenced Object ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
	Requester Simulation ID	1 to 65 534	1 to 65 534	0
	Receiving Simulation ID	1 to 65 534	1 to 65 534	0
		65 535	65 535	0
Linear Object	Object ID	1 to 65 534	1 to 65 534	1 to 65 533
	Referenced Object ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
	Requester Simulation ID	1 to 65 534	1 to 65 534	0
	Receiving Simulation ID	1 to 65 534	1 to 65 534	0
		65 535	65 535	0
Areal Object	Object ID	1 to 65 534	1 to 65 534	1 to 65 533
	Referenced Object ID	1 to 65 534	1 to 65 534	1 to 65 533
		0	0	0
	Requester Simulation ID	1 to 65 534	1 to 65 534	0
	Receiving Simulation ID	1 to 65 534	1 to 65 534	0
		65 535	65 535	0

Table 3—Valid range of identifier numbers *(continued)*

PDU	Field	Site number	Application number	Reference number
IO Action	Originating Simulation ID	1 to 65 534	1 to 65 534	0
	Receiving Simulation ID	1 to 65 534	1 to 65 534	0
		65 535	65 535	0
	IO Attacker Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	IO Primary Target Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
IO Report	Originating Simulation ID	1 to 65 534	1 to 65 534	0
	IO Attacker Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
	IO Primary Target Entity ID	1 to 65 534	1 to 65 534	1 to 65 533
TSPI	Live Entity ID	1 to 255	1 to 255	1 to 65 533
Appearance	Live Entity ID	1 to 255	1 to 255	1 to 65 533
Articulated Parts	Live Entity ID	1 to 255	1 to 255	1 to 65 533
LE Fire	Firing Live Entity ID	1 to 255	1 to 255	1 to 65 533
	Target Live Entity ID	1 to 255	1 to 255	1 to 65 533
	Munition Live Entity ID	1 to 255	1 to 255	1 to 65 533
LE Detonation	Firing Live Entity ID	1 to 255	1 to 255	1 to 65 533
	Target Live Entity ID	1 to 255	1 to 255	1 to 65 533
	Munition Live Entity ID	1 to 255	1 to 255	1 to 65 533

4.2.6 Heartbeat PDUs

4.2.6.1 General

Heartbeat PDUs are PDUs that are issued when there has been no significant data change (e.g., status or parameter update or threshold exceeded) and a predetermined length of real-world time has elapsed since the last PDU was issued for a given entity, environmental process, emission, transmitter, and so on. Heartbeats confirm that the associated entity or data is still active and present.

Heartbeat PDUs are necessary because multicast PDUs are issued using a best effort communication service with no guarantee that they will reach all recipients on the network. They are also needed to support a simulation entering or reentering an exercise to allow it to receive the present state of all entities and their attributes, which would otherwise not be received unless a data change occurred.

Specific heartbeat requirements for each PDU type that has such a requirement are defined in Clause 5 for the applicable PDU. Some heartbeats are based on the values contained in one or more fields of the PDU type.

Subclause 6.1.8 lists the symbolic names, default values, and other information related to heartbeat parameters.

Table 4 provides a list of the heartbeat parameters for all PDU types.

Table 4—Heartbeat timers

PDU	Heartbeat parameter	Symbolic name
Moving entities		
Entity State/ Entity State Update	Air platform entity	HBT_ESPDU_PLATFORM_AIR
	Land platform entity	HBT_ESPDU_PLATFORM_LAND
	Surface platform entity	HBT_ESPDU_PLATFORM_SURFACE
	Subsurface platform entity	HBT_ESPDU_PLATFORM_SUBSURFACE
	Space platform entity	HBT_ESPDU_PLATFORM_SPACE
	Life form entity	HBT_ESPDU_KIND_LIFE_FORM
	Munition entity	HBT_ESPDU_KIND_MUNITION
	Cultural feature entity	HBT_ESPDU_KIND_CULTURAL_FEATURE
	Environmental entity	HBT_ESPDU_KIND_ENVIRONMENTAL
	Supply entity	HBT_ESPDU_KIND_SUPPLY
	Radio entity	HBT_ESPDU_KIND_RADIO
	Expendable entity	HBT_ESPDU_KIND_EXPENDABLE
	Sensor/Emitter entity	HBT_ESPDU_KIND_SENSOR/EMITTER
Time Space Position Information (TSPI)	Live entity	HBT_PDU_TSPI
Stationary entities		
Entity State/ Entity State Update	All stationary entities regardless of kind or platform	HBT_STATIONARY
Other objects		
Environmental Process	Environmental Process object	HBT_PDU_ENVIRONMENTAL_PROCESS
Gridded Data	Gridded Data object	HBT_PDU_GRIDDED_DATA
Aggregate State	Aggregate object	HBT_PDU_AGGREGATE_STATE
IsGroupOf	IsGroupOf object	HBT_PDU_ISGROUPOF
Minefield State	Minefield object	HBT_PDU_MINEFIELD_STATE
Minefield Data	Minefield object	HBT_PDU_MINEFIELD_DATA
Supplemental data heartbeats		
Appearance	Appearance data associated with a live entity	HBT_PDU_APPEARANCE
Designator	Laser designator	HBT_PDU_DESIGNATOR

Table 4—Heartbeat timers ***(continued)***

PDU	Heartbeat parameter	Symbolic name
DE Fire	Directed energy weapon shot in progress	HBT_PDU_DE_FIRE
EE	Specific kinds of electromagnetic emissions	HBT_PDU_EE
Entity Damage Status	Damage status data associated with an entity	HBT_PDU_ENTITY_DAMAGE
IFF	IFF transponders and interrogators	HBT_PDU_IFF
Transmitter	Transmitter associated with a moving entity	HBT_PDU_TRANSMITTER
	Transmitter associated with a stationary entity	HBT_STATIONARY
Receiver	Receiver associated with an entity	HBT_PDU_RECEIVER
SEES	Supplemental emissions and entity state attributes	HBT_PDU_SEES
UA	Underwater acoustics	HBT_PDU_UA

4.2.6.2 General requirements

The following general requirements shall apply to all PDUs that have a heartbeat requirement:

a) All heartbeat timer parameter values in Table 4 should be user-definable and able to be input during initialization of the simulation. The range and format of these parameters shall be as defined in 6.1.8.

b) When a PDU type is issued to meet a heartbeat requirement, it shall contain all the information associated with that PDU type for that object. Multiple PDUs shall be required if all the data cannot be contained in a single PDU due to PDU maximum size limitations or other restrictions. For example, an entity may have two electromagnetic systems for which EE PDUs are being issued. If it takes two PDUs to send all the data for both systems, then a heartbeat shall consist of sending both PDUs.

c) When a PDU associated with an object is issued due to a change of data, and a complete set of data is not contained in the PDU(s) issued, the heartbeat timer shall not be restarted.

d) A separate stationary heartbeat timer (HBT_STATIONARY) shall be used if a local entity is stationary regardless of its kind or domain. A local entity shall be considered stationary if the dead reckoning algorithm being used is set to Static (1). When this occurs, the heartbeat timer value shall be immediately set to the stationary heartbeat timer value. If the dead reckoning algorithm subsequently changes to a value other than Static (1), then the stationary heartbeat timer shall be immediately replaced by the appropriate heartbeat timer value for the entity's kind or platform. See Annex E for information on the Static Dead Reckoning algorithm.

If all PDUs cannot be transmitted in a heartbeat interval, they shall continue to be sent until all PDUs required to meet a heartbeat requirement have been sent.

4.2.6.3 Mixed heartbeat environments

A mixed heartbeat environment exists when all simulations in an exercise are not fully compliant with heartbeat requirements. See Annex F for information on how to achieve heartbeat interoperability in this situation.

4.2.7 Timeouts

4.2.7.1 General

Timeouts are used to clear out received data from other simulations if a PDU is not received for the data within a specified time interval. These are referred to as heartbeat timeouts and only apply to remote entities, other remote objects, and associated supplemental data. Local data shall never be timed out or removed from a simulation because it did not meet its heartbeat issuance requirement.

Heartbeat timeouts shall only apply to failure to receive specific remote data within a specific period of reference time.

Specific timeout requirements for each PDU type, if it has such a requirement, are defined in Clause 5. Some timeouts are based on the values contained in one or more fields of the PDU type.

Subclause 6.1.8 lists the symbolic names, default values, and other information related to timeout parameters. Certain timeout parameters do not have symbolic names.

4.2.7.2 General requirements

The following general requirements shall apply to all PDUs that have heartbeat timeout requirements:

a) The heartbeat timeout multiplier parameter shall be used to compute a timeout parameter for a PDU type that has a heartbeat requirement. This parameter shall be user-definable and able to be input during initialization of the simulation. The range and format of this parameter shall be as defined in 6.1.8.

b) A separate heartbeat timeout parameter shall be associated with each PDU type that has a heartbeat requirement. It shall be computed by taking the value of the heartbeat timeout multiplier times the specific heartbeat parameter for an object PDU or supplemental PDU. Separate symbolic names are not included in 6.1.8 for timeout parameters. For example, if an air platform entity has a heartbeat of 5 s and the heartbeat timeout multiplier parameter value is 3, the air platform entity heartbeat timeout parameter value is 3 times 5 s = 15 s.

c) The heartbeat timeout parameter for a PDU type associated with a specific object or supplemental data shall be restarted upon receipt of an applicable PDU.

d) A separate stationary heartbeat timer shall be used if a remote object that has a Dead Reckoning Algorithm field in the PDU indicates it is stationary regardless of its kind or domain. A remote entity shall be considered stationary if the dead reckoning algorithm received is set to Static (1). When this occurs, the heartbeat timeout parameter value shall be immediately set to the stationary heartbeat timeout value. If the dead reckoning algorithm subsequently changes to a value other than Static (1), then the heartbeat timeout parameter value shall be immediately set to the appropriate value for the entity's kind or platform. See Annex E for information on the Static Dead Reckoning algorithm.

4.2.7.3 Mixed heartbeat timeout environments

A mixed heartbeat timeout environment exists when not all simulations in an exercise are fully compliant with heartbeat timeout requirements. See Annex F for information on how to achieve heartbeat timeout interoperability in this situation.

4.2.8 Thresholds

4.2.8.1 General

Thresholds are used to determine when an update of specific data should be sent based on exceeding a predefined value. Thresholds also include those related to a predefined value that if exceeded causes a different set of logic to be invoked. Table 5 lists the thresholds related to different PDU types. Default thresholds are only used when the exercise agreement has not specified the threshold values. Subclause 6.1.8 lists all threshold parameter values and associated default values.

Thresholds are only applicable to a sending simulation and not to a receiving simulation. There is normally no interoperability issue for a receiving simulation, or the exercise as a whole, if different simulations use different threshold values unless the sending simulation is issuing so many PDUs that it overwhelms the network or a specific simulation's PDU processing capacity.

See F.3 for examples of situations where different threshold update rates may apply to different simulations and where the update rate may change for an entity over its lifetime.

4.2.8.2 Entity thresholds

There is no requirement within this standard that a single position and orientation threshold value be used by all simulations that are participating in an exercise.

Different position and orientation thresholds may be associated with different entities based on entity type and circumstances. The position and orientation thresholds may vary during the life of the entity. The numbers of entity position and orientation thresholds allowed in an exercise and their values will depend on the requirements of simulations and fielded units that are participating in an exercise. DIS gateways may be used to increase or decrease the rate of threshold-related Entity State and Entity State Update PDUs to meet exercise interoperability requirements.

4.2.8.3 Other thresholds

There is no requirement within this standard that a single threshold value be maintained for each data type. DIS gateways may be used to increase or decrease the rate of threshold-related PDUs to meet exercise interoperability requirements. Table 5 provides a list of threshold parameters specific to each PDU type. Rules for use of threshold parameters are found in the issuance rules for each PDU type.

Table 5—Threshold parameters

Applicable PDUs	Threshold	Symbolic name	Issuance rules
All PDUs	Nonsynchronization	NON_SYNC_THRSH	4.6.4
Collision	Collision	COLLISION_THRSH	5.3.3.2
Entity State/ Entity State Update	Position	DRA_POS_THRSH	5.3.2.3.2
	Orientation	DRA_ORIENT_THRSH	

Table 5—Threshold parameters *(continued)*

Applicable PDUs	Threshold	Symbolic name	Issuance rules
DE Fire	Area aiming direction	DE_AREA_AIMING_THRSH	5.4.5.3
	Energy deposition	DE_ENERGY_THRSH	
	Precision aiming direction	DE_PRECISION_AIMING_THRSH	
EE	Beam geometry azimuth	EE_AZ_THRSH	5.7.3.3
	Beam geometry elevation	EE_EL_THRSH	
	Effective radiated power	EE_ERP_THRSH	
	Frequency	EE_FREQ_THRSH	
	Frequency range	EE_FRNG_THRSH	
	High-density track/jam	EE_HIGH_DENSITY_THRSH	
	Pulse repetition frequency	EE_PRF_THRSH	
	Pulse width	EE_PW_THRSH	
IFF	Azimuth	IFF_AZ_THRSH	5.7.6.3
	Elevation	IFF_EL_THRSH	
Articulated Parts	Position	DRA_POS_THRSH	5.3.2.4
	Orientation	DRA_ORIENT_THRSH	
Designator	Position	DRA_POS_THRSH	5.7.4.3
UA	Shaft revolutions per minute (RPMs)	UA_SRPM_THRSH	5.7.5.3
	Shaft RPM rate of change	UA_SRPM_ROC_THRSH	
	Location with respect to the entity	UA_POS_THRSH	
	Main beam steering azimuth change	UA_ORIENT_THRSH	
	Main beam steering depression/elevation change	UA_ORIENT_THRSH	
SEES	Power setting change	SEES_PS_THRSH	5.7.7.3
	Engine RPM change	SEES_RPM_THRSH	
	Nozzle deflection angle (NDA) change	SEES_NDA_THRSH	

Table 5—Threshold parameters *(continued)*

Applicable PDUs	Threshold	Symbolic name	Issuance rules
Transmitter	Antenna location	TRANS_POS_THRSH	5.8.3.3
	Antenna direction (azimuth/elevation)	TRANS_ORIENT_THRSH	
	Time of day (TOD) difference threshold	HQ_TOD_DIFF_THRSH	C.4.3.4
IsGroupOf	Position	DRA_POS_THRSH	5.9.3.4.1
	Orientation	DRA_ORIENT_THRSH	
	Articulated parts	See Articulated Parts	
Environmental Process	Geometry change	EP_DIMENSION_THRSH	5.11.2.2.3
	Location change	EP_POS_THRSH	
	State change	EP_STATE_THRSH	
Gridded Data	State change	GD_STATE_CHANGE	5.11.2.3.5
	Geometry change	GD_GEOMETRY_CHANGE	
TSPI (live entity)	Position	DRA_POS_THRSH	9.4.2.3
	Orientation	DRA_ORIENT_THRSH	

4.2.9 DIS gateways

DIS gateways are used in DIS exercises and when DIS simulations are participating in exercises using other distributed simulation protocols and architectures. The term "gateway," as used in this standard, includes any software application that is not part of a host simulation whose purpose is to affect the transmission, receipt, forwarding, or filtering of DIS PDUs; the modification of PDU content; or the translation between DIS and some other protocol(s) or associated object models. Common applications that translate DIS PDUs are those that are compliant with the High-Level Architecture (HLA) and Test and Training Enabling Architecture (TENA). The exercise agreement should identify gateways, if deemed appropriate, and any requirements related to their operation.

In addition to specific requirements related to DIS gateways that are contained in other sections of this standard, the following general requirements shall apply:

a) A DIS gateway shall be considered a DIS simulation and adhere to the requirements specified in this standard insofar as they affect the transmission or receipt of DIS PDUs.

b) A DIS gateway that processes DIS PDUs shall not send DIS supplemental PDUs if the PDU for conveying the entity itself (i.e., the Entity State or Entity State Update PDU) is not being sent. (See 3.1 for the definition of supplemental PDU.)

4.2.10 Protocol versions

Each published version of the standard is assigned a protocol version enumerated value. This value is included in the Protocol Version field of the PDU Header record (see 6.2.66) to denote the version of the standard to which a specific PDU adheres. A fundamental premise for fostering simulation interoperability is the ability of simulations to receive and properly process PDUs. Because it is not always feasible for a simulation to upgrade to a new version of the standard, a DIS exercise may include a mixture of protocol versions for the same PDU type. Correct usage of the Protocol Version field is essential for interoperability as the standard changes over time.

A simulation shall adhere to the following rules regarding the Protocol Version field:

a) A simulation should be capable of receiving PDUs in the same exercise with different protocol versions.
b) A simulation may issue PDUs with different protocol versions in a single exercise.
c) A simulation shall comply with the PDU content and issuance rules in the version of the DIS standard corresponding with the protocol version contained in the PDU.
d) A simulation that processes a PDU shall comply with the receipt rules in the version of the DIS standard corresponding with the protocol version contained in the PDU.
e) Once a simulation issues a PDU for an object, then it shall use the same protocol version when it issues all subsequent PDUs of that PDU type for that object.

4.3 Issuing simulation

In a DIS exercise, simulation applications may represent one or more entities. The simulation application shall issue PDUs for each of the entities that it simulates.

4.4 Issuance of PDUs

PDUs shall be issued according to the requirements specified in Clause 5**.** A PDU shall never be issued in advance of the occurrence of the event or state communicated by the PDU.

4.5 Receipt of PDUs

Upon receipt of PDUs, simulation applications shall act as specified in Clause 5. Unless otherwise stated, the actions described shall apply to all the simulation entities to which the PDU is addressed. Depending on the PDU type, the PDU can be addressed to a particular entity or to all entities within an exercise. Simulation applications shall accommodate out-of-order delivery of PDUs.

4.6 Time

4.6.1 General

Basic interoperability between two simulations running in the same exercise requires adequate synchronization to provide timely interaction for a "fair fight." Coordination of simulation time engenders interoperability in the processing of PDUs by the different simulations participating in a distributed simulation network to support operational and other data analysis. However, the degree of synchronization that provides adequate coordination varies according to the design of the distributed simulation. The requirements related to time contained in this standard support both various degrees of simulation time usage and synchronization.

This subclause contains the requirements related to specific types of time used in this standard when the Real Time Protocol is being used. The Non-Real-Time Protocol requirements are covered in Clause 8. When the phrase "shall have the capability" is used, it shall mean that the simulation shall have the capability cited without the need for a software modification and that any required additional hardware that is not permanently connected to the simulation is readily available and able to be interfaced in a "timely manner."

The Clock Time record is described in 6.2.14 and the Timestamp in 6.2.88. Time definitions used in requirements sections are contained in 3.1. Additional time definitions and information about time calculations and usage are included in Annex G.

4.6.2 Simulation time

Simulation time is defined as the shared time being simulated within an exercise. It is also referred to as exercise time. This time representation defines the date and time being simulated across a distributed test, experiment, or other simulation event (hereafter referred to as the exercise). The meaning of simulation time and its initial value are established prior to an exercise by simulation management and based on the exercise agreement. Simulation time may be real-world UTC time, an exercise-specific time in the past or future, or some other number such as counting from zero when the exercise starts. The following requirements are applicable to the use of simulation time:

a) All simulations shall maintain a common simulation time as defined in the exercise agreement, regardless of how simulation time is coordinated. It is recognized that, if the time reference system being used is not synchronized, the common simulation time will eventually drift over the course of the exercise and limit the precision of common simulation time at all simulations. The exercise designers should consider precision and other engineering considerations in making the exercise agreement.

b) Each simulation shall have the capability to coordinate simulation time, incremented in step with the reference time defined for the exercise during exercise run, and not incremented during exercise freeze.

c) Each simulation shall have the capability to maintain simulation time to include both date and time using the Clock Time record (see 6.2.14).

d) Each simulation shall have the capability to initially set simulation time before it first begins to issue PDUs or first receives and processes PDUs from other simulations.

e) Simulation time may be set to any value, including past, present, or future, UTC time, exercise-specific time, or even to zero; and then incremented based on time passage. It cannot be assumed that the simulation time is the same as the time contained in timestamps. However, the clock skew between simulation time and the reference time used for timestamps may be defined to be zero, e.g., for exercises run in real-world time with C2 systems, or it may vary, e.g., during simulation freeze.

f) There is no requirement to start or resume an exercise, or a specific entity or other object, using the Start/Resume PDU. However, if the Start/Resume PDU is implemented either to issue a start or resume action or to receive and process one or both actions, then the simulation shall adhere to the requirements specified in 5.6.5.4.

g) There is no requirement to stop or freeze an exercise, or a specific entity or other object, using the Stop/Freeze PDU. However, if the Stop/Freeze PDU is implemented either to issue a stop or freeze action or to receive and process one or both actions, then the simulation shall adhere to the requirements specified in 5.6.5.5.

4.6.3 Timestamp

This subclause describes the use of the Timestamp field, which is part of the PDU Header record (see 5.2.5):

a) *General Requirements*

   1) The timestamp shall reflect the reference time corresponding to the simulation time that the data in the PDU message is valid. Simulations may transmit PDUs in advance of when the data is valid, and receivers shall delay use of such messages or compensate for the early transmission. Simulations are encouraged not to transmit messages after the data is valid, but delays may occur. Receivers shall compensate for any delayed transmissions, but it is understood that anomalies cannot be completely eliminated. Exercise agreements should set

clear bounds on acceptable delays and anomalies, and should establish mechanisms to enforce them as appropriate.

2) If the PDU is not able to be immediately transmitted, the timestamp shall not be changed to match the time the PDU was actually transmitted unless the data has been revalidated or otherwise updated.

3) If a receiving simulation desires to validate a received timestamp, or to determine whether to ignore it, the requirements specified in 4.6.5 shall apply.

4) There is no requirement for all simulations to use the same time server for synchronization or to automatically obtain synchronization information.

5) Simulation management computers may monitor timestamps to determine whether a simulation's Absolute timestamps are not synchronized to the reference time or to identify network latency problems that are preventing PDUs from arriving within the expected latency at a specific location.

6) A simulation shall be able to send either absolute or relative timestamps, or both, for issued PDUs.

7) A simulation shall be able to receive and process either absolute or relative timestamps, or both, for received PDUs.

8) A simulation may send both absolute and relative timestamps in the same exercise.

b) *Absolute Timestamps*

There is no requirement to be able to include an absolute timestamp in an issued PDU or to accept and process a received PDU with an absolute timestamp. However, if a simulation has the capability to set absolute time in the timestamp for an issued PDU, or to receive and process a PDU that has an absolute timestamp, it shall adhere to the following requirements.

1) General Requirements

i) The simulation shall maintain a clock synchronized to the reference time defined by exercise agreement. The capability to synchronize with a UTC time source shall be required for any simulation that issues or processes absolute timestamps. Selection of a time source synchronized to UTC time provides broad interoperability, but use of other time sources may be appropriate in some exercises. The exercise agreement may restrict which time sources are permissible for an exercise. Time sources include the following: a network time server, UTC derived from GPS timing signals, the PDU Absolute Time Source [see item b4) in 4.6.4], or from another time source allowed by the exercise agreement.

ii) The least significant bit shall be set to one to indicate an absolute timestamp.

2) Issuance Rules. The absolute timestamp shall be set in a PDU as follows:

i) It shall be synchronized to an appropriate time source.

ii) An absolute timestamp shall continue to be the type of timestamp included in all subsequent PDUs issued for the same PDU type for the same object during the life of the object.

3) Receipt Rules. The absolute timestamp shall be processed when received in a PDU as follows:

i) If a simulation has the capability to accept a PDU containing an absolute timestamp, it may ignore the timestamp value and assume that the time of validity is the time of receipt except as needed for dead reckoning in item iii). For example, a Signal PDU used to transmit a radio voice communication that is occurring in real time is obviously applicable at the time of receipt.

ii) If a receiving simulation desires to validate the received timestamp, or to determine whether to ignore it, the requirements specified in 4.6.5 shall apply.

iii) If the absolute timestamp in a PDU represents an entity or other object that includes velocity, the receiving simulation may use the timestamp to dead reckon the remote object. If it is used for this purpose, the requirements of 4.6.6 shall apply.

c) *Relative Timestamps*

There is no requirement to be able to include a relative timestamp in an issued PDU or to accept and process a received PDU with a relative timestamp. However, if a simulation has the capability to set relative time in the timestamp for an issued PDU, or to receive and process a PDU that has a relative timestamp, it shall adhere to the following requirements:

1) General Requirements

i) Simulation management should consider using the Start/Resume PDU to start an exercise so that all simulations using relative time are properly correlated at the beginning of the exercise.

ii) The least significant bit shall be set to zero to indicate a relative timestamp.

2) Issuance Rules

i) A relative timestamp may or may not be synchronized to a time source. However, the synchronization of a Relative Timestamp cannot be known by receivers. The clock used for relative timestamps shall not be discontinuously adjusted to maintain synchronization, as this would invalidate skew calculations in all receivers.

ii) A relative timestamp shall continue to be the type of timestamp included in all subsequent PDUs issued for the same PDU type for the same object during the life of the object.

3) Receipt Rules

i) If a simulation has the capability to accept a PDU containing a relative timestamp, it may ignore the timestamp value and assume that the time of validity is the time of receipt except as needed for dead reckoning in item iii). For example, a Signal PDU used to transmit a radio voice communication that is occurring in real time is obviously applicable at the time of receipt.

ii) If a receiving simulation desires to validate the received timestamp, or to determine whether to ignore it, the requirements specified in 4.6.5 shall apply.

iii) If the relative timestamp in a PDU represents an entity or other object that includes velocity, the receiving simulation may use the timestamp to dead reckon the remote object. If it is used for this purpose, the requirements of 4.6.6 shall apply. Even if relative timestamps are synchronized, skew compensation shall be performed as per item b) in 4.6.6.

NOTE—It is possible to calculate the time of validity for a PDU that is received with a relative timestamp by applying the techniques described in Annex G.

4.6.4 Synchronization

This subclause describes the conditions and mechanisms for synchronization of clocks to a time source. This time source may be either a hardware, network, or PDU source system. Clock synchronization is important because absolute timestamps require synchronized clocks in each simulation:

a) *General Requirements*

1) Absolute time shall be synchronized to a time source defined by exercise agreement as defined in item b1i) in 4.6.3.

2) The mechanism for initiating and achieving time synchronization can be automatic or manually initiated.

3) A simulation should have at least one alternative source of time synchronization.

4) The requirement for synchronized time may be met by any of the methods described in item b). The Network Source method shall be the preferred method.

5) A simulation that is using the absolute or relative timestamp value as part of the dead reckoning logic shall correct the simulation time interval as specified in 4.6.6.

b) *Synchronization Methods*

1) GPS Satellite Source. Using this method, a simulation obtains the present synchronized time using the constellation of GPS satellites. An antenna receives signals from the satellites and can use the information contained within to compute the UTC time, usually to within 100 µs.

2) Local Time Source. Using this method, a simulation obtains the present synchronized time directly via a connection to a local master clock. The local master clock may be synchronized to UTC, or it may maintain a local time. See Clause 2 for the IRIG-B document reference if this method is used.

3) Network Source. Using this method, a simulation obtains the present synchronized time using a time server over a network. This may be an automatic process or manually initiated periodically to remain in synchronization. An automatic process shall be preferred. The time server may produce actual UTC time or some other time base for use by a simulation host clock. The accuracy of network sources is reduced when they compete for network resources with PDUs or other traffic. One approach to maximize accuracy using a network source is to have a separate administrative local area network for access to the network source instead of sharing the network containing simulation data.

4) PDU Absolute Time Source. A simulation may use an optional PDU Absolute Time Synchronization Service provided through the exchange of specific DIS PDUs. (This is based on the service described in 8.6.2.4 for the Non-Real-Time Protocol but is applicable to the Real Time Protocol as well.) This service may be used as specified in 4.6.7.

c) *Loss of Synchronization*. Synchronization may be lost for several reasons. The following requirements relate to the loss of synchronization:

1) The most common reasons for loss of synchronization are:

i) The interface to the selected time server has been lost.

ii) The time server is not operational.

iii) The source used by the time server itself is no longer available (e.g., it is no longer receiving a GPS time signal).

iv) A manual initiated time synchronization action has not been taken in sufficient time to prevent clocks going out of synchronization.

2) The exercise agreement should establish an overall exercise Non-Synchronization threshold (NON_SYNC_THRSH) variable parameter value prior to the exercise:

i) If no Non-Synchronization threshold is established, the default threshold specified in 6.1.8 shall be used.

ii) A simulation may have a more stringent synchronization requirement than is required by the exercise agreement. In this case, it shall be up to the simulation itself to correct received absolute timestamps to meet its requirements.

3) A simulation management computer should continuously check for out-of-synchronization simulations especially if they are indicated to be using absolute timestamps. A simulation itself should also monitor whether it has remained in synchronization, using the backup time source (if available). A simulation is out of synchronization if the time in the timestamp differs from the locally held time by more than the Non-Synchronization threshold.

4) If a simulation determines that it is no longer in synchronization:

i) If it is maintaining absolute timestamps, the timestamps shall continue to indicate that they are absolute timestamps.

ii) A simulation shall either initiate resynchronization with the primary time server or attempt to synchronize with another direct or indirect source of time depending on the cause of the loss of synchronization.

4.6.5 Timestamp validation

A simulation may validate the reference time received in an absolute or relative timestamp. Time may be validated, but this feature has a significant impact on distributed simulation cohesion when clocks are not precisely synchronized. Many exercises will continue to correct timestamps outside the valid window to reduce the anomalies introduced in the simulations. If the simulation implements time validation, the following requirements shall apply:

a) If the received time type (absolute or relative time) in a PDU Timestamp does not match the locally used time type, the PDU shall not be discarded solely on the basis that the time types are different. A simulation using relative time should be able to process a received PDU with an absolute timestamp with no adverse affect. Likewise, a simulation using absolute time should be able to process a received PDU with a relative timestamp with no adverse affect so long as the synchronization accuracy needed is not greater than 1 millisecond difference. See G.8 for a description of time error correction techniques.

b) If an exercise agreement requires timestamp validation to determine whether a PDU should be accepted, then the following rules shall apply:

    1) If the timestamp is ahead of the reference time corresponding to the simulation time by greater than the value contained in the TIMESTAMP_AHEAD parameter, the simulation shall discard the PDU. The TIMESTAMP_AHEAD parameter may be very large in exercises where early PDUs are held in queue in accordance to item 4). A data log may be maintained of the discarded PDU and the reason for the discard. An alert should also be generated to an operator at the simulation.

    2) If the timestamp is behind the reference time corresponding to the simulation time by greater than the value contained in the TIMESTAMP_BEHIND parameter, the simulation shall discard the PDU. In this case, a data log should be maintained of the discarded PDU and the reason for the discard. An alert should also be generated to an operator at the simulation.

    3) If the timestamp is behind the reference time corresponding to the simulation time, but within tolerance, and the PDU relates to an entity or other object that has velocity, a simulation that has the capability to use dead reckoning shall use the difference between a received timestamp and locally held simulation time as part of its dead reckoning logic to correct the x, y, z position of the object.

    4) If the timestamp is ahead of the reference time corresponding to the simulation time, but within tolerance, and the exercise agreement specifies that early messages should be saved, the simulation shall store the message until the timestamp matches the simulation time.

    5) If the timestamp is ahead of the reference time corresponding to the simulation time, but within tolerance, and the exercise agreement specifies that early messages should be processed, and the PDU relates to an entity or other object that has velocity, a simulation that has the capability to use dead reckoning shall use the difference between a received timestamp and locally held simulation time as part of its dead reckoning logic to correct the x, y, z position of the object.

    6) For other timestamps within tolerance, where the PDU does not relate to an entity with velocity or the simulation lacks the capability to use dead reckoning, the timestamp shall be considered to hold the reference time corresponding to the simulation time.

4.6.6 Dead reckoning

A method of position/orientation estimation called dead reckoning shall be employed to limit the rate at which Entity State or Entity State Update PDUs are issued, as specified in 5.3.2.3. The "dead reckoning algorithm in use" is the dead reckoning algorithm specified in the Dead Reckoning Algorithm subfield of the Dead Reckoning Parameter field of the Entity State PDU. This algorithm shall be used by the receiving simulation application to perform dead reckoning calculations for the received entity.

Dead reckoning algorithms extrapolate the future position and orientation of an entity using parameters transmitted in an Entity State PDU. The Timestamp is one of these parameters, which is an indication of the reference time corresponding to the simulation time of validity for the start of the extrapolation. Some simulations can meet their requirements without the use of dead reckoning extrapolations, such as a map display that indicates a missile launch with an icon at the launch point and the detonation point. These simulations do not need to have the capability to use dead reckoning.

Any simulation that has the capability to use dead reckoning shall implement the following mechanisms related to time:

a) For absolute timestamps, the Δt parameter in Annex E shall be computed as the difference between the timestamp in the PDU and the absolute time for which the position is be estimated, usually the reference time corresponding to the current simulation time or the simulation time of the next simulator frame.

b) For relative timestamps, the Δt parameter in Annex E shall be computed as the difference between the timestamp in the PDU and the reference time corresponding to the current simulation time or the simulation time of the next simulator frame correcting for the Timestamp offset between the sending host and the receiving host. To perform this correction, each simulation shall maintain an estimate of the timestamp offset with other hosts in the exercise, considering the Simulation Address only. Exercise agreements may provide additional requirements on the accuracy of these estimates, using Annex G as background.

c) For relative timestamps used with timestamp validation, the TIMESTAMP_AHEAD and TIMESTAMP_BEHIND parameters shall not be set smaller than the estimated timestamp offset for the sending host, to permit PDUs to be processed while the timestamp offset estimates are firming up.

4.6.7 PDU absolute time synchronization service

This synchronization service is offered when the simulation has no local mechanisms for clock synchronization. It is based on the Absolute timestamp of a designated Simulation Manager (SM) computer compensated by an estimation of the communication delay between the issuance of the *clock synchronization* Action Request-R PDU and the receipt of the associated *clock synchronization* Action Response-R PDU:

a) *General Requirements*
   1) There is no requirement that an SM computer or simulation has the capability to use the PDU Absolute Time Synchronization Service. However, if it is implemented, the following requirements shall be met:
      i) The SM computer may act as the time source.
      ii) If not acting as a time source, the SM computer shall be synchronized to a time server.
      iii) Only one SM computer shall respond to a request for PDU Absolute Time. The method by which this is accomplished is beyond the scope of this standard.

b) *Initiation of a PDU Time Request*. A simulation that desires to request Absolute Time using the PDU Absolute Time Synchronization shall proceed as follows:
   1) A Time Synchronization Action Request-R PDU shall be issued addressed to the SM computer:
      i) The timestamp shall be set to a Relative Time of zero if it has not been previously synchronized and has not issued any PDUs since the simulation started.
      ii) If PDUs have been issued with Relative Timestamps, the present relative time shall be used.
      iii) If PDUs have been issued with Absolute Timestamps, the present Absolute Time shall be used.

2) The SM shall return the equivalent absolute timestamp in the feedback time parameter of the Time Synchronization Action Response-R PDU. After the receipt of the Action Response-R PDU, the application shall use an absolute timestamp to set its absolute time.
3) The receiving simulation shall process the Time Synchronization Action Response-R PDU, taking into account communications latency, and use the received UTC time to synchronize its absolute time.

5 PDUs for DIS

5.1 Purpose

The specific requirements for use of PDUs in a DIS exercise are established in this clause.

5.2 PDU Header

5.2.1 General

A PDU Header shall be the first part of each PDU and shall contain data common to all PDUs. There is one PDU Header format for non-Live Entity (LE) PDUs (see 6.2.66) and one for LE PDUs (see 9.3.2). A Simulation Management PDU Header consists of the standard PDU Header and an Originating ID and Receiving ID field (see 6.2.82).

NOTE—The requirement to place the PDU Header as the first part of each PDU is a global requirement for all PDUs, and thus, the PDU Header is not listed in the subclauses that identify the information contained in each specific PDU.

5.2.2 Protocol version

The PDU Header shall specify the version of DIS protocol to which the PDU pertains. See 4.2.10 for additional information and associated issuance and receipt rules.

5.2.3 DIS exercise identification

Each DIS exercise shall be distinguished from other exercises by the use of an Exercise Identifier. An identifier that is currently not in use on the network shall be assigned. The management and assignment of Exercise Identifiers to an exercise is outside the scope of this standard.

5.2.4 Families and types of PDUs

The list of protocol families and associated PDU types is as follows. The enumeration values for the PDU Type [UID 4] and Protocol Family [UID 5] fields in the PDU Header record are also given in parentheses:

a) Entity Information/Interaction (1) (defined in 5.3 and 7.2)
   1) Entity State (1)
   2) Collision (4)
   3) Collision-Elastic (66)
   4) Entity State Update (67)
   5) Attribute (72)
b) Warfare (2) (defined in 5.4 and 7.3)
   1) Fire (2)
   2) Detonation (3)

   3) Directed Energy Fire (68)
   4) Entity Damage Status (69)

c) Logistics (3) (defined in 5.5 and 7.4)
   1) Service Request (5)
   2) Resupply Offer (6)
   3) Resupply Received (7)
   4) Resupply Cancel (8)
   5) Repair Complete (9)
   6) Repair Response (10)

d) Simulation Management (5) (defined in 5.6 and 7.5)
   1) Create Entity (11)
   2) Remove Entity (12)
   3) Start/Resume (13)
   4) Stop/Freeze (14)
   5) Acknowledge (15)
   6) Action Request (16)
   7) Action Response (17)
   8) Data Query (18)
   9) Set Data (19)
   10) Data (20)
   11) Event Report (21)
   12) Comment (22)

e) Distributed Emission Regeneration (6) (defined in 5.7 and 7.6)
   1) Electromagnetic Emission (23)
   2) Designator (24)
   3) Underwater Acoustic (UA) (29)
   4) IFF (28)
   5) Supplemental Emission/Entity State (SEES) (30)

f) Radio Communications (4) (defined in 5.8 and 7.7)
   1) Transmitter (25)
   2) Signal (26)
   3) Receiver (27)
   4) Intercom Signal (31)
   5) Intercom Control (32)

g) Entity Management (7) (defined in 5.9 and 7.8)
   1) Aggregate State (33)
   2) IsGroupOf (34)
   3) Transfer Ownership (35)
   4) IsPartOf (36)

h) Minefield (8) (defined in 5.10 and 7.9)
   1) Minefield State (37)
   2) Minefield Query (38)
   3) Minefield Data (39)

    4) Minefield Response Negative Acknowledgment (NACK) (40)

i) Synthetic Environment (9) (defined in 5.11 and 7.10)
    1) Environmental Process (41)
    2) Gridded Data (42)
    3) Point Object State (43)
    4) Linear Object State (44)
    5) Areal Object State (45)

j) Simulation Management with Reliability (10) (defined in 5.12 and 7.11)
    1) Create Entity-R (51)
    2) Remove Entity-R (52)
    3) Start/Resume-R (53)
    4) Stop/Freeze-R (54)
    5) Acknowledge-R (55)
    6) Action Request-R (56)
    7) Action Response-R (57)
    8) Data Query-R (58)
    9) Set Data-R (59)
    10) Data-R (60)
    11) Event Report-R (61)
    12) Comment-R Message (62)
    13) Record Query-R (65)
    14) Set Record-R (64)
    15) Record-R (63)

k) Information Operations (13) (defined in 5.13 and 7.12)
    1) Information Operations Action (70)
    2) Information Operations Report (71)

l) Live Entity (LE) (11) (defined in Clause 9)
    1) Time Space Position Information (TSPI) (46)
    2) Appearance (47)
    3) Articulated Parts (48)
    4) LE Fire (49)
    5) LE Detonation (50)

m) Non-Real-Time Protocol (12) (defined in Clause 8)

PDU type codes and protocol family codes in the range of 129 through 255 have been reserved for experimental purposes.

5.2.5 Timestamp

Each PDU contains a Timestamp field as part of the PDU Header record. The timestamp shall be used to indicate the simulation time that the data in the PDU is valid as determined by the simulation that issued the PDU. See 4.6.

5.2.6 Length of PDU

The Length field shall specify the number of octets from and including the first octet of the header to and including the last octet of the PDU.

5.2.7 PDU status record

The PDU Status record shall specify status information related to one or more PDU types (see 6.2.67).

5.3 Entity information/interaction

5.3.1 General

Information associated with the appearance and location of an entity shall be communicated in a DIS exercise through the use of either an Entity State PDU (see also 7.2.2) or an Entity State Update PDU (see also 7.2.5). Information associated with collisions between entities shall be communicated in a DIS exercise through the use of the Collision PDU (see also 7.2.3) or a Collision-Elastic PDU (see also 7.2.4).

5.3.2 Entity State PDU

5.3.2.1 Purpose

The Entity State PDU shall communicate information about an entity's state. This includes state information that is necessary for the receiving simulation applications to represent the issuing entity in the simulation applications' own simulation.

5.3.2.2 Information contained in the Entity State PDU

The Entity State PDU shall contain the following information:

a) Identification of the entity that issued the PDU
b) Identification of the force to which the entity belongs
c) Issuing entity's specific entity type
d) Issuing entity's alternate entity type for use with the Guise function (see 5.3.2.6)
e) Information about the location of the entity in the simulated world and its orientation, including:
    1) Location with respect to the world
    2) Velocity
    3) Orientation
    4) Dead reckoning parameters that shall be employed when extrapolating the position of the entity. (Values in this field shall include the dead reckoning algorithm in use, linear acceleration, and angular velocity. Other values for this field shall be as defined in Annex E.)
f) Information required for representation of the entity's appearance including:
    1) Appearance of the entity (e.g., normal, smoking, on fire, producing a dust cloud, etc.)
    2) Markings
    3) Presence of Variable Parameter records for articulated parts, attached parts, and other Variable Parameter records
g) Capabilities of the entity, including:
    1) Resupply
    2) Repair

h) Other information to support the management and interaction of this entity with the synthetic environment.

5.3.2.3 Dead reckoning

5.3.2.3.1 Introduction

A method of position/orientation estimation called dead reckoning shall be employed to limit the rate at which Entity State PDUs or Entity State Update are issued. The "dead reckoning algorithm in use" is the dead reckoning algorithm specified in the Dead Reckoning Algorithm subfield of the Dead Reckoning Parameter field of the Entity State PDUs. This algorithm shall be used by the receiving simulation application to perform dead reckoning calculations for the received entity.

5.3.2.3.2 Dead reckoning and the issuing entity

The following rules shall apply to dead reckoning of an entity by the simulation application who owns it:

a) Each simulation application shall maintain two state models of each entity it is representing in support of the dead reckoning process. One model shall be the internal model used by the simulation application to represent its entity. The other model shall be a dead reckoning model of the entity it is representing. Certain thresholds shall be established as criteria for determining whether the entity's actual position/orientation has varied an allowable amount from its dead reckoned position/orientation.

b) When the entity's actual position/orientation has varied from the dead reckoned position/orientation by more than a threshold value, the simulation application shall issue an Entity State or Entity State Update PDU to communicate the entity's actual position and orientation to other simulation applications. The simulation application shall also use the same information communicated to other simulation applications to update its dead reckoning model of that entity.

c) The default method of dead reckoning error calculation shall be a threshold change in the position or orientation. The position threshold parameter shall be identified by the symbolic name DRA_POS_THRSH, and the orientation threshold parameter by DRA_ORIENT_THRSH. (See 6.1.8 for parameter details and default values.) The positional difference shall be calculated in world coordinates using the difference between actual entity location and the dead reckoned location. The position threshold applies to each axis (*X*, *Y*, and *Z*) independently. The orientation threshold shall be calculated using the difference between the actual entity orientation and the dead reckoned orientation as described in E.7.5. In the event that an entity is using the static dead reckoning model and the entity's position/orientation changes, the position threshold shall apply to the difference between the actual entity position and the previously issued position and the orientation threshold shall apply to the difference between the actual entity orientation and the previously issued orientation.

d) Different dead reckoning position and orientation threshold values may be applied by a simulation application to platforms of different domains (e.g., air platform entity) and different kinds of entities (e.g., sensor/emitter entity).

e) A simulation application may change the dead reckoning algorithm in use during the life of an entity as often as required.

f) The 120-bit Other Parameters subfield of the Dead Reckoning Parameter field of the Entity State PDU may be used to provide supplemental data for dead reckoning algorithms other than those specified in Annex E. Any use of such data would normally be included in an exercise agreement. If no additional dead reckoning algorithms are defined, the issuing simulation should put orientation information in the Other Parameters field as described in E.8.2.

g) Any incorporation of a dead reckoning algorithm other than the one specified in Annex E that is required or allowed by an exercise agreement shall be assigned a dead reckoning algorithm number other than a number that is currently assigned to a required algorithm as shown in Table E.1 even if the algorithm is a variation of a required algorithm.

5.3.2.3.3 Dead reckoning and the receiving entity

The following rules shall apply to the dead reckoning of a remote entity received by a simulation application:

a) Each simulation application shall maintain a dead reckoning model of the position/orientation of entities that are of interest (e.g., within sight or range) using the dead reckoning model specified in the Dead Reckoning Algorithm subfield of the Dead Reckoning Parameters field of an Entity State PDU. The dead reckoned position/orientation of these entities shall be used by the simulation's application as required for its execution.

b) Each simulation application shall implement all the dead reckoning algorithms specified in Annex E for receipt and processing of remote entities.

c) When a simulation application receives a new update for a remote entity it is dead reckoning, it shall correct its dead reckoning model and base its estimations on the most recent position/orientation, velocity, and acceleration information. Smoothing techniques may be used to eliminate jumps that occur in, for example, a visual display when the dead reckoning position/orientation of an entity is corrected using more recent position/orientation data.

d) A receiving simulation application shall check the Dead Reckoning Algorithm subfield of the Dead Reckoning Parameter field in the Entity State PDU whenever it receives one of those PDUs for an entity that it processes to determine whether the algorithm has changed. If the algorithm has changed, the receiving simulation shall use that algorithm in place of the previous one to perform dead reckoning of the remote entity.

5.3.2.3.4 Dead reckoning algorithms

Required dead reckoning algorithms for use with this standard are included in Annex E. To allow for parameters associated with the dead reckoning algorithm in use, a field has been set aside in the Entity State PDU for dead reckoning parameters.

5.3.2.3.5 Dead reckoning of frozen entities

An entity with the Entity Appearance record Is Frozen field (bit 21) set to Frozen (1) is considered frozen. A simulation that receives an Entity State PDU from a frozen entity shall not continue dead reckoning extrapolation for that entity. See 5.6.5.5.3 for additional information about the internal state of a frozen entity.

5.3.2.4 Issuance of the Entity State PDU

A simulation shall issue an Entity State PDU when any of the following occur:

a) The discrepancy between an entity's actual state (as determined by its own internal model) and its dead reckoned state (state using specified dead reckoning algorithms) exceeds a predetermined threshold (see Annex E concerning dead reckoning and 5.3.2.3.2 concerning threshold values). This threshold includes changes in position/orientation information and articulated part parameter information (see I.2.2.7).

b) A change in the entity's appearance occurs, for instance, if the entity begins to burn or emit smoke.

c) A predetermined length of real-world time has elapsed since the issuing of the last Entity State PDU. The Entity State PDU heartbeat timer parameter and tolerance shall be identified by the symbolic names for each entity kind and platform and for a stationary entity. (See 6.1.8 for parameter details and default values.)

d) The entity ceases to exist in the synthetic environment. A *final* Entity State PDU shall be issued with the Entity Appearance record State field (bit 23) set to Deactivated (1).

e) A change in the entity's capabilities.

f) A change in the dead reckoning algorithm used by the simulation application.

Simulation applications shall issue an *initial* Entity State PDU before other PDUs that provide supplemental data for the entity. When an Entity State or Entity State Update PDU is no longer being sent for an entity, any associated supplemental PDUs (e.g., IFF PDU and EE PDU) shall also cease to be transmitted.

The Entity State PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.3.2.5 Receipt of the Entity State PDU

Upon receipt of the Entity State PDU, a simulation application shall determine whether the PDU contains more current information than that currently being used to model the transmitting entity. If so, the simulation application shall use the information contained therein to model the position, orientation, and appearance of the entity that issued the PDU. Otherwise the PDU shall be discarded.

If the appearance of an entity indicates it is in the Deactivated state or a predetermined length of reference time has elapsed since any entity's last Entity State PDU, then all simulations shall remove that entity from the exercise. The timeout shall be established at exercise start, although it may be changed during the exercise. The entity timeout parameter shall be computed as shown in NOTE 2 of Table 25.

5.3.2.6 Guise function

5.3.2.6.1 General

The purpose of the Guise function is to allow DIS participants to serve in the opposing force role while operating friendly simulations. This function allows both sides of an engagement to see their team members as friendly and the opposing forces as hostile. The Guise function only affects the depiction of the other force, not any of the internal characteristics of its simulation or behavior.

The Entity Type field shall be used by members of the issuing entity's team to display the issuing entity. The Alternate Entity Type field shall be used by members of all other teams to display the issuing entity. The force identification determines the team membership of the issuing entity.

NOTE—The location in the Entity State PDU is the origin of the Entity Type model, not the origin of the Alternate Entity Type model. For vehicles in contact with the ground, it will generally be necessary to shift the model for the Alternate Type entity in order to keep it in contact with the ground. This shift is required because the height of the center of the bounding volumes of the Entity Type and Alternate Entity Type models will generally be different.

5.3.2.6.2 Issuing entity actions

If the Guise function is used, an issuing entity shall perform the following actions:

a) Entity shall assign the entity type code for its friendly force appearance to the Entity Type field of the Entity State PDUs it issues.

b) Entity shall assign the entity type code for its opposing force appearance to the Alternate Entity Type field of the Entity State PDUs it issues.

c) In the case of neutral entities, the same type code shall be assigned to both the Entity Type field and the Alternate Entity Type field.

If the Guise function is not used, the same entity type shall be assigned to both the Entity Type field and the Alternate Entity Type field.

5.3.2.6.3 Receiving entity actions

The receiving entity shall interpret the Entity Type field and Alternate Entity Type field. Upon receiving an Entity State PDU from an entity that is of interest, the receiving entity shall check the value of the force identification of the entity that issued the PDU. The following conditions apply:

a) If the force identification is the same as that of the receiving entity, the Entity Type field shall be used to display the issuing entity (so that its friendly force appearance will be shown if Guises are in use).

b) If the force identification is different from that of the receiving entity, the Alternate Entity Type field shall be used to display the issuing entity (so that its opposing force appearance will be shown if Guises are in use).

5.3.2.6.4 Example of Guise function

The Guise function might be employed in the following manner:

> In a simulation exercise, the friendly force appearance is an M1 tank and the opposing force appearance is a T-72. All participants would use an entity type of M1 and an alternate entity type of T-72 in the Entity State PDUs they issue.

During the exercise, members of Team 1 receiving Entity State PDUs from their own team will depict them using the friendly force appearance M1. Entity State PDUs received from members of Team 2 will be depicted by members of Team 1 as T-72s by using the opposing force appearance.

Likewise, members of Team 2 receiving Entity State PDUs from their own team will depict them using the friendly force appearance M1. Entity State PDUs received from members of Team 1 will be depicted by members of Team 2 as T-72s.

5.3.3 Collision PDU

5.3.3.1 Purpose

The Collision PDU shall be used to communicate information about a collision between two simulated entities or between a simulated entity and another object in the simulated world (e.g., a cultural feature such as a bridge or building).

5.3.3.2 Collision event

A collision shall be defined as an event that occurs when all of the following conditions are true:

a) Two simulated entities intersect (one of which may be a terrain object).

b) Distance between the origins of the two simulated entities is decreasing.

c) At least one of the entities is moving at a speed greater than COLLISION_THRSH.

5.3.3.3 Information contained in the Collision PDU

The Collision PDU shall contain the following information:

a) Identification of the entity that issued the PDU.

b) Identification of the entity that the issuing entity collided with.

c) Event identification of the specific event marked by the collision of the entities.

d) Information for damage determination. This information, when available, shall be used by each entity to determine the extent of damage received during the collision, including:

   1) Velocity vector of the issuing entity
   2) Mass of the issuing entity
   3) Location of impact in entity coordinates of the entity that the issuing entity collided with

   Any of these three fields may be set to zero if data required for that field cannot be determined by the issuing entity.

e) Information identifying whether the collision should be modeled as an elastic or inelastic type collision.

5.3.3.4 Issuance of the Collision PDU

The Collision PDU shall be issued by an entity when a collision is detected between the issuing entity and an object or some other entity taking part in the simulation exercise. If the collision involves two entities, both entities shall issue the Collision PDU even if only one of them detected the collision. An entity that receives a Collision PDU indicating another entity has collided with it without first detecting such a collision shall issue a Collision PDU naming the entity that issued the first Collision PDU. If the entity subsequently detects the same collision event, it shall not generate a Collision PDU to report it. When a simulation application receives a Collision PDU naming an entity it simulates as the other party involved in a collision after reporting the same collision event, it shall not send another Collision PDU in response. A simulation application shall always issue one, and only one, Collision PDU per collision event it detects or is informed about in which an entity it simulates is a participant, even if that application does not perform collision detection tests.

The following field-specific requirements apply:

a) *Colliding Entity ID*. If the entity with which the issuing entity collided is unknown, this field shall contain ENTITY_ID_UNKNOWN.
b) There are no additional specific field requirements identified at this time.

The Collision PDU shall be issued by using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.3.3.5 Receipt of the Collision PDU

Upon receipt of the Collision PDU, the data contained therein shall be used to record the event and to determine the extent of the damage sustained in the collision.

5.3.4 Collision-Elastic PDU

5.3.4.1 Purpose

The Collision-Elastic PDU provides a mechanism for introducing high-fidelity interactions in DIS exercises. The Collision-Elastic PDU is used to communicate information about an elastic collision between two simulated entities that accounts for:

a) Linear and rotational momentum transfer
b) Variable elasticity
c) Momentum transfer that is dependent on surface orientation

Each Collision-Elastic PDU describes the information required to compute half of a high-fidelity collision. The Collision-Elastic PDU can be a complete replacement for the Collision PDU (see 5.3.3 and 7.2.3) by setting to zero all fields after the Location of Impact field in the Collision-Elastic PDU (see Table 136).

5.3.4.2 Collision-elastic event initiation

The following shall constitute a collision detection:

a) Receipt of a Collision-Elastic PDU from an entity with whom a state of collision does not currently exist.

b) Detection of an intersection with an entity with whom a state of collision does not currently exist.

A state of collision is any state other than the Rest state in the finite state machine described in 5.3.4.9.

5.3.4.3 Collision-elastic event termination

Collision events may be terminated in the following ways:

a) An expected Collision-Elastic PDU response is not received after a simulation time interval of COLLISION_ELASTIC_TIMEOUT.

b) After receiving a Collision-Elastic PDU and after the collision is simulated, the two entities are separated.

If the entities continue to intersect, entities are entitled to move so that they do not intersect in order to reset the collision finite state machine (see 5.3.4.9).

5.3.4.4 Algorithm for collision-elastic computation

The following steps outline how to compute the elastic collision event:

a) Compute the contact velocity $\vec{v}_c$, which is the local velocity of the entities at the point of contact.

$$\vec{v}_{1c} = \vec{v}_1 + \vec{w}_1 \times \vec{r}_1$$

where

$\vec{v}$ is the linear velocity of the entity

$\vec{w}$ is the angular velocity of the entity

$\vec{r}$ is the moment arm generated from the arbitrated location of impact

All parameters are in world coordinates.

b) Transform the own vehicle inverse tensor of inertia into world coordinates in order to represent it appropriately for use in the collision equations. (If a vector $\vec{s}$ in body coordinates is represented in world coordinates via the transformation $\underline{T}\vec{s}$, then $\underline{TI^{-1}}\,\underline{T}^T$ is the representation of the inverse tensor of inertia in world coordinates.) Compute a matrix, which is denoted by $\underline{\underline{Q}}_1$ called the Collision Intermediate Result by $\underline{L^T_{\vec{r}_1} I_1^{-1} L_{\vec{r}_1}}$, where $\underline{L}_{\vec{r}_1} = [\vec{r}_1 X] = \begin{bmatrix} 0 & -r_1[3] & r_1[2] \\ r_1[3] & 0 & -r_1[1] \\ -r_1[2] & r_1[1] & 0 \end{bmatrix}$, i.e., the matrix operator that results in computing the cross product with $\vec{r}_1$.

c) Send η_1 the coefficient of restitution, m_1 the mass, l_1 the location of the point of contact, $\hat{n}_1$ the unit surface normal, $\underline{Q}_1$ the Collision Intermediate Result, and $\vec{v}_{1c}$ the contact velocity in a Collision-Elastic PDU. Receive η_2 the coefficient of restitution, m_2 the mass, l_2 the location of the point of contact, $\hat{n}_2$ the unit surface normal, $\underline{Q}_2$ the Collision Intermediate Result, and $\vec{v}_{2c}$ the contact velocity in a Collision-Elastic PDU.

d) Arbitrate the data as described in 5.3.4.8.

e) Solve for the momentum transfer given by $\kappa = -\dfrac{(1+\eta)(\vec{v}_{c1} - \vec{v}_{c2}) \times \hat{n}}{\dfrac{1}{m_1} + \dfrac{1}{m_2} + \hat{n}^T \underline{Q}_1 \hat{n} + \hat{n}^T \underline{Q}_2 \hat{n}}$, where η and $\hat{n}$ are the arbitrated values of the coefficient of restitution and unit surface normal, respectively.

f) Substitute the momentum transfer into the following velocity update equations: $\vec{v}'_1 = \vec{v}_1 + \hat{n}\kappa / m_1$ and $\vec{w}'_1 = \vec{w}_1 + \underline{I}^{-1}_1 (\vec{r} \times \hat{n})\kappa$, where I is the tensor of the own vehicle in world coordinates.

5.3.4.5 Information contained in a Collision-Elastic PDU

The Collision-Elastic PDU shall contain the following information:

a) Identity of the entity that issued the Collision-Elastic PDU.

b) Identification of the entity that the issuing entity collided with. (If this information is unknown, the Identification field shall contain ENTITY_ID_UNKNOWN.)

c) Event identification of the specific event marked by the collision of the entities.

d) Velocity of the issuing entity at the point of contact, to be distinguished from the velocity of the center of mass.

e) Mass of the issuing entity.

f) Location of impact in entity coordinates of the entity that the issuing entity collided with.

g) Six-member array of the upper triangular part of the symmetric matrix called the Collision Intermediate Result (CIR). The members of the array are given by CIR_{XX}, CIR_{XY}, CIR_{XZ}, CIR_{YY}, CIR_{YZ}, and CIR_{ZZ}. The CIR is represented by the matrix product $\underline{L^T_{\vec{r}_1} I_1^{-1} L_{\vec{r}_1}}$, where $\underline{I_1}$ is the tensor of inertia and $\underline{L^T_{\vec{r}_1}}$ is given by $\underline{L_{\vec{r}_1}} = [\vec{r}_1 X] = \begin{bmatrix} 0 & -r_1[3] & r_1[2] \\ r_1[3] & 0 & -r_1[1] \\ -r_1[2] & r_1[1] & 0 \end{bmatrix}$.

h) Unit surface normal at the point of contact in world coordinates.

i) Coefficient of restitution, which represents a tunable parameter in the collision equation.

5.3.4.6 Issuance of the Collision-Elastic PDU

A Collision-Elastic PDU shall be issued by an entity when a collision is detected between the issuing entity and another entity with whom the issuing entity is not already engaged in a collision simulation. Only after termination of the prior collision simulation may the next collision simulation begin. See 5.3.4.2 for the conditions defining a collision detection. An entity that receives a Collision-Elastic PDU indicating another entity has collided with it without first detecting the collision shall issue a Collision-Elastic PDU naming the entity that issued the first Collision-Elastic PDU. If the entity subsequently detects the same collision event, it shall not generate a Collision-Elastic PDU to report it. When a Newtonian simulation application receives a Collision-Elastic PDU naming an entity it simulates as the other party involved in a Newtonian collision after reporting the same collision event, it shall not send another Collision-Elastic PDU in response. A

Newtonian collision simulation application shall always issue one, and only one, Collision-Elastic PDU per collision event it detects or is informed about in which an entity it simulates is a participant, even if that application does not perform collision detection tests.

The Collision-Elastic PDU shall never be issued in response to the Collision PDU specified in 5.3.3.

The Collision-Elastic PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.3.4.7 Receipt of the Collision-Elastic PDU

Entities that are not prepared to engage in a Newtonian collision simulation shall ignore the Collision-Elastic PDU.

Entities capable of performing Newtonian collision simulations shall simulate the collision upon receipt of the PDU. If a Collision-Elastic PDU has not already been sent, then the recipient of the Collision-Elastic PDU shall respond by issuing a Collision-Elastic PDU. The simulation shall then wait for the conditions necessary to terminate a collision. See 5.3.4.3 for the conditions defining the termination of a collision event.

5.3.4.8 Arbitration of collision parameters

Both the coefficient of restitution η and the unit surface normal $\hat{n}$ shall be arbitrated using the following predetermined arbitration functions:

$$\hat{n} = \frac{\hat{n}_1 - \hat{n}_2}{\|\hat{n}_1 - \hat{n}_2\|}$$

$$\eta = \min(\eta_1, \eta_2)$$

The location of impact shall be arbitrated by converting the location of impact in the received Collision-Elastic PDU into world coordinates. The arithmetic mean shall then be computed. This location of impact shall then be used to recompute the moment arm $\vec{r}$ used in the velocity update equations.

It is recommended that collisions always have elasticity to result in the ultimate separation of the collision entities, thereby guaranteeing that the collision detection finite state machine terminates.

Often the receiver of a Collision-Elastic PDU cannot detect a collision via geometric considerations. The variables necessary to compute a collision may not be accessible. The appropriate variable needs to be synthesized for the response. The recommended approach is to accept all data that would otherwise be arbitrated and format them to be sent in the Collision-Elastic PDU response as follows:

a) Coefficient of restitution shall be retransmitted as it is received.
b) Unit surface normal, which is received in world coordinates, shall be negated and retransmitted.
c) Location of impact is received in own body coordinates. This result shall be translated to the body coordinate of the remote vehicle and retransmitted.
d) Collision intermediate result and the contact velocity both rely on the moment arm, which shall be computed from the location of impact.

5.3.4.9 State information for Newtonian collision simulations

The following paragraphs describe the different states and transitions for Newtonian collision simulations.

Each entity may be in one of four states:

a) *Rest State.* A receiving entity is in the Rest state before interacting with a given entity.

b) *PDU Sent State.* An entity is in the PDU Sent state when a collision is detected without having received a Collision-Elastic PDU.

c) *Simulate State.* This transitory state occurs when a Collision-Elastic PDU is received. Computation and distribution of new kinematic variables occurs here. Damage models are also computed while in this state.

d) *Wait State.* The kinematics of the collision have been simulated and the entity is waiting for the conditions that terminate the collision. One may choose other mechanisms than the collision results to terminate the collision.

The behavior of the Newtonian collision simulation is shown in Figure 5. The associated state transitions are described in Table 6.

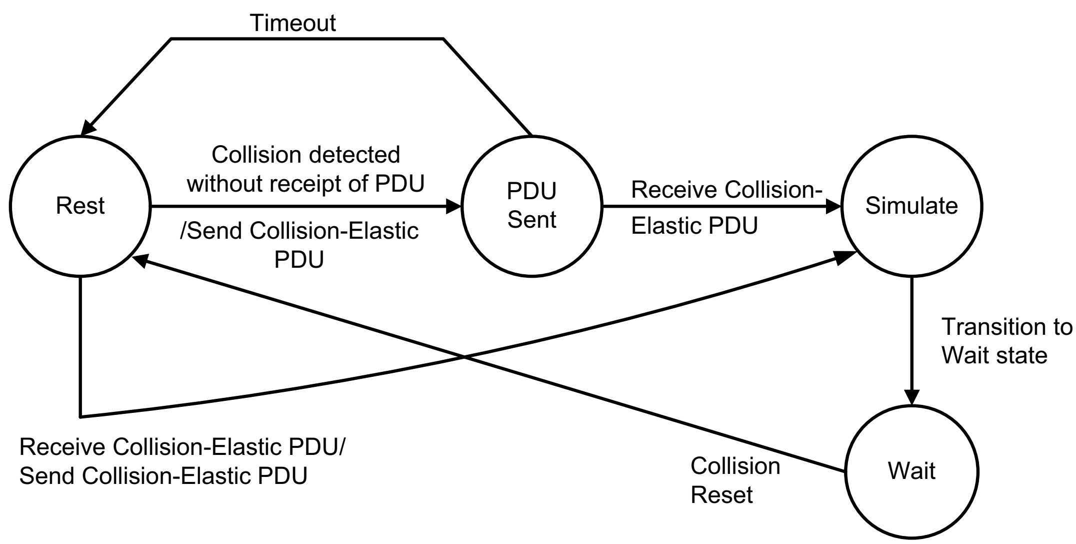


Figure 5—Newtonian collision finite state machine

Table 6—Newtonian collision simulation state transition

Transition	Conditions and actions
Collision detected without receipt of Collision-Elastic PDU/Send Collision-Elastic PDU	This transition is only relevant in the Rest state. When this condition is relevant, a Collision-Elastic PDU shall be issued. The entity shall then proceed to the PDU Sent state.
Receive Collision-Elastic PDU	When a Collision-Elastic PDU is received while the entity is in the PDU Sent state, the entity shall transition to the Simulate state.
Time Out	This transition is only valid in the PDU Sent state. If the entity is in this state for a time greater than COLLISION_ELASTIC_TIMEOUT, then the finite state machine shall reset and transition to the Rest state.
Transition to Wait state	This transition occurs immediately after being in the transitional Simulate state. No condition is required to transition to the Wait state from the Simulate state.
Collision Reset	This transition is only relevant in the Wait state. When the entities have separated, the entity shall return to the Rest state.
Receive Collision-Elastic PDU/ Send Collision-Elastic PDU	When a Collision-Elastic PDU is received in the Rest state, a Collision-Elastic PDU shall be issued. The entity shall then proceed to the Simulate state.

5.3.5 Entity State Update PDU

5.3.5.1 Purpose

The Entity State Update PDU is a network bandwidth-reducing extension of the basic Entity State PDU and may be used to communicate specific nonstatic information about an entity's state. The Entity State Update PDU should not serve as a complete replacement for the Entity State PDUs (see 5.3.2) because the update PDU does not convey an entity's static and nearly static information.

5.3.5.2 Information contained in an Entity State Update PDU

The Entity State Update PDU shall contain the following information:

a) Identification of the entity that issued the PDU
b) Information about the location of the entity in the simulated world and its orientation, including:
    1) Velocity
    2) Location with respect to the world
    3) Orientation
c) Information required for representation of the entity's appearance, including:
    1) Number of Variable Parameter Records and the parameter values (e.g., articulated and attached parts)
    2) Appearance of the entity (e.g., normal, smoking, on fire, or producing a dust cloud)
    3) Presence (including types and numbers) of attached parts or stores

5.3.5.3 Issuance of the Entity State Update PDU

A simulation shall issue an Entity State Update PDU when any of the following occur:

a) Discrepancy between an entity's actual state (as determined by its own internal model) and its dead reckoned state (state using the dead reckoning algorithm; see Annex E, specified in the Entity State PDU) exceeds a predetermined threshold (see 5.3.2.3 concerning threshold values). This threshold includes changes in position/orientation information and articulated part parameter information (see I.2.2.7).

b) A change in the entity's appearance occurs.

c) A predetermined length of real-world time has elapsed since the issuing of the last Entity State or Entity State Update PDU. The Entity State Update PDU heartbeat timer parameter and tolerance shall be identified by the symbolic names for each entity kind and platform and for a stationary entity. (See 6.1.8 for parameter details and default values.)

d) Entity ceases to exist in the synthetic environment. A *final* Entity State PDU or Entity State Update PDU shall be issued to indicate that the entity is in the Deactivated state.

The Entity State Update PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.3.5.4 Receipt of the Entity State Update PDU

Upon receipt of an Entity State Update PDU, a simulation application shall determine whether the PDU contains more current information than that currently being used to model the transmitting entity. If so, the simulation application shall use the information contained therein to model the position, orientation, and appearance (if present) of the entity that issued the PDU. Otherwise, the PDU shall be discarded.

If the appearance of an entity indicates it is in the Deactivated state or a predetermined simulation time interval has elapsed since any entity's last Entity State or Entity State Update PDU, then all simulations shall remove that entity from the exercise. See 4.2.7 for timeout parameters and their default values.

5.3.6 Attribute PDU

5.3.6.1 Purpose

The Attribute PDU has the following primary functions:

a) PDU Extension—A means to extend a PDU with additional records containing information not defined in the PDU itself. It can be used to extend PDUs such as the EE and Fire PDUs but generally would not be applicable to PDUs that have a Standard Variable Specification record or Datum Specification record section, e.g., the Transmitter and Set Data PDUs. When extending a State PDU, the Attribute PDU may be bundled with the PDU it extends or sent independently. A transient PDU can be extended only by bundling an Attribute PDU with it. State PDUs are those listed in Table 4 Heartbeat timers (see 4.2.6.1).

b) The Attribute PDU may also be used to communicate attributes that are not associated with a specific PDU type. See item c) in 5.3.6.3. For example, there are no PDUs for passive sensors because they emit no energy and cannot be externally detected. However, the internal details of a passive sensor could be useful for analytical or training purposes. In those cases, Attribute Records in an Attribute PDU can be used to convey the sensor details without creating a custom PDU.

The Attribute PDU shall not be used to exchange data available in any other PDU except where explicitly mentioned in the PDU issuance instructions within this standard.

The general Attribute record format is specified in 6.2.10.

5.3.6.2 Information contained in the Attribute PDU

The Attribute PDU shall contain the following information:

a) Identification of the originating simulation
b) Identification of the PDU type to which the Attribute records apply, if applicable
c) Identification of the Protocol Version of the PDU type, if present
d) Identification of the Master Attribute Record Type, if only one record type appears in all Attribute records
e) An indication of the action that this Attribute PDU represents
f) One or more Attribute Record Sets, one Entity or Object ID per set

Each Attribute Record Set shall contain the following information:

— The Entity or Object ID to which all Attribute records in the set apply
— One or more Attribute records

5.3.6.3 Issuance of the Attribute PDU

The issuance rules specified here are general rules without regard to specific Attribute records. Specific Attribute records may have additional issuance rules. A simulation shall issue an Attribute PDU as specified herein and as required by the rules associated with the specific Attribute record type:

a) *General Rules*
   1) An issuing simulation may collect multiple Attribute records into an Attribute PDU for one or more entities or other objects. In doing so, the simulation shall not delay the issuance of the Attribute PDU to an extent that affects interoperability requirements.
   2) Field-Specific Requirements
      i) Attribute Record PDU Type. If the Attribute records in the PDU extend or update a specific PDU type, this field shall represent that PDU type. All Attribute records in the PDU shall apply to this type of PDU. The PDU Extension rules stated below shall apply. If the attribute information does not pertain to any PDU type, this field shall be set to Other (0) and the rules for Other Attributes stated below shall apply.
      ii) Attribute Record Protocol Version. This field shall be set to indicate the Protocol Version of the PDU being extended or updated.
      iii) Master Attribute Record Type. If a single Attribute record type applies to every Attribute record in this PDU, the value of the Master Attribute Record Type field shall be set to indicate the one Attribute record type for all attribute records. In that case, the Attribute Record Type field of every Attribute record shall contain the same value as the Master Attribute Record Type. This field value shall be set to Multiples Present parameter (symbolic name MULTIPLES_PRESENT) if multiple Attribute record types are represented in this Attribute PDU. (See 6.1.8 for the symbolic name value.)
      iv) Attribute Record Sets. Each Attribute PDU shall contain one or more Attribute Record Sets. Each Attribute Record Set shall contain one or more Attribute records. An Attribute Record Set shall include only those Attribute records that apply to the Entity ID or Object ID specified for that set. For example, Attribute records that do not pertain to any specific entity (i.e., have an Entity ID of NO_SITE: NO_APPLIC: NO_ENTITY) cannot be mixed in the same Attribute Record Set with Attribute records that do pertain to a specific entity.

b) *PDU Extension Rules*. The Attribute PDU may be used to extend another PDU or multiple PDUs of the same PDU type with information beyond that defined in the PDU itself. The use of the Attribute PDU in this manner is accomplished with either a Coupled PDU Extension or a Noncoupled PDU Extension:

1) Coupled PDU Extension uses PDU bundling and an indicator in the PDU Status field so that the pair of coupled PDUs are not separated in transit and are delivered to the receiving simulation together. Rules for Coupled PDU Extension are as follows:

   i) The Coupled PDU Extension shall be used to extend only one instance of a PDU. The Attribute PDU shall contain Attribute records that pertain only to the one PDU instance being extended.

   ii) The PDU Status field Coupled Extension Indicator (bit 3) of the PDU being extended shall be set to Coupled (1).

   iii) The extending Attribute PDU shall immediately follow the PDU being extended in a PDU bundle (see 6.3.5). These two PDUs form the "couple." No PDU type other than an Attribute PDU shall be allowed to follow a PDU whose Coupled Extension Indicator is set to Coupled (1).

   iv) An Attribute PDU shall not be extended with another Attribute PDU. The PDU Status field Coupled Extension Indicator (bit 3) shall be set to Not Coupled (0) in an Attribute PDU.

   v) The pair of coupled PDUs may be included in a larger PDU bundle in accordance with the bundling rules referenced in 6.3.5.

   vi) The total length of the coupled PDUs shall not exceed MAX_PDU_SIZE_OCTETS.

2) An Attribute PDU is not coupled when the PDU Status field Coupled Extension Indicator (bit 3) is set to Not Coupled (0) in the PDU being extended. Noncoupled PDU Extension is allowed for state PDUs only:

   i) Noncoupled PDU Extension may be used to extend a single PDU or multiple PDUs of the same PDU type, including attributes for multiple entities and other objects.

   ii) In Noncoupled PDU Extension, the extending Attribute PDU may be sent independently of the state PDU or PDUs being extended.

   iii) While the Attribute PDU may appear in a bundle with the state PDU or PDUs it extends, that alone does not constitute a Coupled PDU Extension.

3) Extending transient PDUs

   i) Only Coupled PDU Extension shall be used to extend a transient PDU.

   ii) The Attribute PDU shall contain Attribute records that pertain only to the transient PDU being extended.

4) Extending state PDUs

   i) State PDUs may be extended with either Coupled or Noncoupled PDU Extension.

   ii) When using Coupled PDU Extension, the Attribute PDU shall contain Attribute records that pertain only to the state PDU being extended.

   iii) An Attribute record shall have the same heartbeat period as that specified for the state PDU it extends. Figure shows an example of the timing of a Noncoupled Attribute PDU and the state PDU it extends. Note how both have a 10 s heartbeat period.

   iv) It is not required that a particular Attribute record be included in every issuance of Attribute PDUs for its associated entity. At a minimum, an Attribute record shall be issued when its contents change or to satisfy the appropriate heartbeat requirement.

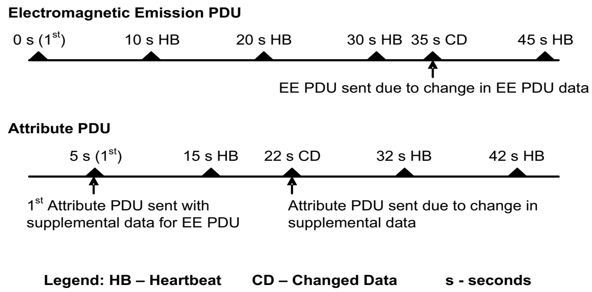


Figure 6—Attribute PDU example

5) Regardless of the type of PDU extension used, the Attribute Record PDU Type and Attribute Record Protocol Version fields in the Attribute PDU shall be set to match the PDU Type and Protocol Version of the PDU or PDUs that are being extended.

c) *Other Attributes*. The Attribute PDU may also be used to communicate attributes that are not associated with a specific PDU type. A specific PDU type is defined as one that has been assigned a nonzero PDU Type enumeration value. If used in this manner, the following rules apply:
   1) The Attribute Record PDU Type and Attribute Record Protocol Version fields shall be set to Other (0).
   2) The Attribute PDU shall not contain Attribute records that pertain to existing PDU types.
   3) Issuance and heartbeat rules for the specific Attribute records shall apply.

5.3.6.4 Receipt of the Attribute PDU

The receipt rules specified here are general rules without regard to specific Attribute records. Specific Attribute records will have additional receipt rules. Upon receipt of the Attribute PDU, the data contained therein shall be used to initialize or update the corresponding data for the specified entity or other object.

Gateways or intermediate software shall not separate a coupled Attribute PDU from the PDU it extends if both PDUs are being forwarded. If the Attribute PDU is being discarded, then the PDU Status field Coupled Extension Indicator (bit 3) of the extended PDU shall be set to Not Coupled (0) before being forwarded.

5.4 Warfare

5.4.1 General

This subclause describes the requirements related to weapons, expendables, and to any type of explosion whether or not it is related to munitions. The general requirements related to the issuance and receipt of Fire and Detonation PDUs are listed here. References to Entity State PDU shall also mean the Entity State Update PDU unless otherwise indicated. The terms “lethality model” and “damage assessment model” are used interchangeably in this subclause.

5.4.2 General requirements

The following general requirements shall apply:

a) *Weapons*. Weapons include both munitions such as bombs, bullets, artillery shells, land or maritime mines, missiles, directed energy beams, and any other object used as a weapon such as a rock or arrow when released on a trajectory to an intended target. The term “weapon” shall include munitions and any other objects used in a manner to inflict intentional damage. A weapon detonation shall be able to be traced back to an originating weapon entity or firing entity. Weapons may be represented by an entity or other physical object such as a mine object that is part of a minefield. The source of a weapon firing shall be indicated by the Entity ID of the firing entity. A projectile originating from a weapon, such as a missile, may also be an entity or its existence may only be represented by the Fire and Detonation PDU. An entity state is not required for a projectile originating from a weapon when the information necessary to model a ballistic trajectory is available in the Fire and Detonate PDU.

b) *Expendables*. Expendables are of two types:
   1) Countermeasure Expendables. These include chaff, flares, and active and passive decoys including both towed and air-launched standoff decoys.
   2) Non-countermeasure Expendables. These include flares, such as signal and illumination flares, and other pyrotechnics that are not used in a countermeasure role.

   An expendable shall always require the issuance of a Fire PDU and will include the issuance of a Detonation when applicable. In all cases, an expendable entity or event shall be able to be traced back to an entity that released the expendable. An expendable, when initially released, shall always originate from an entity for which an Entity State PDU has been issued. An expendable, except for a towed or air-launched standoff decoy, may be represented by only a Fire and Detonation PDU. Towed or air-launched standoff decoys shall be represented by an entity.

   The Environmental Process PDU may be used to provide chaff characteristics once the chaff is dispensed. It should be used if curtain chaff is being dispensed.

c) *Explosions*. Explosions include any type of explosion other than a munition that explodes, for example, a fuel drum or internal ammunition that explodes in a fire. A simulation that determines that one of its entities, or part of an entity, has exploded shall issue a Detonation PDU to reflect the explosion regardless of whether the explosion was caused by a munition.

d) *Detectable Weapons and Expendables*. If a weapon or expendable is required to be detected or tracked visually or by other sensors, and the information necessary to model a ballistic trajectory is not available in the Fire and Detonate PDU, an entity shall be created for the weapon or expendable.

e) Single- and Multiple-Round Munitions
   1) If the firing of the weapon represents a single round, the Quantity field of the Munition Descriptor record in the Fire PDU shall contain a value of one, and the Rate field shall contain a value of zero.
   2) If the firing of a weapon represents multiple rounds, the Quantity field of the Munition Descriptor record in the Fire PDU shall contain the quantity of the munition fired and the Rate field shall indicate the rate at which it was fired.

f) *Damage Assessment*. The simulation that owns an entity shall be responsible for determining the damage to one of its entities caused by a Detonation PDU. The following requirements shall apply:
   1) A simulation, as a minimum, shall be required to perform damage assessment based on Detonation PDUs unless all of the following conditions are satisfied:
      i) The simulation supports the reception of Directed Energy (DE) Fire PDUs
      ii) The Detonation PDU munition type is a DE munition
      iii) The Detonation PDU event ID matches a previously issued DE Fire PDU event ID

      In these circumstances, the simulation shall ignore the Detonation PDU and assess damage from the DE Fire PDU.
   2) A simulation, as a minimum, shall be required to determine whether a target entity contained in the Target Entity ID field is one of its local entities and, if so, to perform a damage assessment.
   3) Damage assessment based on the receipt of a Detonation PDU shall include, as a minimum, the use of data from the following fields:
      i) Detonation Result

ii) Target Entity ID

iii) Descriptor

iv) Location in Entity's Coordinates if a Target Entity ID is specified

v) Location in World Coordinates

g) *Fire and Detonation PDU Relationship*. The following relationship shall exist between the Fire and Detonation PDU:

1) A weapon that is initially launched from another entity shall have a Fire PDU and at least one Detonation PDU associated with it unless the weapon is an entity that is launched and then recovered for reuse because it was not detonated.

2) Multiple Detonation PDUs associated with a single Fire PDU, such as submunitions or chaff expendables launched from a single canister, shall be allowed.

3) A Detonation PDU without a preceding Fire PDU shall be allowed such as the detonation of land mines and non-munition explosions [see item a9) in 5.4.3.3].

4) A target entity contained in the Fire PDU shall not be required to match the target entity in the corresponding Detonation PDU.

5) A Fire PDU representing a chaff or flare expendable shall have an associated Detonation PDU to indicate a chaff burst, flare, or pyrotechnic ignition event.

h) Variable Parameter (VP) records may be included in Entity State, Entity State Update, and Detonation PDUs as necessary to support the warfare function. The following VP records are specifically identified in the standard in support of the Warfare function. This does not preclude the use of other VP records as well:

1) Separation (see 6.2.94.6)

2) Entity Type (see 6.2.94.5)

3) Entity Association (see 6.2.94.4)

4) Attached Part (See 6.2.94.3)

5) Articulated Part (see 6.2.94.2)

5.4.3 Fire PDU

5.4.3.1 Purpose

The Fire PDU shall be used to communicate information associated with the firing of a weapon or expendable. This information is used to support visual and other sensors and effects (e.g., aural, seismic, etc.) at the moment the firing occurs. It is also used to identify the entity that fired the weapon or expendable. There is no Fire PDU associated with certain munitions, unintended-use detonations of a munition, and for non-munition explosion [see item a9) of 5.4.3.3].

5.4.3.2 Information contained in the Fire PDU

The Fire PDU shall contain the following information effective at the instant the munition or expendable enters the simulated world:

a) The identifier of the entity firing the weapon or expendable.

b) The entity or object ID that is the intended target when known.

c) The identifier of any munition or expendable entity that is created.

d) The identifier of the specific event marked by this firing of a munition or expendable.

e) Identification of the fire mission if known or applicable.

f) An indication if this firing is for a weapon or expendable.

g) Information required for representation of the path and impact of the munition or expendable, including:
   1) Location from which the munition or expendable was launched or fired.
   2) Type of munition or expendable fired.
   3) Warhead of the munition (if applicable; otherwise, it has a value of zero).
   4) Fuse employed by the munition (if applicable; otherwise, it has a value of zero).
   5) Quantity and rate at which munition was fired.
   6) Initial velocity of the munition or expendable when visible effects of the launch first become apparent.
   7) Range (three dimension, straight-line distance) that the firing entity's fire control system has assumed for computing the fire control solution if a weapon and if the value is known.

The PDU shall be timestamped with the reference time corresponding to the simulation time the munition or expendable clears the firing entity.

5.4.3.3 Issuance of the Fire PDU

The Fire PDU shall be issued at the moment the weapon or expendable is fired.

a) *Firing a Weapon*
   1) All weapons that are fired shall require the issuance of a Fire PDU. If it is a directed energy weapon, and the simulation firing the directed energy weapon has implemented the DE Fire PDU, then the requirements specified in 5.4.5 shall also apply.
   2) If the firing entity represented the munition as an attached part, then an Entity State PDU for the firing entity shall be issued with the Detached Indicator field in the applicable Attached Part VP record set to Detached (1).
   3) If the number of rounds to be fired is not known when the first round clears the firing entity, then the number of committed rounds shall be indicated instead.
   4) Subsequent Fire PDUs shall report the number of rounds in subsequent bursts, or in half-second bursts in the case of continuous fire.
   5) A Separation Variable Parameter (VP) record may be included in the first Entity State PDU for an entity representing a fired object if the exact station that the object was launched from is known.
   6) The requirements associated with a multistage missile and submunitions are further defined in 5.9.6 as part of entity separation requirements.
   7) The PDU Status record Fire Type Indicator (FTI) field (bit 4) shall be set to Munition (0).
   8) The Descriptor field shall contain the Munition Descriptor record (see 6.2.19.2).
   9) A Fire PDU shall not be issued if the munition is one of the following:
      i) A land, aerial, or naval mine.
      ii) An unintended-use detonation (e.g., accidental detonation).
      iii) A demolition event.
      iv) An Explosive Ordnance Disposal (EOD) activity.
      v) An Improvised Explosive Device (IED) unless thrown, launched, or fired by another entity.

b) *Firing an Expendable*
   1) The PDU Status record Fire Type Indicator (FTI) field (bit 4) shall be set to Expendable (1) and other fields shall be set as appropriate.

2) If an entity is created to represent this expendable, an Entity State PDU shall be issued as follows:
   i) If the firing entity represented the expendable in an Attached Part Variable Parameter (VP) record, then appropriate changes shall be made to that record to reflect the firing of that expendable.
   ii) If the simulation that fired the expendable knows the station the expendable was launched from, then a Separation VP record may be included in the first Entity State PDU for the expendable.
   iii) Additional Variable Parameter records may be included to indicate detailed characteristics of an expendable to support visual and sensor models.
   iv) If this is a decoy that will remain attached to the parent entity by a physical connection, or will otherwise be controlled by the parent entity, an Entity Association VP record shall be included in both the parent and decoy Entity State PDUs until the association no longer exists.
   v) If a decoy entity emits countermeasure electromagnetic signals, an EE PDU shall be issued for it.
3) The Descriptor field shall contain the Expendable Descriptor record (see 6.2.19.4).

c) *Non-Munition Explosions*. No Fire PDU shall be issued for a non-munition explosion.
d) *Specific Field Requirements*
   1) Target Entity ID. If the intended target is unknown, this field shall contain TARGET_ID_UNKNOWN.
   2) Fire Mission Index. The Fire Mission Index field is intended for after-action review purposes and may be used to denote a specific assignment given to the weapon system, i.e., the fire mission, with each assignment being given a different value. If used, the Fire Mission Index shall be set to a value other than NO_FIRE_MISSION; otherwise, this field shall contain NO_FIRE_MISSION.
   3) Munition/Expendable Entity ID. If the munition or expendable being fired is not represented as an entity, this field shall contain NO_SPECIFIC_ENTITY.

5.4.3.4 Receipt of the Fire PDU

Upon receipt of a Fire PDU, a simulation shall update its representation of the entity to reflect the information in the PDU. For either a weapon or an expendable, this may be a muzzle/canister flash, noise and accompanying smoke, water perturbation in the case of a torpedo, or other effects.

5.4.4 Detonation PDU

5.4.4.1 Purpose

The Detonation PDU shall be used to communicate information associated with the impact or detonation of a munition, a non-munition explosion, or the burst or ignition of an expendable. The following list is representative:

a) Munitions with or without warheads
b) Kinetic energy projectiles
c) Directed energy result after the weapon stops firing [see item a) in 5.4.4.3]
d) Land, aerial, and naval mines
e) Chaff bursts
f) Flares and other pyrotechnic ignitions or bursts
g) Demolition events

h) Improvised Explosive Devices (IEDs)
i) Explosive Ordnance Disposal (EOD) events
j) The explosion of a munition for other than its intended use

5.4.4.2 Information contained in the Detonation PDU

The Detonation PDU shall contain the following information:

a) *General Requirements*. These are requirements applicable to munitions, expendables, and non-munition explosions:
    1) Identifier for the event
    2) Velocity
    3) Location with respect to the world
    4) Detonation result
    5) Variable parameter records, as applicable
    6) An indication if this detonation is for a munition or a non-munition explosion or expendable.
b) *Munition Detonation*. Information required for representation of the impact or detonation of a munition:
    1) Identifier of the entity that fired this munition, if applicable
    2) Identifier of the target entity if known
    3) Identifier of the munition entity, if applicable
    4) The munition type
    5) Warhead of the munition
    6) Fuse employed by the munition
    7) Quantity of this munition
    8) Rate of fire of the munition
    9) Location with respect to the target entity, if a target entity is present in the Target Entity ID field
c) *Non-Munition Explosion Detonation*. Information required for representation of a non-munition explosion:
    1) Identifier of the entity that was the source of this explosion, if applicable and known
    2) Identifier of the entity, or portion thereof, that exploded
    3) Event identifier
    4) Explosion location in world coordinates
    5) Explosion location in entity coordinates, if applicable
    6) Detonation result
    7) Number of VP records
    8) Description of the explosion characteristics to include:
        i) The type of entity, or component thereof, that exploded
        ii) The type of explosive material
        iii) The explosive force represented as an equivalent amount of kilograms of trinitrotoluene (TNT)
    9) One or more VP records to provide additional information, if required
d) *Expendable Detonation*. Information required for representation of an expendable burst or ignition:
    1) Identifier of the entity that released or launched this expendable

2) Identifier of the target threat entity, if applicable and known, that caused this entity to release a countermeasure expendable
3) Identifier of the expendable entity, if applicable
4) The expendable type

5.4.4.3 Issuance of the Detonation PDU

The Detonation PDU shall be issued by a simulation to indicate a detonation event for a weapon for which a Fire PDU has been issued, a munition that has detonated as a result of its either intended use or unintended use (e.g., accidental detonation), or a non-munition entity or component of the entity has exploded. The Detonation PDU shall also be used to indicate a chaff burst or a flare ignition.

A Detonation PDU shall be issued at the instant in which the detonation occurs. The presence of a Detonation PDU does not necessarily indicate an explosion except when it represents a non-munition explosion. An entity identified as the munition, non-munition explosion, or expendable entity in the Exploding Entity ID field need not be deactivated when the Detonation PDU is issued. Instead, it may remain active with appropriate damage status indicated or the detonation may result in additional entities being created to represent fragments. The following requirements shall be met:

a) *Munition Detonations*
   1) A Detonation PDU shall be issued for a munition or directed energy beam as follows:
      i) When a munition with an explosive warhead explodes.
      ii) When a directed energy engagement has ceased (see 5.4.5.3).
      iii) When a kinetic energy projectile strikes an entity or other object. In this case, no explosion shall be depicted for the kinetic energy projectile entity itself.
      iv) When a munition being modeled impacts the terrain, a terrain feature, or another entity at the moment it detonates.
   2) The location of the impact or detonation shall be specified in the Location in World Coordinates field. If there is a Target Entity ID present, the Location in Entity's Coordinates field shall be set to any offset values if the detonation was offset from the target.
   3) If the simulation issuing the *munition* Detonation PDU has a three-dimensional (3-D) model of the target entity specified by the Target Entity ID field, the simulation shall set the Location in Entity's Coordinates field based on that model. If the detonation occurs upon physical contact with the surface of the 3-D target entity model, then that location shall be used.
   4) A Detonation PDU shall be issued for a munition that is not represented as a separate entity when the simulation modeling the munition determines that one of the following conditions exists:
      i) For a single munition, such as a bomb, the simulation determines that the bomb has reached the end of its active life and would detonate, whether or not it explodes or is a dud.
      ii) For multiple rounds fired, the simulation determines that the first round or a sufficient number of rounds have struck a target such that damage could have been expected to result. Detonation PDUs shall be issued for succeeding rounds as determined by the simulation.
   5) If the simulation issuing a *munition* Detonation PDU determines that the munition has impacted or otherwise affected an articulated part, attached part, or attached articulated part of a target entity, the simulation shall include the affected VP record (with parameter values known at impact) in the Detonation PDU. If the part is not directly connected to the base model, all intervening parts shall be included in the Detonation PDU.

6) The PDU Status record Detonation Type Indicator (DTI) field (bits 4 to 5) shall be set to Munition (0).
7) The Descriptor field shall contain the Munition Descriptor record (see 6.2.19.2).
8) Source Entity ID. This field shall identify the Entity ID of the entity that fired the munition. If the detonation was not preceded by a Fire PDU, then the Source Entity ID shall be set to NO_SPECIFIC_ENTITY.
9) Target Entity ID. This field shall identify the target entity of the munition. If the target Entity ID is unknown, this field shall contain the value TARGET_ID_UNKNOWN.
10) Exploding Entity ID. This field shall identify the Entity ID of the munition if the munition was represented as an entity. If there was no munition entity, this field shall contain the value NO_SPECIFIC_ENTITY.
11) Location in Entity's Coordinates. This field shall indicate the location of the munition detonation or impact in the target entity's coordinate system when a Target Entity ID is present. If there is no Target Entity ID, this field shall be set to 0.0, 0.0, 0.0 (all bits set to zero).

b) *Explosion Detonations*. A simulation shall issue a Detonation PDU when it determines that an explosion has occurred for a non-munition entity as follows:
   1) A local or remote munition has affected a local entity based on a local detonation, or the receipt of a Detonation PDU, and the simulation determines that the local entity, or part of it, has exploded.
   2) A secondary explosion has occurred. A local entity was affected by another explosion that has taken place.
   3) A local entity has exploded due to a man-made or natural condition not related to another explosion or weapon impact, for example, a spontaneous combustion incident.
   4) The entire entity may explode or part(s) of the entity may explode as follows:
      i) If the entire entity exploded, the Location in Entity's Coordinates field shall be set to 0.0, 0.0, 0.0 (all bits set to zero).
      ii) If a part of the entity exploded, the Location in Entity's Coordinates field shall indicate the coordinates of where the explosion occurred.
   5) If the simulation issuing an *explosion* Detonation PDU determines that the explosion has impacted or otherwise affected an articulated part, attached part, or attached articulated part of an affected entity, the simulation shall include the affected VP record (with parameter values known at impact) in the Detonation PDU. If the part is not directly connected to the base model, all intervening parts shall be included in the Detonation PDU.
   6) The PDU Status record DTI field (bits 4 to 5) shall be set to Non-Munition Explosion (2).
   7) Source Entity ID. This field shall identify the Entity ID of the entity that caused this non-munition entity, or portion thereof, to explode if applicable and known. If the source entity is not known or applicable, then the Source Entity ID shall be set to NO_SPECIFIC_ENTITY.
   8) Target Entity ID. This field shall identify the non-munition entity that exploded. The entire entity may not have exploded.
   9) Exploding Entity ID. This field shall always contain the value NO_SPECIFIC_ENTITY.
   10) Location in Entity's Coordinates. This field shall indicate the location of the explosion in the non-munition entity's (identified by Target Entity ID) coordinate system.

c) *Expendable Detonations*. One *expendable* Detonation PDU shall be required for each *expendable* Fire PDU to indicate the burst or initial bloom of chaff, or the ignition of a flare, as follows:
   1) Once the Detonation PDU is issued, if the expendable is not represented by an expendable entity, no further action is required.
   2) If the detonation is for an expendable entity, the entity will continue to exist to represent the present location of the chaff, flare, or other pyrotechnic until the simulation determines that the

entity no longer needs to be modeled. When this occurs, the expendable entity shall be terminated by issuing a *final* Entity State PDU.

3) The PDU Status record DTI field (bits 4 to 5) shall be set to Expendable (1).
4) The Descriptor field shall contain the Expendable Descriptor record (see 6.2.19.4).
5) Source Entity ID. This field shall identify the Entity ID of the entity that launched the expendable.
6) Target Entity ID. This field shall identify the target threat entity to the entity (Source Entity ID) that launched the expendable, if applicable and known. If the target threat entity is unknown, this field shall contain the value TARGET_ID_UNKNOWN.
7) Exploding Entity ID. This field shall identify the Entity ID of the expendable if the expendable was represented as an entity. If there was no expendable entity, this field shall contain the value NO_SPECIFIC_ENTITY.
8) Location in Entity's Coordinates. This field shall indicate the location of the burst or ignition in the coordinate system of the entity that launched the expendable. If this location is unknown, this field shall be set to 0.0, 0.0, 0.0 (all bits set to zero).

d) *Towed and Air-launched Standoff Decoys*. No Detonation PDU shall be issued for a towed or air-launched standoff decoy entity unless the entity contains explosive material that explodes due to being hit by a munition or collision with the terrain. In that case, an *explosion* Detonation PDU shall be issued.

e) The Detonation Results field for munitions, directed energy weapons, expendables, and explosions shall be set as specified in Table 7. Issuing simulations may support the use of other Detonation Result values as agreed to with other exercise participants.

Table 7—Setting detonation results

Value	Detonation result	Set when the following condition exists
0	No Statement	Not used.
	Munition and Directed Energy Weapons	
1	Entity Impact	The impact or detonation is known to have affected only a specific entity that was physically contacted by the weapon or munition or directed energy beam.
2	Entity Proximate	The impact or detonation is known to have affected only a specific entity, but that entity was not physically contacted by the weapon or munition prior to detonation.
3	Ground/Surface Impact	The impact or detonation is known to have affected only the terrain, including foliage, or water surface and the terrain, foliage, or water surface was physically contacted by the weapon or munition.
4	Ground/Surface Proximate	The impact or detonation is known to have affected only the terrain, including foliage, or water surface, but the terrain, foliage, or water surface was not physically contacted by the weapon or munition prior to detonation.
	Munition, Non-Munition Explosions, and Expendables	

Table 7—Setting detonation results *(continued)*

Value	Detonation result	Set when the following condition exists
5	Detonation	*Munitions*: Neither a specific entity nor the terrain, foliage, or water surface is affected and the munition explodes. *Non-Munition Explosion*: A non-munition entity or part of an entity has exploded. *Expendable*: A chaff expendable has burst or a flare has ignited.
Munition or Expendable		
6	No Detonation	The munition did not explode (e.g., a dud) or the expendable did not activate (burst or ignite).

5.4.4.4 Receipt of the Detonation PDU

Upon receipt of a Detonation PDU, a simulation shall use the information therein as follows: If this is a munition detonation or non-munition explosion, it shall determine damage to its own entities and other objects and for presenting any visual, seismic, aural, or other effects that would be present. The following requirements shall be met:

a) *Munition*. A Detonation PDU shall be processed as a munition if the PDU Status record DTI field (bits 4 to 5) is set to Munition (0):
   1) As a minimum, a damage assessment or lethality model shall process and use the data contained in the Descriptor, Target Entity ID, Location in World Coordinates, Location in Entity's Coordinates, and Detonation Results fields.
   2) If a Target Entity ID is not specified, the Location in World Coordinates field value shall be compared with all local active entity locations in the vicinity that could have been affected by the detonation. If a Target Entity ID is specified, the Location in Entity's Coordinates field value shall be used instead.
   3) A simulation that only maintains an overall damage status for an entity as reflected in the Damage, Mobility, and Fire Power fields of the Entity Appearance record need not perform any further damage assessment for that entity if those three fields are already set to Destroyed (3), Mobility Kill (1), and Fire Power Kill (1), respectively.
   4) A Detonation Result value that indicates the shooter's perspective of the damage to an entity may be used if the receiving simulation determines that it would assist its damage assessment process. If the receiving simulation uses the detonation result value, then it shall support the values specified in Table 7. Receiving simulations may support the use of other Detonation Result values as agreed to with other exercise participants.
   5) The munition entity shall not be depicted as exploding if any of the following conditions exist:
      i) The Detonation Results value is No Detonation (6).
      ii) The Munition Type indicates a kinetic weapon.
   6) An Entity Damage Status PDU may be used by the receiving simulation to indicate a specific area on the surface of an entity that was affected (see 5.4.6).

b) *Non-Munition Explosions*. A Detonation PDU shall be processed as representing an explosion when the PDU Status record DTI field (bits 4 to 5) is set to Non-Munition Explosion (2):
   1) Receiving simulations shall perform processing to determine whether the explosion affected one of their entities.

2) A Detonation Result of Detonation (5) is applicable to an explosion. The optional processing of any other value received is beyond the scope of this standard.

3) An Entity Damage Status PDU may be used by the receiving simulation to indicate a specific area on the surface of an entity that was affected (see 5.4.6).

c) *Expendables*. A Detonation PDU shall be processed as an expendable when the PDU Status record DTI field (bits 4 to 5) is set to Expendable (1):

1) If the expendable is chaff or flares, it shall represent the burst or initial bloom of chaff or the ignition of a flare, respectively:

i) If a receiving simulation can detect chaff or flares, it shall use the information received in the Detonation PDU to affect its sensors and any visual presentation.

ii) If an interceptor missile is en route to the target entity that dropped the chaff or flares, the model should take into account the presence of the chaff or flare countermeasure.

2) A Detonation Result of Detonation (5) is applicable to an expendable. The optional processing of any other value received is beyond the scope of this standard.

5.4.4.5 Termination of entities

The termination of the existence of a munition, an entity that explodes, or an expendable entity in an exercise shall be indicated by the issuance of an Entity State PDU with the Entity Appearance record State field (bit 23) set to Deactivated (1). If a munition that detonates or an entity (or part of it) explodes and remains active after issuing the Detonation PDU, the appropriate damage status shall be reflected in the Entity Appearance record. Additional entities may be created to represent fragments of an entity affected by a munition or explosion.

5.4.5 Directed Energy (DE) Fire PDU

5.4.5.1 Purpose

The DE Fire PDU shall be used to communicate information associated with the firing of a DE weapon. A simulation that transmits the DE Fire PDU shall also be required to implement, as a minimum, the transmission of the Fire and Detonation PDUs and the receipt of the Entity Damage Status PDU. See Annex A for detailed PDU requirements related to directed energy engagements.

A DE weapon may be directed at a specific target or at an area possibly containing several targets. Distinct aimpoint record types are defined to describe these two general cases. When firing at a specific target, a Precision Aimpoint is employed. The DE Precision Aimpoint record is most commonly used to represent the firing of a high-energy laser. Other types of directed energy weapons propagate effects over a wide region. Representation of these weapons uses the DE Area Aimpoint record to describe characteristics and effects.

A high-fidelity DE engagement shall commence with the initial output of the DE Fire PDU immediately followed by a Fire PDU and shall terminate with a final DE Fire PDU followed immediately by a Detonation PDU. The same Event ID shall be used for the DE Fire, Fire, and Detonation PDUs. Pulsed directed energy weapons where the DE beam is turned on and off in rapid succession shall be considered the continuation of a single engagement, and no additional Fire or Detonation PDU shall be issued. The DE Fire PDU may be output more than once while a shot is in progress. This permits a receiving simulation to continuously assess damage while the shot is taking place and to provide immediate feedback by issuing changes to the target in appropriate PDUs (e.g., Entity State and Entity Damage Status PDUs).

All DE weapons shall be attached to an entity. The simulation that owns the DE weapons system shall issue the DE Fire PDU for that system. There is no requirement that this simulation owns the entity to which a DE weapons system is attached.

A simulation implementing the DE Fire PDU shall model environmental conditions that would affect the shot, as well as terrain. Countermeasures that would be known to the firing entity through sensors and derived from other sources available to the DE weapons system should be modeled. The inclusion of specific environmental, terrain, and countermeasure detection requirements is beyond the scope of this standard.

Annex A contains additional requirements related to directed energy weapons including algorithms and specific sequence and minimum implementation rules for both basic and high-fidelity DE simulations.

5.4.5.2 Information contained in the DE Fire PDU

The DE Fire PDU shall contain the following information effective at the instant the DE weapon enters the simulated world:

a) The identifier of the entity firing the weapon.
b) An event ID.
c) The entity type for the DE weapon beam.
d) The DE weapon shot start simulation time.
e) The DE weapon cumulative shot simulation time.
f) The aperture/emitter location in firing entity coordinates.
g) The aperture diameter.
h) The DE weapon wavelength.
i) The DE weapon pulse repetition frequency.
j) The DE weapon pulse width.
k) The DE weapon status flags indicating on or off.
l) The DE weapon pulse shape.
m) Variable DE records, as applicable.

5.4.5.3 Issuance of the DE Fire PDU

A simulation shall issue a DE Fire PDU when any of the following occur:

a) At the moment that an entity begins a DE weapon shot.
b) While a shot is in-progress, a predetermined length of real-world time has elapsed since the issuance of the last DE Fire PDU. The DE Fire PDU heartbeat timer parameter shall be identified by the symbolic name HBT_PDU_DE_FIRE. (See 6.1.8 for parameter details and default values.)
c) At the moment that an entity ends a DE weapon shot.
d) If the value in any one of the following fields changes: pulse repetition frequency, pulse width, pulse shape.
e) If the DE Precision Aimpoint record is present, a DE Fire PDU shall be issued when any of the following occur:
    1) If any of the fields characterizing energy deposition vary from the last updated value by more than a preestablished percentage. The fields considered are the Peak Irradiance and the Beam Spot Cross-Section Semi-Major and Semi-Minor Axes. The percentage threshold parameter for energy deposition shall be identified by the symbolic name DE_ENERGY_THRSH (see 6.1.8 for parameter details and default values).
    2) When the dead reckoned spot location varies from the shooter's calculated location by more than a predetermined threshold distance. The positional threshold parameter for directed energy precision aiming shall be identified by the symbolic name DE_PRECISION_AIMING_THRSH (see 6.1.8 for parameter details and default values).

f) If the DE Area Aimpoint record is present, a DE Fire PDU shall be issued when any of the three angle values in the Beam Direction field of the Beam Antenna Pattern record vary by more than a predetermined threshold. The angular threshold parameter for the directed energy area aiming shall be identified by the symbolic name DE_AREA_AIMING_THRSH (see 6.1.8 for parameter details and default values).

g) A single Fire PDU shall be issued immediately following the initial DE Fire PDU. A detonation PDU shall be issued after the final DE Fire PDU of that engagement indicating a change of the DE weapon to the off state. Only one Fire/Detonation PDU pair shall be issued per DE engagement regardless of the number of DE Fire PDUs that are issued.

5.4.5.4 Receipt of the DE Fire PDU

Upon reception of the DE Fire PDU, a simulation application shall determine damage and lethality effects for all entities owned by that simulation. It shall update the Entity State Appearance field and/or emit Entity Damage Status PDUs to reflect the damage assessed for the entity.

A simulation shall maintain a real-world timer for each active shot. This timer shall be reset with the reception of each DE Fire PDU describing a specific shot (as identified by a unique Event ID). In the event that no DE Fire PDU is received within a simulation time interval defined by the HBT_TIMEOUT_MPLIER multiplied by the basic heartbeat interval [described in item b) in 5.4.3.2], the simulation will mark the shot as "off."

Based on network latency and differences in geometric models, the Target Spot Location information provided by the firing entity in a DE Precision Aimpoint record may not resolve to a spot on the surface of the target entity's geometric model in the receiving application. The following rules shall be applied to determine the exact target spot location in the receiving application:

a) The receiving application shall maintain a dead reckoned target spot location based on the PDU timestamp and the dead reckoning information provided in the DE Precision Aimpoint record.

b) In the world coordinate frame, the receiving application shall generate a vector emanating from the aperture/emitter location on the firing entity directed at the current dead reckoned target spot location.

c) The actual target spot used by the receiving application shall be the first point of intersection of the vector computed in item b) with the target's geometric model.

d) If no intersection is detected using the method described in item a) through item c), the receiving application shall generate a vector from the center of the target entity passing through the current target spot and use the farthest point intersecting with the target entity as the target spot.

These rules only apply to damage assessment methodologies that include the use of geometric target models. These rules are not meant to imply that geometric models are required to implement all possible lethality and damage prediction codes.

5.4.6 Entity Damage Status PDU

5.4.6.1 Purpose

The Entity Damage Status PDU shall be used to communicate detailed damage information sustained by an entity regardless of the source of the damage. The cause of the damage may be a weapon fired at the entity, a collision with another object, or some other reason. The Entity Damage Status PDU enables damage to a specific location on an entity to be conveyed whether or not that location is associated with an articulated or attached part. The information conveyed in this PDU augments damage information communicated in the Entity State and other PDUs.

The Entity Damage Status PDU contains a Standard Variable Specification record (see 6.2.83) used to contain Damage Description records (see 6.2.15).

5.4.6.2 Information contained in the Entity Damage Status PDU

The Entity Damage Status PDU shall contain the following information excluding the PDU Header:

a) The identifier of the damaged entity.

b) Zero or more Damage Description records, as applicable.

5.4.6.3 Issuance of the Entity Damage Status PDU

Except as noted below, a simulation shall not be required to transmit an Entity Damage Status PDU for an undamaged entity.

A simulation shall issue an Entity Damage Status PDU when any of the following occurs:

a) When the detailed damage state changes for an entity owned by the simulation.

b) In the event of a simulation reset for each entity previously marked as damaged, the simulation shall send an Entity Damage Status PDU to reflect an owned entity's reset damage state if the same Entity ID is used again.

c) A predetermined length of real-world time has elapsed since the issuance of the last Entity Damage Status PDU for a given entity. The Entity Damage Status PDU heartbeat timer parameter shall be identified by the symbolic name HBT_PDU_ENTITY_DAMAGE. (See 6.1.8 for parameter details and default values.) This rule shall only be applied after the first Entity Damage Status PDU has been transmitted for a given entity.

d) Once the entity is no longer damaged (e.g., because all the damage has been repaired), then a *final* Entity Damage Status PDU shall be issued with the Number of Damage Description Records field set to zero and without including any Damage Description records. The Damage, Mobility, and Fire Power fields of the Entity Appearance record shall also be set accordingly.

The Entity Damage Status PDU shall not be transmitted following the transmission of an Entity State PDU with the Appearance record State field (bit 23) set to Deactivated (1).

When issuing an Entity Damage Status PDU:

— A separate Entity Damage Description record shall be used for each damage location on the entity.

— All Damage Description records reflecting the present damage to the entity shall be included in each Entity Damage Status PDU regardless of whether the PDU is being issued because one or more records are being added, changed, or deleted.

5.4.6.4 Receipt of the Entity Damage Status PDU

Upon reception of the Entity Damage Status PDU, a simulation application shall update the representation of the entity to reflect the information contained in the PDU. The absence of an Entity Damage Status record previously included indicates that there is no longer damage at that location.

If an Entity Damage Status PDU is received with the Number of Damage Description Records field set to zero or a predetermined length of time has elapsed since any entity's last Entity Damage Status PDU, then all simulations shall update the representation of the entity to reflect that there is no longer any damage for that entity. The length of time shall be established at exercise start, although it may be changed during the exercise. The entity damage status timeout parameter shall be computed as shown in NOTE 3 of Table 25.

5.5 Logistics

5.5.1 General

Information associated with the representation of logistics support in a DIS exercise shall be communicated through the use of several PDUs—Service Request PDU, Resupply Offer PDU, Resupply Received PDU, Resupply Cancel PDU, Repair Complete PDU, and Repair Response PDU (see 7.4).

The procedures associated with logistics support require the definition of several timers. The value of these timers shall be identified by the symbolic names as follows (see 6.1.8 for parameter details and default values):

a) Resupply Receive Timer 1: RESUP_REC_T1

b) Resupply Receive Timer 2: RESUP_REC_T2

c) Resupply Supplier Timer 1: RESUP_SUP_T1

d) Repair Receive Timer 1: REPAR_REC_T1

e) Repair Supplier Timer 1: REPAR_SUP_T1

f) Repair Supplier Timer 2: REPAR_SUP_T2

5.5.2 Procedure for logistics support

Logistics support in DIS shall be accomplished through a series of request and response messages between two entities. Two types of service have been defined for DIS—resupply and repair.

5.5.3 State information for resupply service

5.5.3.1 General

The different states and transitions for resupply service are described in 5.5.3.2 and 5.5.3.3. An example of the resupply function is given in 5.5.9.

5.5.3.2 Receiving entity

The receiving entity may be in one of three states:

a) *Ready State*. A receiving entity is in the Ready state when it is not in either the Requesting state or the Receiving state.

b) *Requesting State*. A receiving entity is in the Requesting state when it has requested supplies and has not received a reply to its request.

c) *Receiving State*. A receiving entity is in the Receiving state when it has been offered supplies and is in the process of receiving them.

The behavior of the receiving entity during resupply service is shown in Figure 7. The associated state transitions for the receiving entity are as defined in Table 8.

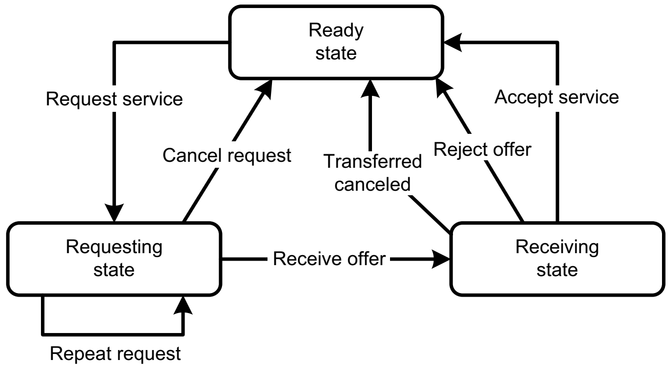


Figure 7—Receiving entity behavior during resupply

Table 8—Resupply state transitions for a receiving entity

Transition	Condition and actions
Request service	When conditions for resupply are met, the entity shall issue a Service Request PDU (see 5.5.5). The entity shall proceed from the Ready state to the Requesting state, and Resupply Receiver Timer 1 shall be set.
Cancel request	When conditions for resupply are no longer met or when a Resupply Cancel PDU (see 5.5.8) is received from a supplying entity, Resupply Receiver Timer 1 shall be canceled and the entity shall proceed from the Requesting state to the Ready state.
Repeat request	When Resupply Receiver Timer 1 expires, the Service Request PDU shall be reissued and the timer shall be reset.
Receive offer	When a Resupply Offer PDU (see 5.5.6) is received, Resupply Receiver Timer 1 shall be canceled and Resupply Receiver Timer 2 shall be set to the period of time required for receiving some portion of the offered supplies. The entity shall proceed from the Requesting state to the Receiving state.
Reject offer	When conditions for resupply are no longer met, a Resupply Cancel PDU shall be issued, Resupply Receiver Timer 2 shall be canceled, and the entity shall proceed from the Receiving state to the Ready state.
Accept service	When Resupply Receiver Timer 2 expires, the count of supplies on board shall be incremented and a Resupply Received PDU (see 5.5.7) shall be issued. The entity shall then proceed from the Receiving state to the Ready state.
Transfer canceled	When a Resupply Cancel PDU (see 5.5.8) is received, the ongoing transfer shall be canceled, Resupply Receiver Timer 2 shall be canceled, and the entity shall proceed from the Receiving state to the Ready state. No supplies (from the canceled transfer) shall be transferred.

The fact that an entity is able to provide resupply or repair service is indicated in the Capabilities field of its Entity State PDU.

Resupply Receive Timer 2 shall be set to a value less than Resupply Supplier Timer 1. The difference between these two timers shall be sufficient to account for network and processing delays. A minimum difference of 5 s is suggested.

5.5.3.3 Supplying entity

The supplying entity may be in one of two states:

a) *Ready State*. A supplying entity is in the Ready state when it is able to receive a request for supplies and is able to offer supplies to a receiving entity.

b) *Offering State*. A supplying entity is in the Offering state when it has made an offer of supplies and is waiting for the receiving entity to indicate the quantity of supplies it has taken.

The behavior of the supplying entity during resupply service is shown in Figure 8. The state transitions for the supplying entity are as defined in Table 9.

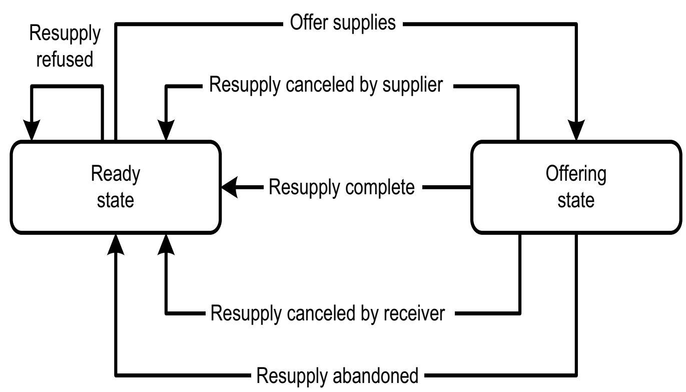


Figure 8—Supply entity behavior during resupply

Table 9—Resupply state transitions for a supplying entity

Transition	Condition and actions
Offer supplies	When a Service Request PDU (see 5.5.5) is received and conditions for resupply are met, a Resupply Offer PDU (see 5.5.6) shall be issued, Resupply Supplier Timer 1 shall be set, and the entity shall proceed from the Ready state to the Offering state.
Resupply refuse	When a Service Request PDU (see 5.5.5) is received and the requested supplies are unavailable, a Resupply Cancel PDU (see 5.5.8) shall be issued. The entity shall remain in the Ready state.
Resupply complete	When a Resupply Received PDU (see 5.5.7) is received, Resupply Supplier Timer 1 shall be canceled. The count of supplies on board shall be decremented and the entity shall proceed from the Offering state to the Ready state.

Table 9—Resupply state transitions for a supplying entity *(continued)*

Transition	Condition and actions
Resupply canceled by receiver	When a Resupply Cancel PDU (see 5.5.8) is received, Resupply Supplier Timer 1 shall be canceled. The count of supplies on board shall not change and the entity shall proceed from the Offering state to the Ready state.
Resupply canceled by supplier	When conditions for resupply are no longer met, a Resupply Cancel PDU (see 5.5.8) shall be issued, Resupply Timer 1 shall be canceled, and the entity shall proceed from the Offering state to the Ready state.
Resupply abandoned	When Resupply Supplier Timer 1 expires, the transfer shall be abandoned, the count of supplies on board shall not change, and the entity shall proceed from the Offering state to the Ready state.

5.5.4 State information for repair service

5.5.4.1 General

The different states and transitions for repair are described in 5.5.4.2 and 5.5.4.3. An example of repair service is given in 5.5.13.

5.5.4.2 Receiving entity

The receiving entity may be in one of two states:

a) *Ready State*. A receiving entity is in the Ready state when it is able to request repairs from an entity with repair capabilities.

b) *Requesting/Receiving State*. A receiving entity is in the Requesting/Receiving state when it has requested repairs and has not received a response to its request.

The behavior of the receiving entity during repair service is shown in Figure 9. The associated state transitions are described in Table 10.

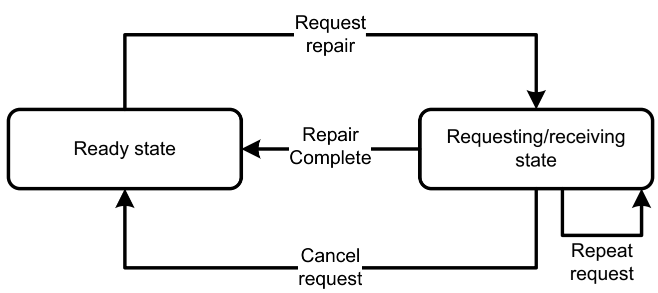


Figure 9—Receiving entity behavior during repair

Table 10—Repair service state transitions for a receiving entity

Transition	Condition and actions
Request repair	When conditions for repair service are met, the entity shall issue a Service Request PDU (see 5.5.5), Repair Receiver Timer 1 shall be set, and the entity shall proceed from the Ready state to the Requesting/Receiving state.
Repair complete	When a Repair Complete PDU (see 5.5.10) is received, the entity shall issue a Repair Response PDU (see 5.5.11), Repair Receiver Timer 1 shall be canceled, and the entity shall proceed from the Requesting/Receiving state to the Ready state.
Cancel request	When conditions for repair service are no longer met, the receiving entity shall cease to issue Service Request PDUs, Repair Receiver Timer 1 shall be canceled, and the entity shall proceed from the Requesting/Receiving state to the Ready state.
Repeat request	When Repair Receiver Timer 1 expires, the Service Request PDU shall be reissued and the time reset. The entity shall remain in the Requesting/Receiving state.

5.5.4.3 Repairing entity

The repairing entity may be in one of three states:

a) *Ready State*. A repairing entity is in the Ready state when it is able to offer repairs to a receiving entity.

b) *Offering State*. A repairing entity is in the Offering state when it has received a request for repairs and is responding to the request.

c) *Repair Completed State*. A repairing entity is in the Repair Completed state when it has completed a repair and it is waiting for a response.

The behavior of the repairing entity during repair service is shown in Figure 10. Repair service state transitions for a repairing entity are described in Table 11.

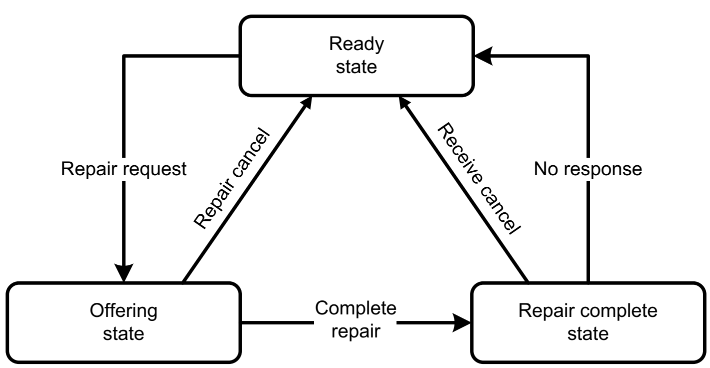


Figure 10—Repairing entity behavior during repair

Table 11—Repair service state transitions for a repairing entity

Transition	Condition and actions
Repair request	When the repairing entity has received a Service Request PDU (see 5.5.5) and has identified itself as the intended repairing entity, Repair Supplier Timer 1 shall be set. Every time a Service Request PDU is received from the entity requesting the repair service, Repair Supplier Timer 1 shall be reset. The repairing entity shall then proceed from the Ready state to the Offering state.
Complete repair	When the repair is complete, the entity shall issue a Repair Complete PDU (see 5.5.10), shall cancel Repair Supplier Timer 1, shall set Repair Supplier Timer 2 and shall proceed from the Offering state to the Repair Completed state.
Receive response	When a Repair Response PDU is received from the receiving entity, the entity shall cancel the Repair Supplier Timer 2 and shall proceed from the Repair Completed state to the Ready state.
No response	When Repair Supplier Timer 2 has expired, the entity shall proceed from the Repair Completed state to the Ready state.
Repair canceled	When Repair Supplier Timer 1 has expired, the entity shall proceed from the Offering state to the Ready state.

5.5.5 Service Request PDU

5.5.5.1 Purpose

The Service Request PDU shall be used to communicate information associated with one entity requesting a service from another.

5.5.5.2 Information contained in the Service Request PDU

The Service Request PDU shall contain the following information:

a) Identification of the entity issuing the PDU
b) Identification of the entity that is able to provide the service required by the requesting entity
c) Type of service being requested. Services defined are:
   1) Resupply
   2) Repair
d) Number and types of supplies if the service required is resupply

5.5.5.3 Issuance of the Service Request PDU

The Service Request PDU shall be issued by an entity requesting logistics support when appropriate conditions exist. Appropriate conditions include internal conditions (such as certain crew actions in the simulator) as well as external conditions (such as conditions existing in the simulated world, entities not destroyed, or being within a certain distance of the entity requesting logistics support).

The Service Request PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.5.5.4 Receipt of the Service Request PDU

Upon receipt of a Service Request PDU, the entity that receives the PDU shall respond in one of the following ways:

a) If the service requested is resupply and the entity that receives the PDU is able to provide the needed supplies, the supplying entity shall issue a Resupply Offer PDU (see 5.5.6).

b) If the service requested is repair, the repairing entity shall simulate the needed repairs in the following manner: The repair process is allowed to proceed as long as the repairing entity continues to receive Service Request PDUs. If Service Request PDUs cease to be received and are not seen for the period of Repair Supplier Timer 1, the repairing entity shall assume that conditions for repair no longer exist and therefore shall abort the process.

c) If the service requested is resupply and the entity receiving the PDU is unable to provide the supplies requested, then the supplying entity shall issue a Resupply Cancel PDU (see 5.5.8).

5.5.6 Resupply Offer PDU

5.5.6.1 Purpose

A Resupply Offer PDU shall be used to communicate the offer of supplies by a supplying entity to a receiving entity.

5.5.6.2 Information contained in the Resupply Offer PDU

The Resupply Offer PDU shall contain the following information:

a) Identification of the entity requesting resupply

b) Identification of the supplying entity that issued the PDU

c) Number of types of supplies that the supplying entity is able to provide

d) Supply types available and the amount of each

5.5.6.3 Issuance of the Resupply Offer PDU

The Resupply Offer PDU shall be issued by an identified supplying entity that has received a Service Request PDU (see 5.5.5) requesting resupply service.

The Resupply Offer PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.5.6.4 Receipt of the Resupply Offer PDU

Upon receipt of a Resupply Offer PDU, the receiving entity shall proceed from the Requesting state to the Receiving state. When receipt of the supplies is complete, the receiving entity shall respond by issuing a Resupply Received PDU.

5.5.7 Resupply Received PDU

5.5.7.1 Purpose

A Resupply Received PDU shall be used to acknowledge the receipt of supplies by the receiving entity.

5.5.7.2 Information contained in the Resupply Received PDU

The Resupply Received PDU shall contain the following information:

a) Identification of the entity requesting resupply
b) Identification of the supplying entity
c) Number of types of supplies that the receiving entity took
d) Supply types available and the amount of each taken by the receiving entity

5.5.7.3 Issuance of the Resupply Received PDU

The Resupply Received PDU shall be issued by an identified receiving entity to indicate the supplies actually transferred from the supplying entity to the receiving entity.

The Resupply Received PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.5.7.4 Receipt of the Resupply Received PDU

Upon receipt of a Resupply Received PDU, the supplying entity shall decrement the number of supplies on board and shall proceed to the Ready state.

5.5.8 Resupply Cancel PDU

5.5.8.1 Purpose

The Resupply Cancel PDU shall be used to communicate the canceling of a resupply service provided through logistics support.

5.5.8.2 Information contained in the Resupply Cancel PDU

The Resupply Cancel PDU shall contain the following information:

a) Identification of the entity receiving supplies
b) Identification of the entity providing supplies

5.5.8.3 Issuance of the Resupply Cancel PDU

The Resupply Cancel PDU may be issued by either the receiving entity or the supplying entity during the resupply period to cancel the resupply service.

The Resupply Cancel PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.5.8.4 Receipt of the Resupply Cancel PDU

The entity receiving the Resupply Cancel PDU shall cancel its timers and return to the Ready state. No supplies (for the canceled transaction) are transferred.

5.5.9 Example of resupply service

If the service requested by the receiving entity is resupply, a scenario such as the following may take place:

A resupply interaction begins when a simulation application determines that one of its entities is in a state such that it should be resupplied. For example, a ground vehicle might come to a stop and open its refueling port. When this occurs, the simulation application issues a Service Request PDU to a potential resupplying entity and sets Resupply Receiver Timer 1. Potential resuppliers may be identified by their proximity, state, and the content of the Capabilities field (as determined from their Entity State PDUs). This event is shown as the request service transition in Figure 7.

A simulation entity that receives a Service Request PDU and is identified as the specified supplying entity responds by offering some portion of whatever supplies are currently loaded on board. This condition is shown in Figure 9 as a transition from the Ready state to the Offering state. Meanwhile, the receiving entity reissues its Service Request PDU when Resupply Receiver Timer 1 expires until such an offer is forthcoming. The offer takes the form of a Resupply Offer PDU issued by the supplying entity. The supplies offered should be a subset of those possessed by the supplying entity, and a subset of those requested by the receiving entity.

Upon receiving the offer of supplies, the receiving entity changes from the Requesting state to the Receiving state. The receiving entity then has until the Resupply Receiver Timer 2 expires to acknowledge the receipt of those supplies by returning to the supplying entity a Resupply Received PDU listing the exact supplies taken. The receiving entity need not accept all of the supplies offered, but instead it can indicate in its receipt how many it did accept. After delaying until Resupply Receiver Timer 2 expires, the receiving entity issues its Resupply Received PDU and returns to the Ready state. When the supplying entity receives the Resupply Received PDU, it also returns to the Ready state, and the procedure is complete.

The simulation time required to return the Resupply Received PDU, and the quantity of supplies reported by that PDU as taken, determine the rate at which the supplying entity and the receiving entity are able to transfer munitions. For example, an M1 tank obtaining 105 mm shells from an ammunition supply truck might acknowledge receipt of a single round after 40 s; this results in a simulated rate of resupply for the M1 tank of one round every 40 s.

Throughout the transfer process, both the receiving entity and the supplying entity continue to monitor the conditions necessary for the transfer. If any of these conditions ceases to hold, either entity can abort the transfer by issuing a Resupply Cancel PDU, with the result that no supplies are transferred (for the transfer that was in process). Alternatively, the receiving entity can terminate the transfer early but accept some of the supplies offered by issuing a Resupply Received PDU for the partial load. Finally, if the supplying entity waits in the Offering state for the entire period of Resupply Supplier Timer 1 but receives no Resupply Received PDU (perhaps the receiving entity has withdrawn from the exercise), it should return to the Ready state and assume that no supplies were taken.

5.5.10 Repair Complete PDU

5.5.10.1 Purpose

The Repair Complete PDU shall be used by the repairing entity to communicate the repair that has been performed for the entity that requested repair service.

5.5.10.2 Information contained in the Repair Complete PDU

The Repair Complete PDU shall contain the following information:

a) Identification of the entity requesting repair service

b) Identification of the entity providing the repair

c) Repair performed by the repairing entity

5.5.10.3 Issuance of the Repair Complete PDU

The Repair Complete PDU shall be issued by a repairing simulation entity upon completion of a repair service requested by the receiving entity in a Service Request PDU (see 5.5.5).

The Repair Complete PDU shall be issued by using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.5.10.4 Receipt of the Repair Complete PDU

Upon receipt of the Repair Complete PDU, the receiving entity shall issue a Repair Response PDU (see 5.5.11) and shall proceed from the Requesting/Receiving state to the Ready state.

5.5.11 Repair Response PDU

5.5.11.1 Purpose

A Repair Response PDU shall be used by the receiving entity to acknowledge the receipt of a Repair Complete PDU (see 5.5.10).

5.5.11.2 Information contained in the Repair Response PDU

The Repair Response PDU shall contain the following information:

a) Identification of the entity requesting repair service
b) Identification of the entity providing the repair
c) Result of the repair

5.5.11.3 Issuance of the Repair Response PDU

The Repair Response PDU shall be issued by the entity receiving repair service upon receipt of a Repair Complete PDU from the repairing entity.

The Repair Response PDU shall be issued by using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.5.11.4 Receipt of the Repair Response PDU

Upon receipt of the Repair Response PDU, the repairing entity shall note that the receiving entity has received the repair.

5.5.12 Cancellation of repair service

If the receiving entity intends to cancel the repair service before the repairs are completed, it shall cease to issue Service Request PDUs and shall return to the Ready state. The supplying entity that does not receive Service Request PDUs for the period of Repair Supplier Timer 1 shall abandon the repair service and shall return to the Ready state. If the supplying entity intends to cancel the repair service, it shall issue a Repair Complete PDU and shall set the Repair field to No Repairs Performed (0). The receiving entity shall respond with the issue of a Repair Response PDU indicating the Repair Result as Service Canceled by the Supplier (4).

5.5.13 Example of repair service

If the service requested by the receiving entity is repair, a scenario such as the following may take place:

A simulation application issues a Service Request PDU and sets Repair Receiver Timer 1 when it determines that one of its simulation entities is in need of repairs. The Service Request PDU is issued to an entity capable of performing repairs as determined by its proximity, state, and the content of the field capabilities conveyed in its Entity State PDU. This event is shown as the request repair transition in Figure 9.

If the repair process successfully runs to completion, the repairing entity may then accomplish the repair by issuing a Repair Complete PDU to notify the receiving entity of the repair, and returning to the Ready state. The receiving entity's simulation application acknowledges the receipt of the Repair Complete PDU by returning a Repair Response PDU. (This acknowledgment simply indicates that the repair was performed, not that the repair was appropriate or that the disabled vehicle has been made operational because of the repair.)

5.6 Simulation management

5.6.1 General

The content and procedure for use of Simulation Management (SIMAN) PDUs in a DIS exercise are established in 5.6.2 through 5.6.7.4. SIMAN PDUs may or may not be required for participation in an exercise, depending on the requirements of each exercise and which non-SIMAN PDUs have been implemented. See 5.12 for the requirements for Simulation Management with Reliability (SIMAN-R) PDUs, which incorporate a mechanism to allow the specification of the application-level communications reliability service. See Transfer Ownership (5.9.4) for the special use of SIMAN and SIMAN-R PDUs to support that function.

NOTE 1—When the term "simulation" is used, it is understood to mean a simulation application and it will be represented by a Simulation Identifier (see 6.2.81). When a SIMAN PDU is stated to be issued "to an entity," it is understood to be issued to the simulation application that owns the entity. When the phrase "the receiving entity" is used, it is understood to mean that the simulation application that owns the entity will perform the indicated action pertaining to the entity.

NOTE 2—Prior to DIS Version 6, the Site/Application fields of an Entity ID always identified the simulation application that owned the entity. With the advent of the transfer ownership function in DIS Version 6, this is not necessarily true as when an entity is transferred between simulations, the Site/Application (i.e., Simulation Address) remains unchanged for an Entity ID. The Transferred Entity Indicator of the PDU Status record of an Entity State or Entity State Update PDU, introduced in DIS Version 7, can be used to determine whether the Site/Application fields of an Entity ID contained in the Originating or Receiving ID field of a SIMAN PDU are also the owner of the entity [see item b) in 5.9.4.2.1].

PDU types comprising the Simulation Management protocol family are:

a) Create Entity
b) Remove Entity
c) Start/Resume
d) Stop/Freeze
e) Acknowledge
f) Action Request
g) Action Response
h) Data Query
i) Set Data
j) Data
k) Event Report
l) Comment

5.6.2 Simulation Management PDU Header

5.6.2.1 Introduction

The Simulation Management PDU Header consists of the PDU Header common to all PDUs and an Originating ID and Receiving ID field.

5.6.2.2 PDU Header field

The PDU header field of the Simulation Management PDU Header record shall be the PDU Header record common to all PDUs (see 6.2.66).

5.6.2.3 Originating ID field

The Originating ID field identifies the simulation or entity that issues the SIMAN or SIMAN-R PDU. All SIMAN and SIMAN-R PDUs may be issued by an entity or a simulation. When a SIMAN or SIMAN-R PDU is issued by an entity, then that entity shall exist at the time of issuing the PDU.

5.6.2.4 Receiving ID field

The Receiving ID field identifies the simulation, entity, set of simulations, or set of entities that the SIMAN or SIMAN-R PDU is sent to. The Site Number field may contain ALL_SITES or NO_SITE, the Application Number field may contain ALL_APPLIC or NO_APPLIC, and the Reference Number field may contain ALL_ENTITIES or NO_ENTITY. The Reference Number field in the Create Entity PDU or Create Entity-R PDU may also contain RQST_ASSIGN_ID. The meaning of the various combinations of allowed values for the Receiving ID are shown in Table 12.

Table 12—Receiving ID combinations

Site number	Application number	Reference number	Meaning
65 535	65 535	0	All simulation applications
1 to 65 534	65 535	0	All simulation applications at the specified site
65 535	1 to 65 534	0	One specified simulation application at every site; for example, if every site had a logger with an Application ID of 100, then 65 535:100:0 would address all these loggers
1 to 65 534	1 to 65 534	0	One simulation application
65 535	65 535	65 535	All entities
1 to 65 534	65 535	65 535	All entities at the specified site
65 535	1 to 65 534	65 535	All entities at one simulation application (with the specified application ID) at every site
1 to 65 534	1 to 65 534	65 535	All entities at one simulation application
65 535	65 535	1 to 65 533	One entity at every simulation application

Table 12—Receiving ID combinations *(continued)*

Site number	Application number	Reference number	Meaning
1 to 65 534	65 535	1 to 65 533	One entity at every simulation application at a specified site; for example, if site 42 contained only manned flight simulators and the manned entity was the first entity created, then 42:65 535:1 would address all the manned entities
65 535	1 to 65 534	1 to 65 533	One entity at one simulation application (with the specified application ID) at every site
1 to 65 534	1 to 65 534	1 to 65 533	One entity at one simulation application
0	0	0	No entity
0	0	1 to 65 535	No site or application
0	1 to 65 535	0	No site
0	1 to 65 535	1 to 65 535	Application only
1 to 65 535	0	0	No application
1 to 65 535	0	1 to 65 535	Site only
1 to 65 534	1 to 65 534	65 534	Special case for Create Entity and Create Entity-R PDUs only: create entity at the specified simulation application. The receiving simulation application shall allocate the Entity ID for the new entity.
65 535	65 535	65 534	Special case for Create Entity and Create Entity-R PDUs only: create entity at every simulation application. The receiving simulation application shall allocate the Entity ID for the new entity.
65 535	1 to 65 534	65 534	Special case for Create Entity and Create Entity-R PDUs only: create entity at the simulation application with the specified application ID at every site. The receiving simulation application shall allocate the Entity ID for the new entity.
1 to 65 534	65 535	65 534	Special case for Create Entity and Create Entity-R PDUs only: create entity at all simulations at the specified site. The receiving simulation application shall allocate the Entity ID for the new entity.

5.6.2.5 Receipt of PDUs and the Receiving ID field

When a SIMAN or SIMAN-R PDU is received, the simulation application shall determine whether it is the intended recipient of the PDU by inspecting the contents of the Receiving ID field as follows:

a) If the Site Number in the Receiving ID is set to NO_SITE, then the PDU is intended for logging purposes only. Similarly if the Application Number in the Receiving ID is set to NO_APPLIC, then the PDU is intended for logging purposes only. There is no intended recipient of the PDU.

b) If the Reference Number in the Receiving ID is set to NO_ENTITY, then the PDU is addressed to one or more simulation applications. The simulation application shall be an intended recipient if its simulation address matches the Receiving ID.

c) If the Reference Number in the Receiving ID is set to RQST_ASSIGN_ID, then the PDU is addressed to one or more simulation applications. The simulation application shall be an intended recipient if its simulation address matches the Receiving ID.

d) Otherwise the PDU is addressed to one or more entities. The simulation application shall be an intended recipient if it owns one or more entities whose Entity ID matches the Receiving ID.

NOTE—Simulation applications that implement Transfer Ownership and Simulation Managers need to take care when addressing entities since the Site Number and Application Number of an entity may not be the same as the Site Number and Application Number of the simulation application that currently owns the entity. So, for example, if entity 10.13.42 has been transferred to simulation application 72.55, then a SIMAN PDU with the Receiving ID set to 10.13.65 535 will target all entities that have been created by simulation application 10.13 whether or not they are currently owned by simulation application 10.13. In particular, in this example, simulation application 72.55 is also an intended recipient of the SIMAN PDU.

Simulation applications shall match a Simulation Address or a Simulation Identifier with the Receiving ID as follows:

— A Simulation Address (see 6.2.80) shall match the Receiving ID if the Site Number in the Simulation Address matches the Site Number in the Receiving ID, the Application Number in the Simulation Address matches the Application Number in the Receiving ID, and the Reference Number in the Receiving ID is set to NO_ENTITY.

— A Simulation Identifier (see 6.2.81) shall match the Receiving ID if the Site Number in the Simulation Identifier matches the Site Number in the Receiving ID, the Application Number in the Simulation Identifier matches the Application Number in the Receiving ID, and the Reference Number in the Simulation Identifier matches the Reference Number in the Receiving ID.

— A site number shall match the Site Number in the Receiving ID if it equals the Site Number in the Receiving ID or the Site Number field in the Receiving ID is set to ALL_SITES.

— An application number shall match the Application Number in the Receiving ID if it equals the Application Number in the Receiving ID or the Application Number field in the Receiving ID is set to ALL_APPLIC.

— A reference number shall match the Reference Number in the Receiving ID if it equals the Reference Number in the Receiving ID or the Reference Number field in the Receiving ID is set to ALL_ENTITIES.

5.6.3 The simulation management computer

The simulation management function is used by exercise management personnel to manage the simulation, such as starting and stopping the exercise and requesting or directing that a simulation application perform some action related to one of their entities.

Simulation management may be performed by any computer on the DIS network. The computer may host a simulation application dedicated to the task of simulation management, or the same or another simulation application on the same computer may also be creating entities (or other objects) and their associated PDUs for the synthetic environment.

A Simulation Manager (SM) is a simulation application that issues one of the following SIMAN or SIMAN-R PDUs:

a) Create Entity PDU
b) Remove Entity PDU
c) Start/Resume PDU
d) Stop/Freeze PDU
e) Action Request PDU
f) Data Query PDU
g) Set Data PDU
h) Create Entity-R PDU
i) Remove Entity-R PDU

j) Start/Resume-R PDU
k) Stop/Freeze-R PDU
l) Action Request-R PDU
m) Data Query-R PDU
n) Set Data-R PDU

An SM is not an entity or object, itself, and it only interacts with other simulation applications regarding the application itself or one of its entities. When it needs to indicate a specific entity to which a SIMAN PDU pertains, it will be contained in either the Originating or the Receiving ID field, as applicable, as there is no separate Entity ID field in a SIMAN PDU. An exercise may have one SM or multiple SMs. The exercise agreement may specify that only a specific SM, as identified by its Simulation ID, is allowed to issue certain SIMAN PDUs (e.g., the Start/Resume and Stop/Freeze PDUs).

5.6.4 Simulation management functions

Simulation management functions can be categorized as entity/exercise management and data management. Some simulation management actions are directed to a specific simulation application, an entity, or in some cases, to all simulation applications. For example, both the overall exercise or a specific entity may be initialized, started, or stopped. The general use of SIMAN PDUs is to allow exercise personnel to manage an exercise by requesting that certain actions be taken by another or all simulation applications, or by a specific entity owned by a simulation. In this case, any identification of the SM making the request will use the Simulation ID of the SM.

The PDUs required for simulation management are described in 5.6.5. See 5.6.6 for a discussion of the simulation management functions and how the PDUs shall be used to accomplish them.

5.6.5 Simulation Management PDUs

5.6.5.1 General

The PDUs described in 5.6.5.2 through 5.6.5.13.4 shall be used to perform simulation management functions.

5.6.5.2 Create Entity PDU

5.6.5.2.1 Purpose

The Create Entity PDU shall communicate a request by an SM to create a new entity for a DIS exercise. There are three methods whereby an SM can request that an entity be created. The requirements associated with each of these methods are specified in 5.6.6.2.2 through 5.6.6.2.4 and are summarized here:

a) *Method 1*. Entity creation, query, and initialization sequence (see 5.6.6.2.2).
b) *Method 2*. Entity creation and initialization sequence (see 5.6.6.2.3).
c) *Method 3*. Entity creation and acknowledgment sequence (see 5.6.6.2.4).

NOTE—Entities can enter an exercise without the use of the simulation management protocol (see 5.6.6.2).

5.6.5.2.2 Information contained in the Create Entity PDU

The Create Entity PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) A unique Request Identification number for the action being requested.

5.6.5.2.3 Issuance of the Create Entity PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).

b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).

c) The Request ID field shall be set to an initial value and then incremented by one for each subsequent issue of this type of SIMAN PDU.

The Create Entity PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.2.4 Receipt of the Create Entity PDU

Upon receipt of the Create Entity PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the receiving simulation can comply with the Create Entity PDU request, then it shall assign an entity identification number to the new entity as requested by the SM.

b) The receiving simulation shall then respond to the Create Entity PDU by issuing an Acknowledge PDU.

5.6.5.3 Remove Entity PDU

5.6.5.3.1 Purpose

The Remove Entity PDU shall communicate the removal of an entity from a DIS exercise. This PDU indicates to the receiving simulation that the entity is to be removed from the exercise.

NOTE—An entity can affect the removal of itself from an exercise by setting the Entity Appearance record State field (bit 23) to Deactivated (1) in the last Entity State PDU that it issues (see 5.3.2.4).

5.6.5.3.2 Information contained in the Remove Entity PDU

The Remove Entity PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).

b) A unique Request Identification number for the action being requested.

5.6.5.3.3 Issuance of the Remove Entity PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).

b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).

c) The Request ID field shall be set to an initial value and then incremented by one for each subsequent issue of this type of SIMAN PDU.

The Remove Entity PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.3.4 Receipt of the Remove Entity PDU

Upon receipt of the Remove Entity PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) The receiving simulation shall immediately cease simulating the entity identified in the Remove Entity PDU and shall remove it from the simulation exercise as quickly as possible.
b) The receiving simulation shall then acknowledge the receipt of the Remove Entity PDU by issuing an Acknowledge PDU.

The simulation entity that is being removed from the exercise shall issue at least one more Entity State PDU following the acknowledgment of the Remove Entity PDU. The last Entity State PDU issued by the deactivated entity shall have the Entity Appearance record State field (bit 23) set to Deactivated (1).

NOTE—Entities interacting with the removed entity will be able to recognize that the removed entity has left the exercise through the reception of the Entity State or Entity State Update PDU from the removed entity with the Entity Appearance record State field (bit 23) set to Deactivated (1) and the absence of further Entity State or Entity State Update PDUs for the removed entity.

5.6.5.4 Start/Resume PDU

5.6.5.4.1 Purpose

The Start/Resume PDU shall be issued by an SM to start or resume one or more simulations, or one or more entities (see 5.6.6.4 for additional information on simulation states).

5.6.5.4.2 Information contained in the Start/Resume PDU

The Start/Resume PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) A unique Request Identification number for the action being requested.
c) Real-World Time.
d) Simulation Time.

5.6.5.4.3 Issuance of the Start/Resume PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Request ID field shall be set to an initial value and then incremented by one for each subsequent issue of this type of SIMAN PDU.
d) The requirements related to time shall be as follows:
    1) The Start/Resume PDU may be used to start or resume an exercise regardless of whether Absolute or Relative Time is being used in order to support those simulations that require a Start/Resume PDU.

2) When an exercise is started without using the Start/Resume PDU and Absolute Time has been designated by simulation management, each simulation (site/application) shall set its host computer to use Absolute Time for the Timestamp.

3) Any simulation that can initiate a Start/Resume PDU containing a simulation time that is set to a Relative Time value shall also maintain UTC time in order to properly fill the Real-World Time field.

It is possible for a Start or Resume PDU to contain a Real-World Time that is ahead of the present Real-World Time as known to the host System Clock. For example, simulation management may desire for an exercise to start 5 min from the present Real-World Time. In this case, if the actual Real-World Time at the time the Start/Resume PDU is sent is, for example, 14:30:00, the Real-World Time field in the Start/Resume PDU would be set to 14:35:00.

The Start/Resume PDU should be sent far enough in advance that any receiving simulation will have time to comply. The maximum expected time between transmission and reception of PDUs under various conditions is described in IEEE Std 1278.2.

The Start/Resume PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.4.4 Receipt of the Start/Resume PDU

Upon receipt of the Start/Resume PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) A simulation may restrict the receipt of Start/Resume PDUs to those PDUs that originated from one or more designated SMs, or it may discard received Start/Resume PDUs if it is in the overall simulation management role.

b) A simulation shall discard a Start/Resume PDU if it is currently active in an exercise, not paused, and is issuing or receiving PDUs.

c) The receiving simulation shall acknowledge the receipt of the Start/Resume PDU by issuing an Acknowledge PDU.

d) When an exercise is using Relative Time, any simulation that is joining or rejoining the exercise shall not start transmitting PDUs and interacting with data until either (1) it receives an appropriate Start/Resume PDU or (2) there is coordination between the SM and the simulation to provide the necessary Relative Time value and a manual procedure to activate the simulation at the specified Relative Time that matches a Real-World Time.

e) Each simulation shall protect itself from a Start/Resume PDU that is not applicable to it or that is received after it is already active in the exercise.

f) A simulation shall set its local simulation time to the time specified in the Simulation Time field when the local host System Clock reaches the time specified in the Real-World Time field.

g) If the Real-World Time field contains a time that is earlier than the present real-world time, the receiving application shall start or resume immediately. This condition may be considered an error that may be reported with an Event Report PDU.

5.6.5.5 Stop/Freeze PDU

5.6.5.5.1 Purpose

The Stop/Freeze PDU shall be issued by an SM to stop or freeze one or more simulations, or stop or freeze one or more entities that are in the Simulating State (see 5.6.6.4 for additional information on Simulation states.) The Stop/Freeze PDU should be sent far enough in advance that any receiving simulation will have time to comply. The maximum expected time between transmission and reception of PDUs under various conditions is described in IEEE Std 1278.2.

5.6.5.5.2 Information contained in the Stop/Freeze PDU

The Stop/Freeze PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) A unique Request Identification number for the action being requested.
c) Real-World Time at which the stop or freeze is to occur.
d) The reason why the Stop/Freeze PDU is being issued.
e) The behavior to be performed if a frozen state is indicated by the reason.

5.6.5.5.3 Internal state of a frozen entity

How the internal state of a simulation or a specified entity is affected by a stopped internal simulation clock is outside the scope of this standard, except that no reaction to a Nonsimulation Management/Simulation Management with Reliability PDU is permitted by the entity or all entities of the simulation(s) as specified by the value of the Receiving ID field.

5.6.5.5.4 Issuance of the Stop/Freeze PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Request ID field shall be set to an initial value and then incremented by one for each subsequent issue of this type of SIMAN PDU.
d) The Reason field shall be set to an appropriate reason.
e) The Frozen Behavior field shall be set to the appropriate value.

The Stop/Freeze PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.5.5 Receipt of the Stop/Freeze PDU

Upon receipt of the Stop/Freeze PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) The receiving simulation shall acknowledge the receipt of the Stop/Freeze PDU by issuing an Acknowledge PDU.

b) The receiving simulation(s) or the specified entity shall leave the Simulating state at the real-world time indicated in the PDU.

c) The stop/freeze condition may be temporary or indefinite based on the reason for stop/freeze.

d) If the Real-World Time field contains a time that is earlier than the present real-world time, the receiving application shall stop or freeze immediately. This condition may be considered an error that may be reported with an Event Report PDU.

5.6.5.6 Acknowledge PDU

5.6.5.6.1 Purpose

The Acknowledge PDU shall be used to acknowledge the receipt of a Create Entity PDU, a Remove Entity PDU, a Start/Resume PDU, a Stop/Freeze PDU, or an IO Action PDU. This PDU verifies to the sending simulation the receipt of the issued PDU. The Acknowledge PDU shall be sent as soon as possible after the receipt of the above PDUs.

5.6.5.6.2 Information contained in the Acknowledge PDU

The Acknowledge PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).

b) A Request Identification number that matches the Request ID received in the PDU to which this acknowledgment is responding.

c) A flag to indicate the type of PDU message to which the acknowledgment pertains.

d) A response flag to indicate the level to which the simulation or its entity was able to comply with the action being acknowledged.

5.6.5.6.3 Issuance of the Acknowledge PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set as follows:

    1) The Simulation Address in the Originating ID field shall be set to the Simulation Address of the simulation application issuing the PDU.

    2) The Reference Number in the Originating ID field shall be set as follows:

        i) If this is an acknowledgment of a Create Entity PDU that included a Special Create Entity ID in the Receiving ID field, it shall be set to the Reference Number of the Entity ID that was assigned by the simulation to the entity that was created.

        ii) Otherwise it shall be set to the Reference Number of the Receiving ID field in the PDU being acknowledged.

b) The Receiving ID field shall be set to the Originating ID field in the PDU being acknowledged.

c) The Request ID field shall match the Request ID from the PDU that is being acknowledged.

d) The Acknowledge Flag field shall indicate the type of PDU being acknowledged (see [UID 69]).

e) The Response Flag field shall be set to either Able to Comply (1) or Unable to Comply (2).

The Acknowledge PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.6.4 Receipt of the Acknowledge PDU

Upon receipt of the Acknowledge PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving SM shall note that the simulation has successfully received the previous PDU transmission.

5.6.5.7 Action Request PDU

5.6.5.7.1 Purpose

The Action Request PDU shall be issued by an SM to request that a specific action be performed. Information required to perform the requested action shall also be included in this PDU.

5.6.5.7.2 Information contained in the Action Request PDU

The Action Request PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) A unique Request Identification number for the action being requested.
c) Identification of the specific action to be taken.
d) The data required for performing the requested action.

5.6.5.7.3 Issuance of the Action Request PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Request ID field shall be set to an initial value and then incremented by one for each subsequent issue of this type of SIMAN PDU.
d) The Action ID field shall be set to an appropriate action value.
e) A Datum specification record shall be included with applicable data (see 6.2.18).

The Action Request PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.7.4 Receipt of the Action Request PDU

Upon receipt of the Action Request PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) The receiving simulation(s) or the specified entity/entities shall implement the request action if possible.
b) The receiving simulation shall then acknowledge the receipt by issuing an Action Response PDU.

5.6.5.8 Action Response PDU

5.6.5.8.1 Purpose

The Action Response PDU shall be used to acknowledge the receipt of an Action Request PDU. This PDU shall provide information on the status of the request and may also be used to provide additional information depending on the type of action requested.

5.6.5.8.2 Information contained in the Action Response PDU

The Action Response PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) The Request Identification number for the action being processed.
c) Status of the action request.
d) Datum values that may be pertinent as a by-product of the action request.

5.6.5.8.3 Issuance of the Action Response PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set as follows:
    1) The Simulation Address in the Originating ID field shall be set to the Simulation Address of the simulation application issuing the PDU.
    2) The Reference Number in the Originating ID field shall be set to the Reference Number of the Receiving ID field in the Action Request PDU being acknowledged.
b) The Receiving ID field shall be set to the Originating ID field in the Action Request PDU being acknowledged.
c) The Request ID field shall be set to the Request ID received in the Action Request PDU to which this response is being made.
d) The Request Status field shall be set to an appropriate status value.
e) A Datum specification record shall be included with applicable data (see 6.2.18).

The Action Response PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.8.4 Receipt of the Action Response PDU

Upon receipt of the Action Response PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the originator of the Action Request PDU shall note that the action request was received and the status of that request.

5.6.5.9 Data Query PDU

5.6.5.9.1 Purpose

The Data Query PDU shall be used to communicate a request for data.

5.6.5.9.2 Information contained in the Data Query PDU

The Data Query PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) A unique Request Identification number for this data query.
c) An indication of the time interval between issuing the requested data or that the data is only to be sent once.
d) The specific data contained in the query.

5.6.5.9.3 Issuance of the Data Query PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Request ID field shall be set to an initial value and then incremented by one for each subsequent issue of this type of SIMAN PDU.
d) The Time Interval field shall specify the time interval between issuance of requested Data PDUs or shall be set to zero to indicate that the Data PDU shall be issued once in response to this data query and not at any previously specified time interval.
e) A Datum specification record shall be included with applicable data (see 6.2.18).

The Data Query PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.9.4 Receipt of the Data Query PDU

Upon receipt of the Data Query PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving simulation shall respond by issuing a Data PDU containing the value or values (if any) of the datum requested in the Data Query PDU that the addressed simulation or entity is capable of replying to.

5.6.5.10 Set Data PDU

5.6.5.10.1 Purpose

The Set Data PDU shall be used by the SM to set or change certain parameters.

5.6.5.10.2 Information contained in the Set Data PDU

The Set Data PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) A unique Request Identification number for this data query.
c) The data to be set.

5.6.5.10.3 Issuance of the Set Data PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).

b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).

c) The Request ID field shall be set to an initial value and then incremented by one for each subsequent issue of this type of SIMAN PDU.

d) A Datum specification record shall be included with applicable data (see 6.2.18).

The Set Data PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.10.4 Receipt of the Set Data PDU

Upon receipt of the Set Data PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the addressed simulation or entity shall set the appropriate parameters as specified in the Set Data PDU. It shall be up to the receiver to determine which (if any) parameters described in the Set Data PDU can be set. The receiving simulation shall then respond with a Data PDU.

5.6.5.11 Data PDU

5.6.5.11.1 Purpose

The Data PDU shall be used to respond to a Data Query PDU or Set Data PDU. This PDU allows the recipient to provide the information requested in a Data Query PDU or to confirm the information received in a Set Data PDU.

Data PDUs may be issued at a periodic rate or may only be sent once. This rate can be set in the Data Query PDU (see 5.6.5.9.2).

Data PDUs may be issued without first receiving a Set Data or Data Query PDU. The contents of the Data PDU in this case can be determined by the originator. Error reporting shall be implemented in the Event Report PDU (see 5.6.5.12).

5.6.5.11.2 Information contained in the Data PDU

The Data PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).

b) A unique Request Identification number for this data query.

c) Specific data being sent.

5.6.5.11.3 Issuance of the Data PDU

The following specific requirements shall be met:

a) When the Data PDU is issued in response to a Data Query PDU:

1) The Originating ID field shall be set as follows:
    i) The Simulation Address in the Originating ID field shall be set to the Simulation Address of the simulation application issuing the PDU.
    ii) The Reference Number in the Originating ID field shall be set to the Reference Number of the Receiving ID field in the Data Query PDU to which the simulation application is responding.
2) The Receiving ID field shall be set to the Originating ID field in the Data Query PDU to which the simulation application is responding.
3) The Datum Specification record shall be used with the value or values (if any) of the datum requested in the Data Query PDU that the simulation is capable of replying to. Parameters to which the receiving entity cannot comply shall not be included in the Data PDU response.
4) It shall be issued either once or at the specified interval as defined by the received Time Interval field value.

b) When the Data PDU is issued in response to a Set Data PDU:
    1) The Originating ID field shall be set as follows:
        i) The Simulation Address in the Originating ID field shall be set to the Simulation Address of the simulation application issuing the PDU.
        ii) The Reference Number in the Originating ID field shall be set to the Reference Number of the Receiving ID field in the Set Data PDU to which the simulation application is responding.
    2) The Receiving ID field shall be set to the Originating ID field in the Set Data PDU to which the simulation application is responding.
    3) Verify the receipt of the Set Data PDU by returning the parameter values that were set in response to the Set Data PDU:
        i) Parameters that were set in the simulation to the same values as in the Set Data PDU shall be set to those values in the Data PDU.
        ii) Parameter values that were set to different values in the simulation than requested in the Set Data PDU shall be set to their actual values in the Data PDU.
        iii) Parameters to which the receiving entity cannot comply shall not be included in the Data PDU response.

c) When the Data PDU is issued independent of receiving a Data Query or Set Data PDU, it shall be sent as determined by the simulation except that the following requirements shall be met:
    1) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
    2) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
    3) The Request ID field shall be set to the next available incremental value.
    4) The Data PDU may be issued once or at some specified interval defined by the simulation.

The Data PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.11.4 Receipt of the Data PDU

Upon receipt of the Data PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving simulation shall record the information for simulation management purposes.

5.6.5.12 Event Report PDU

5.6.5.12.1 Purpose

The Event Report PDU shall be used to communicate the occurrence of a significant event. What constitutes a significant event may have been set previously by the SM, or it may be internal to the simulation.

5.6.5.12.2 Information contained in the Event Report PDU

The Event Report PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) The type of event that has occurred.
c) Event Report data.

5.6.5.12.3 Issuance of the Event Report PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Event Type field shall be set to an appropriate event.
d) A Datum Specification record shall be included with information about the event (see 6.2.18).

The Event Report PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.12.4 Receipt of the Event Report PDU

Upon receipt of the Event Report PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving simulation may process, record, or ignore the PDU.

5.6.5.13 Comment PDU

5.6.5.13.1 Purpose

The Comment PDU shall be used to input a message into a data stream either for use as a comment, an error, a test message, or a place holder in a sequentially stored exercise.

5.6.5.13.2 Information contained in the Comment PDU

The Comment PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82).
b) Comment data.

5.6.5.13.3 Issuance of the Comment PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).

b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).

c) The comment data shall be contained in a Datum Specification record (see 6.2.18).

The Comment PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulations that are members of the addressed multicast group.

5.6.5.13.4 Receipt of the Comment PDU

Upon receipt of the Comment PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving simulation may process, record, or ignore the PDU.

5.6.6 Entity/exercise management

5.6.6.1 General

Management of an entity or exercise includes the capability to create new entities, initialize or change entity or exercise parameters, start or stop an entity or exercise, request an entity to perform a specific action, and record significant entity data or exercise events. These actions are described in 5.6.6.2 through 5.6.6.7.

5.6.6.2 Entity creation

5.6.6.2.1 General

The Simulation Management protocol provides three ways to create a new entity. The first method allows the SM to establish the identification (using Create Entity PDU) of the new entity, query for data about the new entity (using Data Query PDU), and set initial parameters for the new entity (using Set Data PDU). The second method is similar to the first method except the SM does not query for data. The third method is even more streamlined, requiring only the Create Entity PDU. It should be noted that the third method requires that certain database information be established in advance of the exercise start, whereas the first method allows the entire creation and initialization process to proceed with little information established in advance. The three methods of entity creation are described in 5.6.6.2.2 through 5.6.6.2.4.

Entities may also enter an exercise without being created by means of the Simulation Management protocols. An entity enters an exercise by exchanging PDUs (Entity State PDUs, for example) with other entities in the DIS exercise. For instance, a simulation can create a munition entity that requires tracking information by starting to issue Entity State PDUs for the munition.

5.6.6.2.2 Entity creation, query, and initialization

To create, query, and initialize a new entity, either for an existing exercise or for a new exercise, the SM shall begin by issuing a Create Entity PDU to the Simulation Application (SA) that will be controlling the simulation entity. The receiving simulation application shall respond with an Acknowledge PDU. These actions simply assign an identification number to a new entity.

The SM shall then request that certain data be issued by the simulation application controlling the new entity. This is accomplished by issuing a Data Query PDU. The simulation application shall respond by sending the requested data using a Data PDU.

The SM shall then send necessary initialization information to the new entity using a Set Data PDU. The receipt of the Set Data PDU by the entity shall be indicated by the return of a Data PDU.

The process of entity creation, query, and initialization is illustrated in Figure 11.

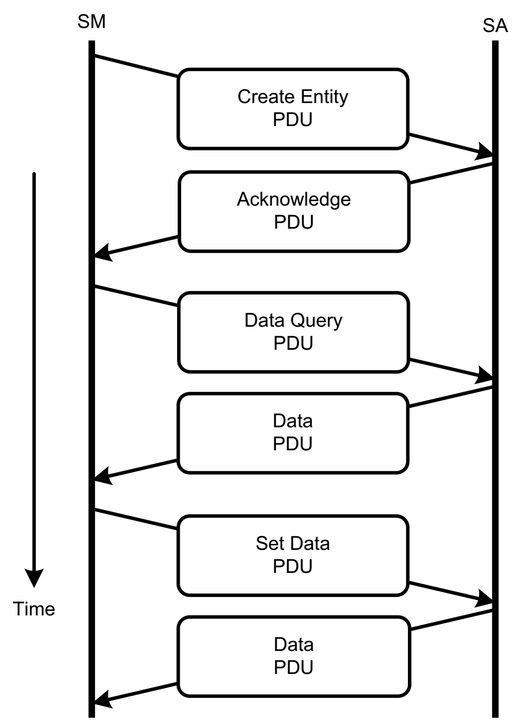


Figure 11—Entity creation, query, and initialization

5.6.6.2.3 Entity creation and initialization

To create and initialize a new entity, the SM shall begin by issuing a Create Entity PDU to the simulation application that will be controlling the simulation entity. The receiving simulation application shall respond with an Acknowledge PDU. These actions simply assign an identification number to a new entity.

It is assumed that the SM has all the necessary data it needs for the new entity (e.g., type of entity and characteristics). This information shall be established offline and prior to the entity creation. The SM shall then initialize the new entity by issuing Set Data PDU to the simulation application controlling the new entity. The receipt of the Set Data PDU by the entity shall be indicated by the return of a Data PDU.

The process of entity creation and initialization is illustrated in Figure 12.

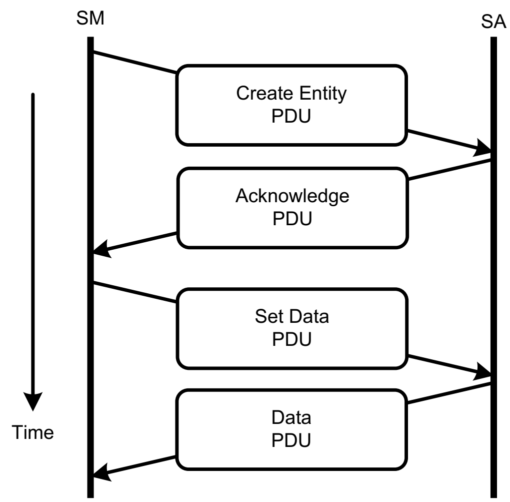


Figure 12—Entity creation and initialization

5.6.6.2.4 Entity creation

In the case when necessary entity data and initialization data have already been established offline and prior to the exercise, it is possible to create a new entity by assigning a particular entity identification. To create a new entity, the SM shall issue a Create Entity PDU to the simulation application that will be controlling the simulation entity. The receiving simulation application that will contain the new entity shall respond with an Acknowledge PDU.

The process of entity creation is illustrated in Figure 13.

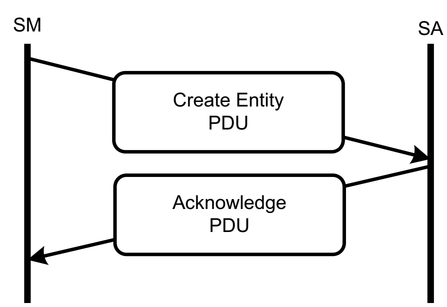


Figure 13—Entity creation

5.6.6.3 Changing entity parameters

Parameters within a particular entity shall be changed by the SM by the issue of a Set Data PDU. For example, during initialization, the Set Data PDU requests that an entity sets (or changes) certain parameters of its internal state to specified values. As in initialization, the receiving entity shall respond to the Set Data PDU by issuing a Data PDU. This will serve as an acknowledgment to the SM that the correct changes were made. Another Set Data PDU may be issued by the SM if the first was incorrectly received.

5.6.6.4 Starting or stopping an entity

5.6.6.4.1 General

An entity, or the simulation application supporting an entity, shall be in one of three states—Wait, Stopped/Frozen, or Simulating. An entity does not participate in the exercise (i.e., does not transmit Entity State PDUs) while in the Wait state, although it is ready to be created and then initialized by the simulation application controlling it. The entity may also be in the Stopped/Frozen state when it is not simulating, but it can be started at any time. Finally, an entity may be in the Simulating state; in which case, it is actually participating in an exercise.

These states are represented in Figure 14.

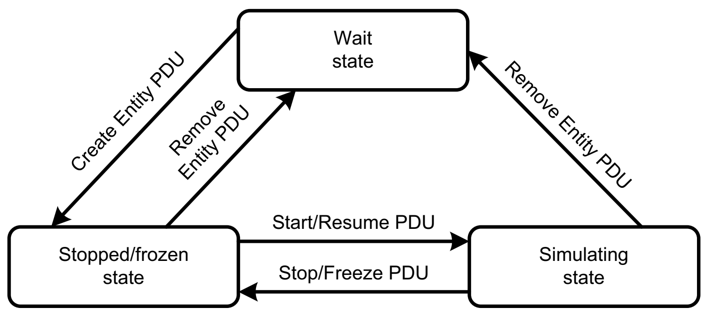


Figure 14—Entity states in simulation management

5.6.6.4.2 Change of frozen behavior

An entity shall change its frozen behavior based on the Frozen Behavior field of the latest Stop/Freeze PDU received as indicated by the Request ID field. The Stop/Freeze PDU shall cause an entity to transmit or not transmit PDUs, update or freeze its internal simulation clock, and react to or ignore received PDUs.

An entity is in the Wait state until it is created and initialized (see 5.6.6.2). At this point, the entity is in the Stopped/Frozen state. To start an entity, an SM shall issue a Start/Resume PDU to the entity to be started. The receiving entity shall respond with an Acknowledge PDU. Similarly, to stop or freeze a simulating entity, an SM shall issue a Stop/Freeze PDU to the entity to be stopped. The receiving entity shall likewise respond with an Acknowledge PDU.

These actions are represented in Figure 15.

5.6.6.5 Removing an entity from an exercise

An entity may be removed from an exercise by an SM. To remove the entity, an SM shall issue a Remove Entity PDU to the entity to be removed. The simulation application controlling the entity shall respond with an Acknowledge PDU and shall cease simulating its entity. A removed entity shall issue a *final* Entity State PDU with the Appearance record State field (bit 23) set to Deactivated (1).

These actions are represented in Figure 16.

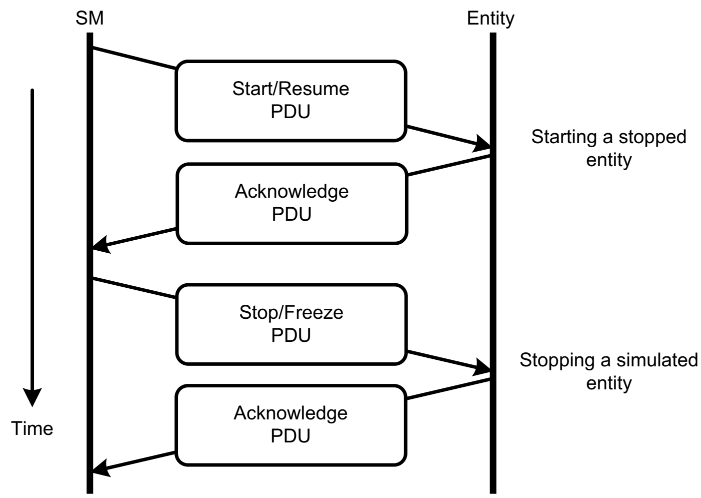


Figure 15—Starting/stopping an entity

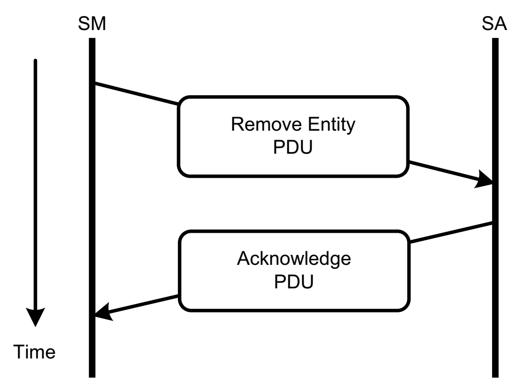


Figure 16—Removing an entity

5.6.6.6 Requesting an entity or simulation to perform an action

An entity or simulation may be requested to perform a specific action by the SM. In addition to requesting the action, the SM may provide needed information for performance of the requested action.

To request an action, the SM shall issue an Action Request PDU to the entity to perform the action. Upon receipt of the Action Request PDU, the receiving entity or simulation shall act upon the request and respond with an Action Response PDU.

This interaction is represented in Figure 17.

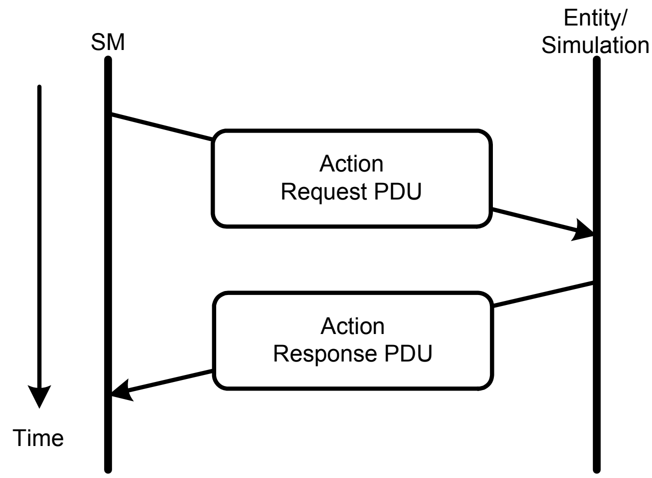


Figure 17—Action request/response

5.6.6.7 Event recording

To keep record of certain events that may occur during the course of a DIS exercise, an Event Report PDU shall be used in the reporting of such events (5.6.5.12.2). The SM may set certain parameters such that a particular event will initiate the issuance of an Event Report PDU. When that event occurs, the entity involved issues an Event Report PDU to report the event.

5.6.7 Data management

5.6.7.1 General

In addition to managing entities and the exercise, data management may be accomplished using the Simulation Management PDUs.

5.6.7.2 Request for data

An SM shall request data concerning the internal state of an entity or simulation by issuing a Data Query PDU. The receiving entity or simulation shall respond by providing the requested data in a Data PDU.

These actions are represented in Figure 18.

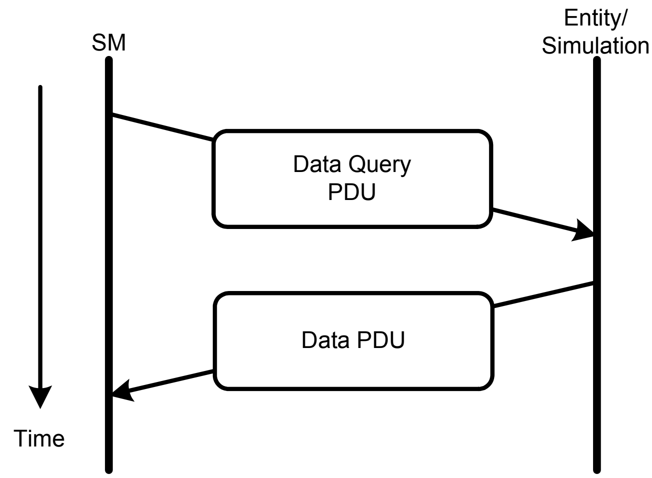


Figure 18—Requesting entity/simulation data

5.6.7.3 Setting or changing internal state values

Internal state information can be set or modified by the SM. This shall be accomplished by the SM issuing a Set Data PDU indicating the data to be changed or set. It shall be up to the receiver of the Set Data PDU to determine which (if any) parameters described in the Set Data PDU it can set. The receiver shall respond by issuing a Data PDU. The Data PDU shall verify the receipt of the Set Data PDU by returning the parameter values that were set in response to the Set Data PDU. Parameters that were set to the same values as in the Set Data PDU shall be set to those values in the Data PDU. Parameter values that were set to different values than requested shall be set to their actual values in the Data PDU. Parameters to which the receiver cannot comply shall not be included in the Data PDU response. This serves as an acknowledgment to the SM. If the SM determines that the original request was incorrectly received, another Set Data PDU may be issued by the SM. The assurance that the data values contained in the Set Data PDU are within the valid range for the specified variable, are of the correct data structure, are properly named, and so on, is outside the scope of this standard.

These actions are represented in Figure 19.

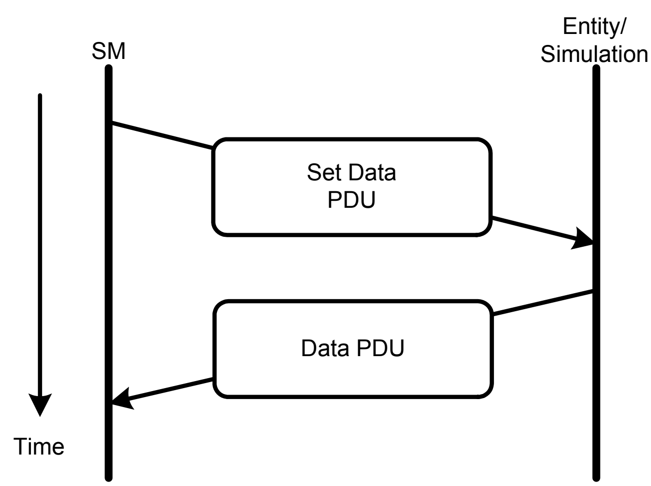


Figure 19—Setting/changing internal state values

5.6.7.4 Entity reconstitution

A killed or damaged entity shall allow its reconstitution by an SM when requested. If the same entity (in the exercise) is to be reconstituted, then the unique entity identification shall remain the same during this process. If the new reconstituted entity is a different entity (in the exercise), then the entity is not reconstituted but removed, and a new different entity would be created following 5.6.6.2.4. To reconstitute an entity, an SM shall issue a Stop/Freeze PDU. The receiving entity shall respond with an Acknowledge PDU. The entity shall not be removed but shall be in the Stopped/Frozen state as in Figure 14. The SM shall then issue a Set Data PDU to reset the entity parameters. The receiving entity shall respond with a Data PDU. The SM shall then issue a Start/Resume PDU for the receiving entity to rejoin the exercise. The receiving entity shall respond with an Acknowledge PDU.

These actions are represented in Figure 20.

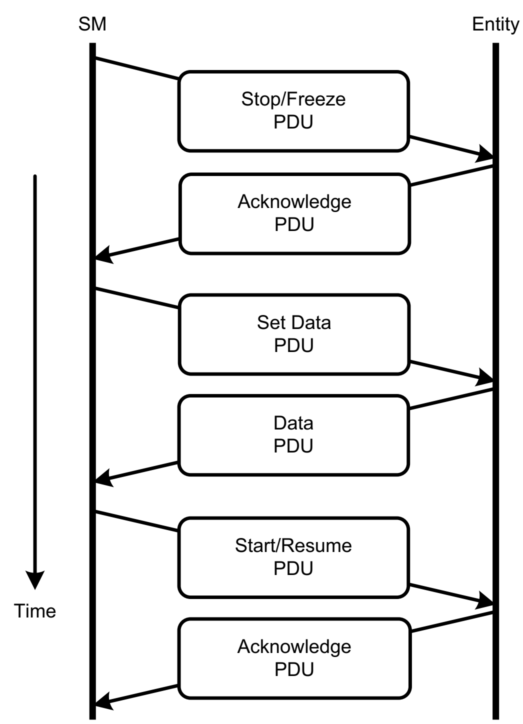


Figure 20—Reconstituting an entity

5.7 Distributed emission regeneration

5.7.1 General

The content and procedure for the use of Distributed Emission Regeneration (DER) in a DIS exercise is established below.

5.7.2 Distributed emission regeneration approach

The following approach shall be followed for DER in DIS:

a) Simulation applications with emitters shall simulate their emitter and shall output predefined, realistic operational parameters via the DIS communications network.

b) Simulation applications with receivers shall regenerate the transmitter signal to the fidelity level required by that particular receiving simulation application. For electromagnetic emissions, regeneration shall be accomplished by using the operational parameters provided in the EE PDU along with information from stored databases that describe the transmitter capabilities (i.e., beam scan patterns, etc.).

c) Scan patterns shall be regenerated based on beam-center and field-of-regard data sent from the transmitter, coupled with receiver-stored database parameters.

d) The DER approach for electromagnetic emissions contains provisions to allow emitters to communicate information on entities that are being tracked by the system. This capability allows tracking truth to be established for such purposes as ensuring a fair fight in multifidelity applications and supporting monitoring systems. This capability is also required because the beam movement (or array control) in complex emitter systems (such as phased-array systems) occurs too fast to issue separate PDUs for each beam position.

e) The DER approach for electromagnetic emissions allows radar jammers to communicate information to the emitter(s) for which the jamming is intended or expected. This capability is provided to support indirect jamming techniques.

5.7.3 Electromagnetic Emission (EE) PDU

5.7.3.1 Purpose

The EE PDU shall be used to communicate active electromagnetic emissions, including radar and radar-related electronic warfare (e.g., jamming). Exceptions include IFF interrogations and replies, navigation aids, voice, beacon and data radio communications, directed energy weapons, and laser ranging and designation systems, which are handled by other PDUs.

5.7.3.2 Information contained in the EE PDU

The EE PDU shall contain the following information:

a) Identification of the emitting entity.

b) Identification of the event.

c) Number of emitter systems described in the PDU. When associated with the EE PDU, the terms *emitter*, *emitter system*, and *system* refer to the device generating electromagnetic emissions and may be used interchangeably.

d) An emitter system data section to provide information for one or more emitter systems associated with the entity, including:

    1) Length of the emitter system data provided in the PDU.

    2) Number of beams described in the emitter system data section.

    3) Emitter system identification, including the emitter name, function for which the emitter system was designed, and emitter number.

    4) Location relative to the entity, providing the source location of all beams on the emitter system.

    5) For each emitter system, a beam data section to provide information for one or more beams associated with the emitter system, including:

        i) Length of the beam data in the PDU.

        ii) Beam number.

iii) Beam parameter index, with which receivers reference detailed local database parameters.

iv) Fundamental parametric data, describing basic characteristics of a beam that may vary for a given beam function. Fundamental parameters are defined in the EE Fundamental Parameter Data record (see 6.2.22).

v) Beam data, indicating the geometry of each emitting beam. Beam data are defined in the Beam Data record (see 6.2.11).

vi) Beam function identifier, indicating the purpose of each emitting beam.

vii) Number of targets for which information is provided in the track/jam data section of the beam.

viii) High-Density Track/Jam field to indicate that a large number of entities may be under track or may be the target of jamming.

ix) Jamming technique to define electronic warfare methods being applied (e.g., noise jamming and Velocity Gate Pull-Off).

x) Beam status information for each beam (e.g., the beam is active or deactivated).

xi) For each beam provided in the emitter system, a track/jam data section to list one or more entities tracked or under illumination, or beams that are targets of jamming (see 6.2.90).

5.7.3.3 Issuance of the EE PDU

5.7.3.3.1 General

Simulation applications shall issue EE PDUs: (1) when significant changes occur in any active emitter or (2) when the reference time elapsed since the emitter's last published EE PDU exceeds a predetermined timeout. Specific issuance instructions are provided below.

5.7.3.3.2 Issuance timing requirements

A simulation shall issue an EE PDU when any of the following occur:

a) An inactive emitter system becomes active or an active emitter system becomes inactive. An active emitter system is one that has at least one active beam. An active beam is one that is emitting or one that is in transition from emitting to not emitting (see 5.7.3.6).

b) Emitter system operational parameters change. Operational parameters include number of emitter systems and beams, location with respect to entity, beam parameter index, beam function, number of targets, jamming technique, and track/jam data records.

c) Changes in beam fundamental parameters contained in the PDU fields exceed specified thresholds. Fundamental parameters include Effective Radiated Power (ERP), frequency, frequency range, Pulse Repetition Frequency (PRF), and Pulse Width (PW). Fundamental parameter thresholds are identified by the symbolic names EE_ERP_THRSH, EE_FREQ_THRSH, EE_FRNG_THRSH, EE_PRF_THRSH, and EE_PW_THRSH, respectively (see 6.1.8).

d) Changes in beam geometry descriptors exceed specified thresholds. Beam geometry descriptors include beam azimuth center, beam azimuth sweep, beam elevation center, and beam elevation sweep. The azimuth and elevation thresholds shall be identified by the symbolic names EE_AZ_THRSH and EE_EL_THRSH, respectively. (See 4.2.8.3 for parameter details and default values.)

e) A predetermined timeout has elapsed since issuing the last emitter system update. A separate heartbeat timer shall be maintained for each emitter system associated with an entity (see 5.7.3.3.4). The EE PDU heartbeat parameter shall be identified by the symbolic name HBT_PDU_EE (see 6.1.8).

5.7.3.3.3 Other issuance requirements

For each emitting entity, EE PDUs shall be issued as follows:

a) Construction of the EE PDU shall not cause the length of the PDU to exceed the maximum PDU size (see 6.3.3).

b) When an entity is destroyed but still active [Entity Appearance record Damage field (bits 3 to 4) set to Destroyed (3) but still issuing Entity State PDUs], each simulation application that models emitter systems associated with that entity shall issue final EE PDUs that turn off all active beams for all emitter systems modeled.

c) When the number of tracked or illuminated entities or the number of beams being jammed exceeds a certain threshold, then the High-Density Track/Jam field shall be set to Selected (1), zero shall be entered in the Number of Targets field, and no Track/Jam Data records shall be included in the track/jam data field. The threshold parameter shall be identified by the symbolic name EE_HIGH_DENSITY_THRSH (see 6.1.8 for parameter details and default values).

d) Continuous wave emissions shall be represented by 0.0 values in both the Pulse Repetition Frequency and Pulse Width fields.

e) *Specific Field Requirements*

    1) System Data Length. When the size of an active emitter system is large, this field shall be set to zero. Large emitter systems are defined in 7.6.2.

    2) Emitter Name. Once established for an exercise, the Emitter Name for each emitter system shall not be changed during that exercise.

    3) Emitter Function. Once established for an exercise, the Emitter Function for each emitter system shall not be changed during that exercise.

    4) Emitter Number. No emitter system shall be assigned a number containing NO_EMITTER or ALL_EMITTERS. Once established for an exercise, the Emitter Number for each emitter system shall not be changed during that exercise. Each Emitter Number shall be unique for a given entity.

    5) Beam Data Length. When the size of an active beam is large, this field shall be set to zero. Large beams are defined in 7.6.2.

    6) Beam Number. No beam shall be assigned a number containing NO_BEAM or ALL_BEAMS. The mechanism by which beam numbers are assigned is outside the scope of this standard. Beam numbers for active beams shall not be changed during an exercise.

    7) Beam Sweep Sync. The pattern and origin data shall be derived from database parameters indexed by a primary key consisting of the Emitter Name in conjunction with the Beam Parameter Index. If a scan pattern is not applicable (such as for stationary beams or pseudo-randomly scanning phased array radars), this field shall contain the value NO_PATTERN. If not employed, this field shall contain the value NO_PATTERN (see 6.1.8). A change in beam sweep sync value alone shall not require issuance of an EE PDU (see 6.2.11).

    8) Beam Function. This field may be updated to reflect changes in emitter mode.

    9) Jamming Technique. Subclause 5.7.3.8 describes the use of this field when jamming is indicated. When jamming is not indicated, the field shall be represented by NO_KIND:NO_CATEGORY: NO_SUBCAT: NO_SPECIFIC (see 6.1.8).

    10) Track/Jam Data. This field shall identify the targets for each beam. When the Number of Targets value is zero, this field shall be omitted. Otherwise, the number of Track/Jam Data records shall match the value in the Number of Targets field. No Entity ID field in a Track/Jam Data record shall be assigned a number equal to NO_SITE, NO_APPLIC or NO_ENTITY (see 5.7.3.7 and 5.7.3.8).

    11) State Update Indicator. Use of this field is optional. If it is not used, it shall be set to Heartbeat Update (0).

5.7.3.3.4 Issuance methods

Simulation applications may issue EE PDUs using a Complete-Entity, Complete-Emitter, or Complete-Beam issuance method. Simulation applications may change from one issuance method to another as required. This flexibility provides support for both simple and complex modeling environments:

a) In the Complete-Entity issuance method, a single EE PDU fully describes all active emitters associated with an entity. When sending an update to an EE PDU, the simulation application issues a single PDU describing all active beams for all active emitter systems. Figure 21 illustrates the Complete-Entity issuance method for a bomber flight simulator. The simulator models three emitters: a Fire Control Radar (FCR) with three beams, a Tail Warning Radar (TWR) with one beam, and a self-protection jammer with two beams. Large boxes represent each emitter system provided in the EE PDU. Small boxes indicate beams within each emitter. Small filled boxes indicate beams for which data have changed since issuance of the last EE PDU describing that emitter. For the Complete-Entity issuance method, all EE PDUs contain complete data describing all active beams for all active emitter systems.

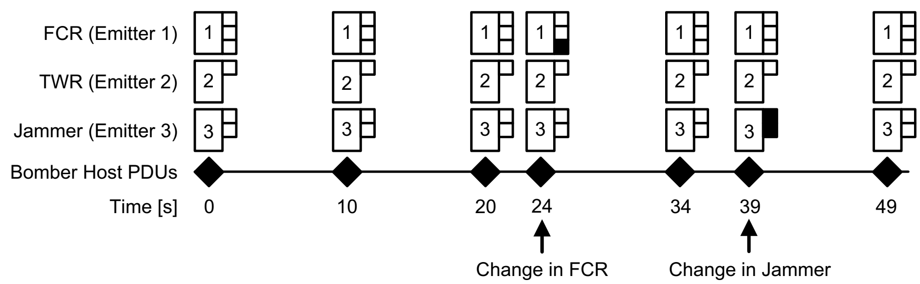


Figure 21—Complete-Entity issuance method

b) In the Complete-Emitter issuance method, a single or multiple EE PDUs may be issued to describe the emitters associated with an entity, with each PDU fully describing all active beams associated with each emitter described in that PDU. In contrast with the Complete-Entity issuance method, the Complete-Emitter method does not require that all of an entity's active emitters be described in each EE PDU issued:

  1) In the case where a single simulation application models all emitters for a given entity, the Complete-Emitter issuance method involves issuance of a single series of EE PDUs. Emitter data changes result in issuance of PDUs describing only the emitters that have changed (see Figure 22). In the example shown, when the data for a single emitter change, the resulting PDU fully describes all active beams on that emitter, but it omits any data for the unchanged emitters. Thereafter, the simulation application issues EE PDUs for the changed emitter in accordance with heartbeat requirements [see item e) in 5.7.3.3.2].

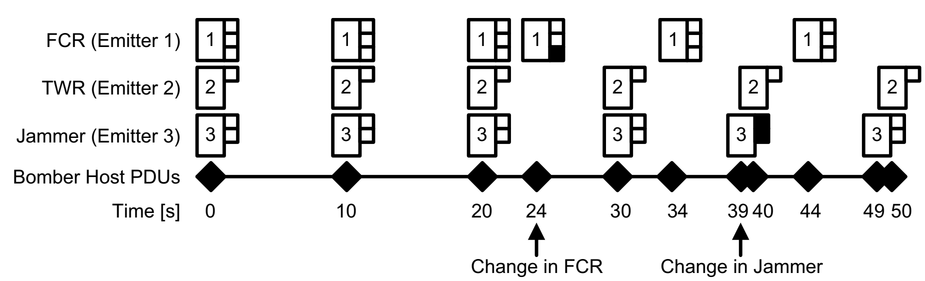


Figure 22—Complete-Emitter issuance method (single PDU series)

2) In the case where independent models or simulation applications are employed to model a given entity's emitters, the Complete-Emitter issuance method may involve issuance of more than one independent series of EE PDUs. In the example shown in Figure 23, the bomber's host simulation issues the first series of EE PDUs containing data for the entity's fire control radar and tail warning radar. An Electronic Warfare (EW) server simulation (potentially located at another physical location) issues a second series of PDUs that only contain data for the entity's self-protection jammer. When the data for emitters change, the resulting PDUs omit data for the unchanged emitters in the series (see Figure 23). Thereafter, the simulation application issues EE PDUs for the changed emitter in accordance with heartbeat requirements [see item e) in 5.7.3.3.2].

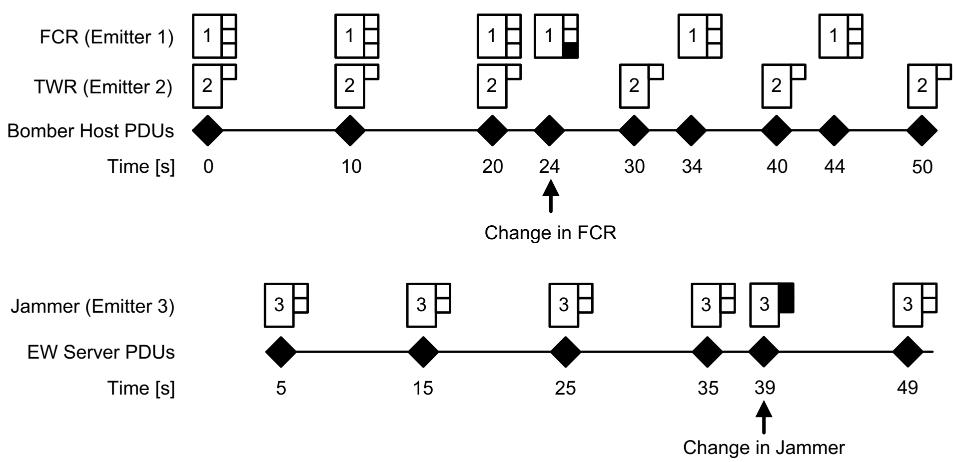


Figure 23—Complete-Emitter issuance method (multiple PDU series)

c) In the Complete-Beam issuance method, a single EE PDU describes a subset of the entity's active beams, with each beam fully described. This method supports multibeam emitters in cases where some beams update much more frequently than others. In the current example, the bomber's fire control radar may be modeled with one ground mapping beam, one air search beam, and one air tracking beam. Compared with the relatively stable scan volumes of the ground mapping and air search beams, the beam center of the air tracking beam may update rapidly. As shown in Figure 24, the Complete-Beam method allows frequent updates of the air tracking beam. Issuance of PDUs containing all beams is limited to conditions dictated by heartbeat requirements [see item e) in 5.7.3.3.2].

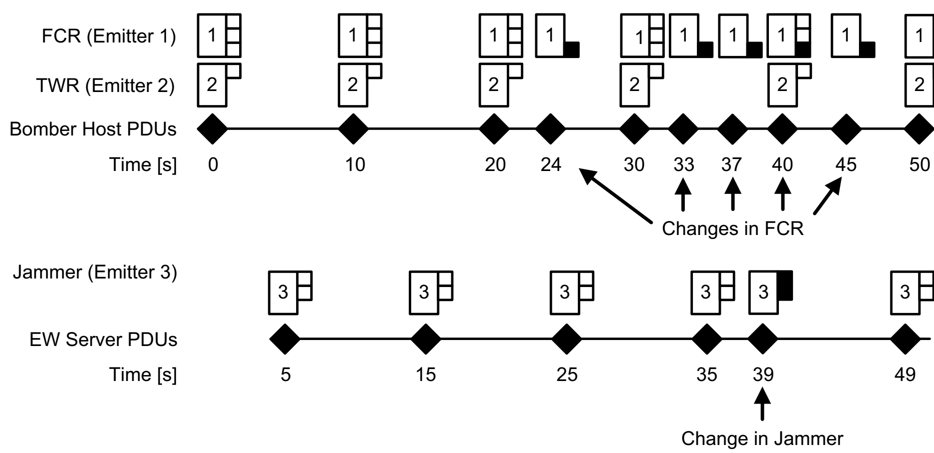


Figure 24—Complete-Beam issuance method

NOTE—In Figure 24, changes to select beams on the FCR emitter do not disrupt the FCR heartbeat pattern; yet simultaneous changes to all beams on the jammer result in "resetting the heartbeat timer" for the EW Server PDU series (see 4.2.6).

5.7.3.4 Receipt of the EE PDU

Upon receipt of an EE PDU, the receiving simulation application shall determine whether the emissions are detectable and use the information in the EE PDU to influence emission-detection equipment in the simulation appropriately. A receiving simulation shall update the information it stores locally about an emitter system with data provided in each EE PDU received.

The method used to update emitter systems is independent of the issuance method. For example, consider the example in Figure 24 where at 24 s, the bomber host simulator updates the air tracking beam on the fire control radar using the Complete-Beam issuance method. The PDU only contains data for one beam of the three active on the bomber, indicating that only the air tracking beam has changed since the last PDU. The Radar Warning Receiver (RWR) model on a fighter simulator need not compare previously stored values with the newly received values. Rather, the fighter simulator updates the RWR model with all data contained in the PDU (data for only one beam in this case). If the bomber had issued the EE PDU using the Complete-Emitter issuance method illustrated in Figure 23, the PDU would contain data for all three beams. As before, the fighter simulator would update the RWR model with all data contained in the PDU (data for three beams in this case).

The size of an emitter system or beam shall be considered large when the System Data Length or Beam Data Length value is zero, respectively. For large emitter systems, the receiving application shall calculate the size of the system using the Number of Beams and Number of Targets fields. Likewise, for large beams, the receiving application shall calculate the size of the beam using the Number of Targets field. See 5.7.3.3 and 7.6.2.

Simulation applications regenerating emissions shall maintain an accounting of the reference time elapsed since each emitter system has been updated with data from a received EE PDU. If a predetermined timeout has elapsed since receiving an EE PDU containing data for an emitter system, receiving simulations shall consider the emitter system to be inactive. The timeout shall be established at exercise start, although it may be changed during the exercise. The timeout parameter shall be established as the value of HBT_PDU_EE multiplied by HBT_TIMEOUT_MPLIER (see 6.1.8 for details of these two parameters and their default values).

When modeling the effects of jamming on a local beam, a simulation application shall interpret the Jamming Technique record in the received EE PDU and react according to the highest level of fidelity modeled locally (e.g., range gate pull off with inverse gain jamming). If the jamming technique provided in the PDU is not modeled, the simulation shall promote the technique to the most specific technique modeled locally (e.g., range gate pull off jamming, gate stealer jamming, or deception jamming).

5.7.3.5 Emission regeneration

Emissions shall be regenerated by the receiving simulation application based on data provided in the EE PDU. Higher fidelity regeneration may be based on additional parameters stored in the databases of each receiving simulation application. The EE PDU Emitter Name and Beam Parameter Index fields together shall provide the database primary key.

5.7.3.6 Beam and emitter system activation and deactivation

When a beam becomes active, the simulation modeling the beam shall issue an EE PDU with the Beam Status record Beam State field (bit 0) set to Active (0).

When a beam becomes inactive, the simulation modeling the beam shall issue a single EE PDU with the Beam Status record Beam State field (bit 0) set to Deactivated (1). Until the beam again becomes active, no subsequent EE PDUs shall contain data for the beam.

When an emitter system becomes active, the simulation modeling the system shall issue an EE PDU that includes the active system and its active beams.

When an emitter system becomes inactive, the simulation modeling the system shall issue an EE PDU that includes those beams changing from emitting to not emitting. Each such beam included shall have its Beam Status record Beam State field (bit 0) set to Deactivated (1). Until at least one beam associated with that system becomes active, no subsequent EE PDUs shall be issued containing that emitter system.

5.7.3.7 Track/jam targets

This subclause provides a mechanism for emitters to identify targets. This mechanism is intended to help receiving entities determine whether the EE PDU is of interest to the systems simulated by that entity. The number of emission targets for a given beam is defined as the number of entities tracked or illuminated or the number of beams targeted with jamming, as appropriate for the emitter beam's function. Definitions of tracking, illumination, and jamming-targeted beams are beyond the scope of this standard. Normally, each emission target is identified by a Track/Jam Data record (see 6.2.90). However, when the number of emission targets exceeds a certain threshold, Track/Jam Data records are omitted. In this case, receiving simulation applications may assume that all targets in the beam's scan volume are potentially tracked, illuminated, or jammed. The threshold parameter shall be identified by the symbolic name EE_HIGH_DENSITY_THRSH (see 6.1.8):

a) When the number of targets exceeds the high-density threshold:
   1) The Number of Targets field shall be set to zero.
   2) The High-Density Track/Jam field shall be set to Selected (1).
   3) The Track/Jam Data records shall be omitted.
b) Otherwise:
   1) The Number of Targets field shall reflect the number of entities tracked or illuminated or the number of beams targeted with jamming.

2) The High-Density Track/Jam field shall be set to Not Selected (0).
3) The Track/Jam Data records shall identify each target. When jamming is not indicated, the Emitter Number and Beam Number fields shall be set to NO_EMITTER and NO_BEAM, respectively.

5.7.3.8 Jamming

The presence of jamming shall be indicated by use of a beam function value representing jamming. Electronic Warfare methods shall be indicated by use of the Jamming Technique record (see 6.2.49). The Jamming Technique record provides a multiresolution method for indicating electronic warfare methods and is intended to help receiving entities determine whether the EE PDU is of interest to the systems simulated by that entity:

a) Jamming techniques are arranged in a hierarchical structure such that higher fidelity simulations may depict detailed representations of a technique (such as a combination technique called Range Gate Pull Off and Swept Square Wave). Lower fidelity simulations may depict the same technique in a more generic manner (such as Angle and Gate Stealer or simply Deception). Receiving simulations may “promote” techniques to a less specific level supported by the local radar receiver model.
b) The noise kind shall indicate that the jammer is attempting to affect the target beam by raising the level of background noise against which target returns are detected.
c) The deception kind shall indicate that the jamming technique is designed to offer false information to deny specific information on either bearing, range, velocity, or a combination of these.
d) The special kind shall be reserved for nonrealistic modes and may be used for testing or special training needs.

The following general requirements shall apply:

a) When jamming is indicated:
    1) The Jamming Technique record shall indicate the methods used to conduct electronic warfare.
    2) The Track/Jam Data records shall indicate a list of targeted emitter beams corresponding to the value in the Number of Targets field.
    3) No Jamming Technique record shall be assigned a value equal to NO_KIND:NO_CATEGORY:NO_SUBCAT:NO_SPECIFIC.
b) For each known targeted emitter beam:
    1) The Entity ID in the Track/Jam Data record shall indicate the targeted entity.
    2) The Emitter Number in the Track/Jam Data record shall indicate the targeted emitter.
    3) The Beam Number in the Track/Jam Data record shall indicate the targeted beam.
c) For each known targeted emitter, if the targeted beams are not known or if all beams on the emitter are targeted:
    1) The Entity ID in the Track/Jam Data record shall indicate the targeted entity.
    2) The Emitter Number in the Track/Jam Data record shall indicate the targeted emitter.
    3) The Beam Number in the Track/Jam Data record shall indicate ALL_BEAMS.
d) For each known targeted entity, if the targeted emitters are not known or if all emitters on the entity are targeted:
    1) The Entity ID in the Track/Jam Data record shall indicate the targeted entity.
    2) The Emitter Number in the Track/Jam Data record shall indicate ALL_EMITTERS.
    3) The Beam Number in the Track/Jam Data record shall indicate ALL_BEAMS.
e) If the targeted entities are not known (e.g., preemptive noise jamming):
    1) The Number of Targets field shall be set to zero.

2) The High-Density Track/Jam field shall be set to Selected (1).
3) The Track/Jam Data records shall be omitted.

5.7.3.9 Phased array radars

Phased array emitters shall be modeled in a manner consistent with conventional emitters. Time-sharing between modes on a single emitter system may be indicated by use of specific beam function values. For example, a single emitter may be modeled using separate beams with time-shared search, time-shared acquisition, time-shared track, time-shared command guidance, and time-shared illumination beam functions (see 6.2.11).

5.7.3.10 Supplementing the EE PDU

The following attribute records may be implemented and issued in Attribute PDUs to supplement active emitter beams contained in EE PDUs:

a) Blanking Sector attribute record (see 6.2.21.1).
b) Angle Deception attribute record (see 6.2.21.2).
c) False Targets attribute record (see 6.2.21.3).

The issuance and receipt rules associated with each attribute record are provided in the specification paragraph for that record.

5.7.4 Designator PDU

5.7.4.1 Purpose

The Designator PDU shall be used to communicate information for designation functions such as a lasing function to support a laser-guided weapon engagement.

5.7.4.2 Information contained in a Designator PDU

The Designator PDU shall contain the following information:

a) Identification of the entity performing the designation.
b) Code name for the designator system.
c) Identification of the entity being designated (provided only if the designator spot is on an entity).
d) Code of the designator (e.g., laser code).
e) Output power of the designator.
f) The designator wavelength.
g) Designator spot position with respect to an entity. (This data provides specific detail of the spot position with respect to a designated entity's coordinate system.)
h) Location of the designator spot in the world coordinate system.
i) Dead reckoning parameters that should be employed when extrapolating the world coordinate position of the designator spot. (Values in this field shall include the dead reckoning algorithm in use and linear acceleration.)

5.7.4.3 Issuance of the Designator PDU

The simulation shall issue a Designator PDU when a designator is active and any of the following occur:

a) Discrepancy between the designator spot's actual world coordinate position (as determined by its own internal model) and its dead reckoned position (as determined by using specified dead reckoning algorithms) exceeds a predetermined threshold (see Annex E concerning dead reckoning and 5.3.2.3.2 concerning threshold values).

b) The designator goes inactive. A final Designator PDU shall be issued with the Power field set to zero to indicate that the designator is inactive.

c) A predetermined length of real-world time has elapsed since the issuing of the last Designator PDU. The Designator PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_DESIGNATOR. (See 6.1.8 for parameter details and default values.)

d) Entity to which the designator spot is attached ceases to exist in the synthetic environment. A final Designator PDU shall be issued with the Designator Power field set to zero to indicate that the designator is deactivated.

The following field specific requirements apply:

— *Designated Entity ID*. If the designator spot is not on an entity, the Designated Entity ID field shall contain D-SPOT_NO_ENTITY.

— *Designator Spot with Respect to Designated Entity*. If the Designated Entity ID field contains the value D-SPOT_NO_ENTITY, the Designator Spot with Respect to Designated Entity field shall be set to 0.0, 0.0, 0.0 (all bits set to zero).

5.7.4.4 Receipt of the Designator PDU

Upon receipt of the Designator PDU, a simulation application controlling a designation-guided munition shall use the information contained in the PDU to simulate the guidance and the final detonation of the weapon after it is fired. The simulation application controlling the weapon shall check to see whether there is information in the Designated Entity field of the Designator PDU for identification of a designated entity. If there is no entity identified as being designated, then the simulation application controlling the weapon shall use the designator spot world coordinate location information to model the weapon guidance. If there is an entity identified as being designated, then the simulation application controlling the weapon shall use the designator spot position with respect to the entity and information from that entity's Entity State PDU to provide accurate input to the guidance model. Because the guidance simulation may consider factors such as atmospheric effects, guidance system errors, or countermeasures, the location of the actual detonation could be different from the location of the designator spot in the Designator PDU.

A receiving simulation application may also use the information in the Designator PDU to stimulate laser detection systems or anti-laser active protection systems.

5.7.5 Underwater Acoustic (UA) PDU

5.7.5.1 Purpose

The UA PDU shall be used to communicate UA active emissions (intentional emissions) and passive signature (unintentional emissions) information.

5.7.5.2 Information contained in the UA PDU

The UA PDU shall contain the following information:

a) Identification of the emitting entity.

b) Identification of the event.

c) Identification of the type of update information in the UA PDU:

1) A state update shall provide a full description of the emission system(s) and passive signature information identified in the UA PDU.

2) A change data update shall provide a method of allowing changes of UA emitter state or passive signature to be communicated between state updates.

d) Information required to define the entity's passive signature, including:

1) Passive parameter index.

2) Propulsion plant configuration.

3) Number of shafts.

4) Current and ordered shaft speed.

5) Shaft speed rate of change.

6) Number of Additional Passive Activities (APAs).

7) APA parameter index.

8) APA value.

e) Information required to identify the active emission platform characteristics, including:

1) Number of UA emitter systems for which information is being provided in the UA PDU.

2) Information for one or more UA emitter systems, including:

i) Length of the emitter system data for each system.

ii) Number of main beams for each system.

iii) Acoustic emitter system information, which includes the acoustic emitter system name, function of the system, and acoustic identification number.

iv) Location of the UA emitter with respect to the entity.

v) Information for one or more acoustic beams that the system has, including:

— Length of the beam data.

— Beam identification number for each beam.

— Fundamental parametric data used to define the entity's active emissions. This field defines the active emission parameter index, beam scan pattern, orientation, and beamwidth, which can vary dynamically during system operation.

5.7.5.3 Issuance of the UA PDU

Entities that simulate UA noise sources will output acoustic noise parameters that, in conjunction with database information, describe intentional and unintentional UA emissions. The UA PDU shall be issued in the following instances:

a) *State Update*

1) An Entity State PDU for the associated entity is first issued, and the entity is generating passive signature information or has at least one acoustic emitter that is currently active.

2) A predetermined length of real-world time has elapsed since issuing the last UA PDU. The UA State Update heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_UA. (See 6.1.8 for parameter details and default values.)

b) *Changed Data Update*. Emitter operational parameters or entity configuration data contained in the UA PDU fields change. The UA PDU for a changed data update shall contain only data for those parameters that exceed the defined thresholds. The parameters that can initiate a changed data update are:

1) Propulsion plant configuration change, i.e., any change in the entity propulsion plant configuration.

2) A change in Shaft revolutions per minute (RPMs). The Shaft RPM change threshold shall be identified by the symbolic name UA_SRPM_THRSH. (See 6.1.8 for parameter details and default values.)

3) A change in Shaft RPM rate of change (ROC). The Shaft RPM ROC change threshold shall be identified by the symbolic name UA_SRPM_ROC_THRSH. (See 6.1.8 for parameter details and default values.)

4) APA additions, deletions, or parametric changes to current activity.

5) A change in the number of active acoustic emitter systems.

6) A change in the number of active main beams per system.

7) A change in location with respect to the entity (e.g., a towed active emitter). The location with respect to the entity change threshold shall be identified by the symbolic name UA_POS_THRSH. (See 6.1.8 for parameter details and default values.)

8) Fundamental Data Parameters change, i.e., a change in the Active Emission Parameter Index values (e.g., pulse type, pulse duration, pulse interval, source power, pulse frequency content, and pulse beam pattern), Scan Pattern (if applicable), or main beam steering. The azimuth or depression/elevation change threshold shall be identified by the symbolic name UA_ORIENT_THRSH. (See 6.1.8 for parameter details and default values.)

The following field specific requirements apply:

— *APA Status*. This field of the APA Parameter Index record (see Table 163) shall be used to indicate an on/off/change status of the record. It shall be set to one of the following:

  1) Deselected/Off (0). Indicates that the record has been deselected or turned off.

  2) APA Value Change Only (1). Indicates that the associated APA Value field has changed without a corresponding state change.

  3) State Change (2). Indicates that the state has changed.

  4) Record Activation (3). Indicates that the record shall be activated commensurate with the conditions in the database and the APA Value field.

The UA PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.7.5.4 Receipt of the UA PDU

Upon receipt of an UA PDU (and corresponding Entity State PDU), the receiving UA simulations shall determine whether the emission or signature is detectable and use the information in the UA PDU and predefined databases to appropriately influence its UA detection equipment or simulation of that equipment in those simulations. The receiving simulation shall be responsible for simulating the environment and transmission effects (including reflections) from the noise source to the sensor(s). Environmental conditions may be provided by the Environmental Process PDU (see 5.11.2.2) and the Gridded Data PDU (see 5.11.2.3).

5.7.5.5 UA emission regeneration

Active and passive entity emissions and signatures shall be generated based on parameters stored in databases of the receiving UA simulations as enumerated by the UA PDU and based on the dynamic operational parameters included in the UA PDU.

5.7.6 Identification Friend or Foe (IFF) PDU

5.7.6.1 Purpose

The IFF PDU shall be used to communicate information for functions such as cooperative or noncooperative military IFF transponder and interrogator systems (e.g., Mark XII and Mark XIIA) and cooperative civilian Air Traffic Control (ATC) radar beacon systems such as Mode A, Mode C, and Mode Select (Mode S). It also includes military Battlefield Target Identification Device (BTID) systems that are used on military vehicles. The term "IFF" is used in this subclause for brevity and shall include any military or civilian transponder or interrogator system. The term "layer" is synonymous with the term "information layer," and the term "system" is used to mean real-world equipment whose emissions are included in an IFF PDU.

NOTE—Although Navigational Aids (NAVAIDS) were originally planned to be implemented using the IFF PDU, there has been no known use of the IFF PDU for this purpose. There are a wide variety of NAVAIDS available for aircraft navigation. The Transmitter PDU has been used by some simulations to identify NAVAIDS in support of virtual, high-fidelity flight simulators. The requirements for the use of NAVAIDS by other than virtual flight simulators and the impact of using the Transmitter PDU for NAVAIDS distribution in an exercise has not been assessed.

The System Type, System Name, and System Designator fields when combined uniquely identify a transponder or interrogator system. Unless otherwise specified, references to the Entity State PDU shall be taken to also include the Entity State Update PDU and Time Space Position Information (TSPI) PDU. The following general information and requirements shall apply:

a) *General Requirements*
   1) The IFF PDU shall be used to represent both functional data and electromagnetic emissions generated by transponder/interrogator systems that support surveillance, tracking, and identification of aircraft, ships, and vehicles.
   2) The EE PDU shall not be used to represent emissions for any types of systems listed in Table 14.
   3) Functional data shall always be required to be conveyed in the IFF PDU to represent the specific system being described in the PDU. Emissions data may be included as needed to support interactions at the functional level or for meeting requirements of higher fidelity systems.
   4) The IFF PDU supports active emissions for transponder systems that either reply to an interrogation or use squitter to transmit information without receiving an interrogation. Support is provided for uplink or downlink data report formats associated with both civilian and military transponders and interrogators [e.g., Mode 5, Mode S, and Automatic Dependent Surveillance—Broadcast (ADS-B)]. The IFF Data record is used to define specific data report formats.
   5) Every transponder, interrogator, or other system represented by the issuance of an IFF PDU shall be associated with an active entity.
   6) Every transponder, interrogator, and other system device for which the IFF PDU is appropriate shall be required to issue an IFF PDU to denote the presence of such devices in the synthetic environment whenever they are turned on and emitting.
   7) A transponder or interrogator model shall be able to operate using the Regeneration Mode. It may, in addition, implement the Interactive Mode to cause specific interrogations or replies to interrogations to be generated. Both the Regeneration and Interactive Mode methods may be used simultaneously by a simulation to interoperate with different entities and simulations:
      i) Regeneration Mode. In this mode, a transponder IFF PDU is not issued in response to a specific interrogator IFF PDU. Also, a transponder does not expect a specific interrogator IFF PDU stimulus to activate visual or other cues indicating the presence of an interrogation. Instead, functional and emissions data reflecting the truth state of the system are initially issued and then issued upon a change in data and at a heartbeat rate. This provides the truth data that a receiver system needs to model receipt of the transmission.

ii) Interactive Mode. In this mode, an interrogation is sent with the expectation that a transponder will reply to the presence of that interrogation by issuing a reply transponder IFF PDU. A transponder may use a received interrogation IFF PDU as a specific instance to which it will reply. The interrogator IFF PDU may also be used to stimulate on-board systems that can detect specific interrogations. See 5.7.6.5 for general requirements for the Interactive Mode.

8) If a receiving simulation has the capability to set transponder data for received entities for distribution to other applications not directly connected on the DIS network, receipt of transponder data on the received entity shall always override locally entered data for the received entity.

9) A receiving simulation may perform validity checks on parameter data to check that they are within the limits specified in Annex B. Only invalid data for the specific parameter processed shall be ignored. Any previously accepted data for that parameter may be either retained or cleared.

10) The implementation of the Damage and Malfunction Status fields contained in an IFF PDU is beyond the scope of this standard.

b) *Information Layer Implementations*

1) Information layers (layers) define functional, emission, and transponder/interrogator link message data associated with a specific system type. A maximum of seven layers shall be able to be defined and used. The Information Layers field of the Fundamental Operational data record (6.2.39) defines the layers that are present in an IFF PDU.

2) Only layers associated with a single, specific system shall be included in a single IFF PDU excluding the inclusion of basic Traffic Collision Avoidance System (TCAS)/Airborne Collision Avoidance System (ACAS) system information that may be included in Layer 1 [see item c5) of 5.7.6.1].

3) Multiple layers may be included in a single IFF PDU up to the maximum size of a PDU. When multiple layers are included, the layers shall be positioned within the IFF PDU in ascending numerical order (i.e., Layer 1 followed by applicable layers 2, 3, 4, and 5).

4) Layer 1 shall always be present and shall be the first layer in the IFF PDU. As such, it does not require a Layer Header record. All other layers that are present begin with a Layer Header record (see 6.2.51). The content of the IFF PDU fields and associated records contained in a layer will vary depending on the system type.

5) Table 13 provides a general description of the use(s) of each layer to support the various system types. A more detailed description can be found in 7.6.5.1.

Table 13—Information layers

Layer	Description
1	Basic System Data
2	Basic Emissions Data
3	Mode 5 Functional Data
4	Mode S Functional Data
5	Data Communications
6	Not defined
7	Not defined

c) *IFF Systems*. The types of systems that are required to use the IFF PDU to indicate their electronic presence in the synthetic environment are listed in Table 14. These systems shall exist as either sensor/emitter entities or be associated with another kind of entity. In the latter case, the system may be attached to or be carried internally by the entity (e.g., an aircraft platform). Other systems may also qualify to use the IFF PDU based on meeting all of the following criteria:

1) The system is an interrogator, transponder, or a TCAS/ACAS system associated with an entity.

2) The system emits analog or digital, decodable information that is used for battlefield target identification device (BTID) systems.

3) Requirements unique to a system type or system name are contained in Annex B.

4) A TCAS or an ACAS system shall be considered a subsystem of the transponder system on board an entity that has a TCAS or ACAS capability when the System Type contained in the System field of the IFF PDU is not TCAS or ACAS. Information on a TCAS or an ACAS system shall be included in the appropriate fields of Layer 1 using the Basic TCAS status record or TCAS/ACAS Status record for the associated transponder system type. If a System Type of TCAS or ACAS is the system type specified in the System field of an IFF PDU, then a separate IFF PDU shall be issued with only TCAS or ACAS interrogator or other transmitted message data excluding reply data from transponders.

Table 14—Transponder/interrogator systems

System	Examples (not inclusive)
IFF	Interrogators/Transponders — Aircraft, ship, submarine, and land vehicle entities. — Military Modes 1, 2, 3/A, 4, and 5.
ATCRB	Interrogators/Transponders — Aircraft, ship, submarine, and land vehicle entities. — Civilian ATC Radar Beacon Systems for Mode 3, C, TCAS, and Mode S.
BTID	Battlefield Target Identification Devices (BTIDs)

d) *Multiple Systems for a Local Entity*. A local entity may have more than one transponder or interrogator system associated with it. Also, more than one system with the same combination of System Type and System Name may be present for a transponder or interrogator. The following rules shall apply to multiple systems:

1) A separate IFF PDU shall be issued for each transponder and interrogator system except that basic TCAS/ACAS system information shall be treated as a special subsystem of Mark XII, Mark XIIA, ATCRBS, Mode 5, and Mode S transponders to reduce the proliferation of IFF PDUs. When included, the TCAS/ACAS record shall be included as Parameter 6 of the Fundamental Operational Parameter record.

2) The System Designator field shall be used to differentiate between two or more identical transponder, interrogator, or combined interrogator/transponder IFF systems. An identical IFF system is defined as two or more IFF systems carried by an entity that have the same system type/system name combination. As a separate IFF PDU is required to be sent for each IFF system carried by an entity, even if the IFF systems have the same system type/system name combination, a receiving simulation has to have a way to distinguish which IFF PDU goes with which identical IFF system. Once a System Designator is assigned to a specific IFF system, it

shall remain assigned to that IFF system and reflected in its IFF PDU as long as the entity is active in the exercise:

i) The IFF PDU for the first (or only) specific IFF system associated with an entity shall have the System Designator field set to zero (0). Each IFF PDU issued for any additional, identical IFF system(s) associated with the same entity shall be assigned System Designators in sequential order (i.e., 1, 2, and 3).

ii) When a simulation issues IFF PDUs for multiple, identical IFF systems, there is no requirement that the IFF system with the lowest System Designator number be sent first.

iii) A combined interrogator/transponder IFF system shall be denoted by the words “Combined Interrogator/Transponder (CIT)” as part of the System Type name. As separate IFF PDUs are required to be sent for the interrogator and the transponder for a CIT IFF system, a single CIT System would be sending two PDUs with System Designators of 0 and 1, respectively. If there is a second identical CIT IFF system for the same entity, the interrogator and transponder IFF PDUs associated with the second identical CIT IFF system would be assigned System Designators 2 and 3, and so forth.

NOTE—As an example, if there are two identical transponder systems for an entity, the first system will have a System Designator = 0 and the second system will have a System Designator = 1. If the same entity also has two identical interrogator IFF systems, the two interrogator IFF systems will also be given System Designators of 0 and 1. There will be four separate IFF PDUs issued for the entity, one for each physical IFF system that is aboard the entity. Another example is where an entity has two identical Combined Interrogator/Transponder (CIT) IFF Systems. As separate IFF PDUs are required to be sent for the interrogator and transponder associated with a CIT IFF System, a total of four IFF PDUs would be issued for the entity and assigned System Designators 0, 1, 2, and 3.

e) *Multiple Systems for a Remote Entity*. A remote entity may have more than one transponder or interrogator system associated with it. Also, more than one system with the same combination of System Type and System Name may be present for a transponder or interrogator. A receiving simulation may only have the capability to store a single system, or one interrogator and one transponder for a remote entity. In that case, the first system processed of each type system (a transponder or an interrogator) shall remain as the system stored for the remote entity until IFF PDUs are no longer being received for that system.

f) *Nonentity Owner Issuance of IFF PDUs*. An IFF PDU may be issued for a system associated with an entity by a simulation that is not the owner of the entity. If this occurs, the following requirements shall apply:

1) Pre-coordination and the exercise agreement are necessary to achieve interoperability.

2) The nonentity owner simulation that issues IFF PDUs for another simulation’s entities shall be capable of detecting when another simulation is also issuing IFF PDUs for the same system associated with an entity. In this event, the simulation shall cease issuing the IFF PDU for that system.

3) All requirements specified elsewhere in 5.7.6 regarding IFF PDU information, issuance, and receipt rules shall apply to a nonentity owner simulation.

g) *Combined Transponder/Interrogator Systems*. When an IFF system associated with an entity is a combined transponder and interrogator system as indicated by the system type, the following general requirements shall apply:

1) A separate IFF PDU shall be issued, one for the transponder and one for the interrogator.

2) The Transponder/Interrogator field of the Change/Options record shall be set to indicate whether this is a transponder or interrogator IFF PDU.

3) The issuance and receipt rules applicable for a transponder or an interrogator contained herein and in Annex B shall apply to each of the IFF PDUs that represents a combined transponder/interrogator system.

h) *Interrogator System Set to Off*. An interrogator whose System On/Off Status field is set to Off (0). If an interrogator does not process received IFF PDUs when its System On/Off Status field is set to Off (0), then it shall not use any previously received transponder data.

5.7.6.2 Information contained in an IFF PDU

The IFF PDU shall contain the following information:

a) Entity ID of the entity emitting IFF or ATC interrogator or transponder signals.
b) Event ID of a specific event (optional use).
c) Type of system emitting the signals.
d) Location of the emitting system antenna with respect to the emitting entity.
e) Layer 1 information reflecting the basic status and functional data associated with a specific system. This includes which modes of signals the system is capable of emitting and the settings for those modes (e.g., codes or other data transmitted). The intent is to capture the state of such systems as may be set automatically by the system or manually by operator action.
f) Layer 2 emissions data (optional). When included, such emissions data shall include, as a minimum, the following:
    1) Beam data.
    2) IFF Fundamental Parameter Data record (see 6.2.44).
    3) Secondary Operational data record (see 6.2.76).
g) Layer 3 when a system has a Mode 5 capability.
h) Layer 4 when a system has a Mode S capability.
i) Layer 5 when transponder/interrogator link messages emulating actual message formats need to be sent or when Layer 1 and 2 needs to be extended.
j) Each Layer, other than Layer 1, that is included in an IFF PDU shall include a Layer Header record.

5.7.6.3 Issuance of the IFF PDU

The general requirements for issuing the IFF PDU independent of a specific system type are contained herein. Additional issuance requirements applicable to a specific system type are defined in Annex B. An IFF PDU shall be issued as follows:

a) This is the initial IFF PDU for an entity because the System has been turned on.
b) Only the layer(s) required to convey initial, changed, or heartbeat data for a system shall be included in the IFF PDU except that Layer 1 shall always be included in an IFF PDU regardless of whether any data in Layer 1 has changed, excluding changes to PDU Header fields.
c) A predetermined length of reference time has elapsed since issuing the last *heartbeat* IFF PDU for an IFF system that is associated with an entity. An IFF PDU that contains all IFF data associated with a specific IFF system on board an entity is referred to as a *heartbeat* IFF PDU. If there is more than one specific IFF system associated with an entity, then a separate IFF PDU is issued for that IFF system including the initial IFF PDU, subsequent IFF PDUs issued because of a change in data, and the required, periodic *heartbeat* IFF PDU. The *heartbeat* IFF PDU periodic transmission parameter value shall be identified by the symbolic name HBT_PDU_IFF. (See 6.1.8 for parameter details and default values.)
d) A change to either of the Layer 2 beam center descriptors exceeds the specified change thresholds since issuance of the last IFF PDU. If no value is established, the default threshold azimuth shall be IFF_AZ_THRSH and the default threshold elevation shall be IFF_EL_THRSH.
e) A change to any other data field that has a change threshold where the change has exceeded the threshold.
f) When an IFF PDU is issued because of a change, it shall be issued as follows:

1) Within a predetermined length of real-world time. This Change Latency Parameter value may be established at the start of the exercise or during the exercise by means of Simulation Management PDUs. If no value is established, a default value of IFF_CHG_LATENCY shall be used. (See 6.1.8 for parameter details and default values.)

2) The Change Indicator (bit 0) of the Change/Options field in the System Identifier record shall be set to Initial Report/Change (1). The next IFF PDU issued under item c) above (heartbeat) shall also have the Change Indicator set to indicate that a change has occurred. Thereafter, the IFF PDU when issued shall have the Change Indicator set to Not Initial Report/No Change (0) until another change occurs.

g) Once an IFF PDU is issued for an entity, it shall continue to be periodically issued even if the System is turned off until one of the conditions identified in item h) below exists.

h) IFF PDUs for specific system(s) being carried by an entity shall cease to be transmitted under the following conditions:

1) The associated entity for which an IFF PDU is being issued is deactivated and is no longer active in the exercise.

2) The entity is active but being filtered such that Entity State, Entity State Update, or TSPI PDUs are not being placed on the distributed simulation network:

i) The application that is performing the filtering function shall not pass IFF PDUs through the filter while the associated entity is being filtered.

ii) If the associated entity ceases to be filtered, an IFF PDU shall be issued within the simulation time interval specified in the IFF PDU Resume variable parameter following the initial issuance of the unfiltered Entity State, Entity State Update, or TSPI PDU. The IFF PDU Resume variable parameter symbolic name is IFF_PDU_RESUME. (See 6.1.8 for parameter details and default values.)

i) If the Entity State or Entity State Update PDU Appearance: Damage Status subfield value indicates damage, the following requirements shall be met:

1) If the model has a capability to determine whether damage to the entity would affect one or more of its transponder or interrogator systems such as to cause the cessation of IFF emissions, and such a determination is made, then that system shall cease to be output in the IFF PDU in accordance with item k) in 5.7.6.3.

2) If the simulation does not model separate damage to an interrogator or transponder associated with an entity and the Entity Appearance record Damage field (bits 3 to 4) is set to Destroyed (3), it shall be assumed that any interrogator or transponder system(s) was also destroyed and IFF PDUs for those systems shall automatically cease to be issued in accordance with item k) in 5.7.6.3.

j) If the model issuing the IFF PDU determines that there has been damage to a specific component of a system (e.g., Layer 1, Parameter 2), any associated IFF PDU Damage Status field shall be set to Damaged (1) and any associated component On/Off Status shall be set to Off (0).

k) If an IFF PDU will no longer be issued due to damage, the following requirements shall be met:

1) An IFF PDU shall be automatically issued within the reference time interval specified in the Final IFF PDU variable parameter when a simulation determines that the entity can no longer transmit information that is contained in the IFF PDU for a specific system. The Final IFF PDU variable parameter symbolic name is IFF_PDU_FINAL. (See 6.1.8 for parameter details and default values.)

2) The Layer 1 System Status: The System On/Off subfield shall be set to Off (0), and all Parameter 1 to 6 Status fields (Bit 13) for Capable modes shall be set to Not Capable (1).

3) The Layer 1 Operational Status field may be set to System Failed (1) and the Parameter 1 to 6 Damage Status field may be set to Damaged (1), or the Malfunction Status field set to Malfunction (1) to indicate Damage and Malfunction, respectively.

4) Appropriate functional or emission data fields in other layers shall be set to indicate that no emissions are being output.

5) If the simulation desires to emulate the activation of an electronic locator system (e.g., Emergency Locator Transmitter), an IFF PDU shall be initially issued when it is activated.

l) *Specific Field Requirements*

1) The System On/Off Status field of the System Status record contained in the Fundamental Operational Data record of Layer 1 emulates the Master Control switch for a system. The System On/Off Status setting shall be set independent of, and not affect, the present On/Off Status of any IFF component except for the System Mode field of the System ID record. The System Mode field provides additional settings (states) for the Master Switch (e.g., Standby and Low Sensitivity). The setting of the System Mode field and the System On/Off Status field are coupled such that they reflect a comparable setting. The System Mode field is only required to be implemented by specific system types. When this field is implemented, the relationship between it and the System On/Off Status field shall be as follows:

   i) If the System On/Off Status field is set to Off (0), the System Mode field shall also be set to Off (1) and vice versa.

   ii) If the System On/Off Status field is set to On (1), the System Mode field shall be set to set to Normal/On (3).

   iii) Whenever the System Mode field is set to a different state, its effect on the System On/Off Status field shall be as specified in item b) of B.2.1.3.1.

2) If the entity associated with an IFF PDU is deactivated, no further IFF PDUs shall be issued for that entity except that a simulation may issue a Final IFF PDU for the entity as follows:

   i) The PDU Header, Emitting Entity ID, Antenna Location, and System ID fields shall contain valid data.

   ii) The Layer 1 System Status On/Off field shall be set to Off (0).

   iii) No other layers are required to be output. However, if output, the fields shall be either set to the last values available or set to zero.

   iv) Any Final IFF PDU shall be automatically issued within the reference time interval specified in the Final IFF PDU variable parameter following the issuance of the *final* Entity State PDU. The Final IFF PDU variable parameter symbolic name is IFF_PDU_FINAL. (See 6.1.8 for parameter details and default values.)

3) Use of the Event ID field is optional. If it is not used, it shall be set to NO_EVENT_ID (see 6.2.33).

4) Antennas. An actual IFF system has one or more antennas associated with it. The IFF PDU supports identifying the relative location of each IFF system antenna with respect to the entity. None, some, or all of the antenna locations may be included in an IFF PDU for a specific IFF system as follows:

   i) If no IFF system antenna location is modeled, the Layer 1 Relative Antenna Location field shall be set to 0.0, 0.0, 0.0 (all bits set to zero).

   ii) If there is only one antenna location modeled, the Layer 1 Relative Antenna Location field shall contain the location of that antenna.

   iii) If multiple antenna locations are modeled, one of the antenna locations shall be designated as the primary antenna location by the simulation and its location included in the Layer 1 Relative Antenna Location. Each additional antenna shall be considered an auxiliary antenna and contained in separate Antenna Location IFF Data records in the appropriate layer (see B.2.3).

5) The Change Indicator of the Change/Options field shall be used as follows:

i) The Change Indicator shall be set to Initial Report/Change (1) for all initial reports and whenever a change is being reported even if the change coincides with a *heartbeat* IFF PDU.

ii) When the Change Indicator is set to Initial Report/Change (1) to indicate a change in data, the next *heartbeat* IFF PDU shall also have the Change Indicator set to Initial Report/Change (1). Thereafter, the Change Indicator shall be reset to Not Initial Report/No Change (0) until another change occurs.

6) Beam Data. The Beam Data field within Layer 2 shall be set as follows: If the main beam antenna activity of the IFF emitting system before the next IFF PDU update can be represented by a single beam, and then a description of the beam shall be entered in this field. If multiple beams would be required, then this field shall be filled with zeroes.

m) *IFF Fundamental Parameter Data Record*

1) At least one IFF Fundamental Parameter Data record shall be included whenever Layer 2 is present.

2) If all modes have the same emissions characteristics, then the Applicable Modes field shall be set to All Modes (1) and only one record need be included. Otherwise, a simulation shall include a separate record for each of the modes or combination of modes that share similar emissions characteristics as deemed appropriate.

5.7.6.4 Receipt of the IFF PDU

The general requirements for the receipt of the IFF PDU independent of a specific system type are contained herein. Additional receipt requirements applicable to a specific system are defined in Annex B. An IFF PDU shall be processed upon receipt as follows:

a) A receiving simulation that processes the IFF PDU shall use the information contained therein to model applicable system receipt of the data. This includes determining whether, in fact, it would be able to receive the data and whether it was valid data:

1) The formats of layers 1, 2, and 5 do not change based on the system type. Layers 3, 4, 6, and 7 of an IFF PDU may have different formats depending on the system type. A given system type shall only have one format defined for a layer. All layer formats for all system types are defined in 7.6.5.

2) The System Type field of the System ID record (6.2.87) contained in Layer 1 shall always be processed to determine the layer format(s) of layers applicable to that system type. See Table B.61 for a list of the required and optional layers for each system type.

b) A receiving simulation shall, as a minimum, be able to process all the applicable data from the functional data layer(s) for any received systems that are modeled. Processing of emissions data shall be as defined by the simulation agreement and the capabilities of the receiving simulation's model(s).

c) The System On/Off Status field of the System Status record contained in the Fundamental Operational Data record of Layer 1 shall be processed as follows:

1) When the System On/Off Status field is set to On (1), any individual IFF component represented by a record or field (e.g., Mode 1 Code record) having an On/Off status shall be processed in accordance with that status.

2) When the System On/Off Status field is set to Off (0), any individual IFF component for the associated remote entity having an On/Off status field, or a field whose setting is indicative that the component is emitting, shall be processed as if that IFF component was turned off regardless of its present On/Off status setting. This includes the System Mode field and any special replies, such as an Identification of Position (I/P) reply that may be temporarily active.

d) If the entity associated with an IFF PDU is filtered upon receipt, the associated IFF PDUs for that entity shall also be filtered.

e) *Storing Remote Entity IFF Systems*

1) If the receiving simulation only has the capability to store a single system (interrogator or transponder) or to store one interrogator and one transponder for a remote entity, whatever system it initially stores, or stores for each type, shall be the only one it shall maintain for that entity. Other transponders associated with the entity shall be discarded.

2) If the receiving simulation has the capability to store multiple interrogator or transponder systems for a remote entity, the System Type, System Name, and System Designator values shall be stored for each received system. Subsequent interrogator or transponder IFF PDUs received for the same entity shall be processed based on the System Type, System Name, and System Designator values to correctly update an existing remote IFF system or to add a new one.

f) A receiving simulation shall only process system types for which they have a specific model for that system type. If they do not process a specific system type, then the IFF PDU shall be discarded. If a System Type is a combined transponder/interrogator, the receiving simulation need only model the receipt of the transponder for that system.

g) For each IFF system, the receiving simulation shall maintain a countdown timer that is restarted when an update is received for that system:

1) If the countdown timer for a particular IFF system expires, then that IFF system shall be considered inactive. When an IFF system becomes inactive, an interrogation model shall not use any previously received transponder data.

2) The duration of the timeout of an IFF PDU for a specific IFF System associated with an entity shall be established at exercise start, although it may be changed during the exercise. The timeout parameter shall be established as the value of HBT_PDU_IFF multiplied by HBT_TIMEOUT_MPLIER (see 6.1.8 for details of these two parameters and their default values).

5.7.6.5 Interactive Mode

5.7.6.5.1 General

The following requirements apply to a simulation that is capable of issuing or receiving an *interactive* IFF PDU. The purpose of the Interactive Mode is to allow an interrogator simulation to directly interact with a specific entity's transponder to more closely emulate the real-world interrogation and reply sequence. This could also be used for two interrogators to communicate directly with each other when the IFF system has that capability such as is the case of the Mode S system. The Interactive Mode is primarily used for test and evaluation of individual IFF systems and for individual pilot training in a high-fidelity, virtual flight simulator where cockpit displays need to more realistically portray simulated interrogations. However, an exercise agreement may allow the presence of multiple interrogators and transponders operating in the Interactive Mode together with other IFF systems operating in the Regeneration Mode.

The term "Interactive Mode" indicates that an interrogator or transponder is currently operating in that mode. An interrogator or transponder is operating in the Interactive Mode when the Interactive Capable field is set to Capable (1).

The term "Regeneration Mode" indicates that an interrogator or transponder is currently operating in that mode. An interrogator or transponder is operating in the Regeneration Mode when the Interactive Capable field is set to Not Capable (0).

The term "*interactive* IFF PDU" is used to indicate that the PDU represents an interactive issuance of the IFF PDU as indicated by the Simulation Mode field of the Change/Options record being set to Interactive (1) [and the Interactive Capable field being set to Capable (1), as a Not Capable (0) IFF system would not be able to issue *interactive* IFF PDUs].

The term "*regeneration* IFF PDU" is used to indicate that the PDU represents a regeneration issuance of the IFF PDU as indicated by the Simulation Mode field of the Change/Options record being set to Regeneration (0). The Interactive Capable field will be set to Capable (1) if the IFF system is currently operating in the Interactive Mode or to Not Capable (0) if the IFF system is not currently operating in the Interactive Mode.

Although there is no DIS requirement for an IFF system simulation to be able to operate in the Interactive Mode, if it does have the capability and is operating in that mode, it shall adhere to the requirements herein.

5.7.6.5.2 General requirements

The following general requirements apply to both interrogators and transponders operating in the Interactive Mode:

a) Any IFF PDU that represents an interactive interrogation or reply shall be indicated by both the Change/Options record Simulation Mode field being set to Interactive (1) and the Interactive Capable field being set to Capable (1) (see B.2.4).

b) An IFF system that is currently in the Interactive Mode shall also issue *regeneration* IFF PDUs for that IFF system in accordance with the Regeneration issuance requirements contained in this standard except as noted in item d) below. These *regeneration* IFF PDUs shall have the Change/Options record, Simulation Mode field set to Regeneration (0), and the Interactive Capable field set to Capable (1) (see B.2.4).

c) An *interactive* interrogator or transponder IFF PDU for an entity, as a minimum, shall meet the following requirements (see Annex B for additional interactive requirements related to specific IFF system types, modes, and special IFF capabilities):

    1) Any layer required as listed below shall include the same data that is currently being included in the *regeneration* IFF PDU for this IFF system except for specific fields required to be set based on being an *interactive* IFF PDU. This may include data that is not related to the specific modes or other information being requested in the interrogation or contained in a reply.

    2) If an *interactive* interrogation or reply IFF PDU contains data that is different from that provided in the *regeneration* IFF PDU for the IFF system, excluding specific field settings required to for any *interactive* IFF PDU, the data shall be contained in appropriate IFF Data records in Layer 5 instead.

    NOTE—Regeneration data is based on what would be sent by an interrogator or a transponder in usual circumstances. Interactive Mode 4 and Mode 5 replies are examples of cases where the interactive reply data may be different from the regeneration data contained in the corresponding data fields of applicable layers in a *regeneration* IFF PDU due to special relationships between a specific transponder and a specific interrogator (see B.5.1.2.2 and B.5.1.2.4).

    3) IFF PDU Layer 1 is always required and shall always include all the data included in a *regeneration* IFF PDU.

    4) Layer 2 shall be included if it is being included in the *regeneration* IFF PDU for this IFF system.

    5) Layer 3 shall be included if this is a Mode 5 interrogation or reply.

    6) Layer 4 shall be included if this is a Mode S interrogation or reply.

    7) Layer 5 shall always be included and contain, as a minimum, the Basic Interactive IFF Data record.

d) If all interrogators and transponders of all entities in a DIS exercise will be operating in the Interactive Mode, the exercise agreement may allow such simulations to not issue *regeneration* IFF PDUs for those entities.

e) A simulation may change the status of an interrogator or transponder from Capable (1) to Not Capable (0) during a DIS exercise. If this occurs, this status shall be immediately communicated by the issuance of a *regeneration* IFF PDU for the IFF system with the Simulation Mode set to Regeneration (0) and the Interactive Capable field set to the appropriate value.

f) A simulation that can issue or receive *interactive* IFF PDUs may restrict which of its entities' IFF systems can operate in the Interactive Mode and which remote entities' IFF systems it may interact with in the Interactive Mode.

g) All interactive interrogators operating in a simulation environment where there is a mixture of Interactive and Regeneration Mode transponders that they would be able to interrogate shall process the *regeneration* IFF PDUs for those transponders if not interactive capable; i.e., Interactive Capable field = Not Capable (0).

h) The PDU Status record in the PDU Header shall have the Simulation Mode field set to Interactive (1). This allows *interactive* IFF PDUs to be filtered based on PDU Header information for simulations that do not process *interactive* IFF PDUs.

i) The Interactive Event ID field contained in the Basic Interactive IFF Data record shall be maintained and incremented separately from the Event ID field contained in Layer 1 of an IFF PDU. The Interactive Event ID is set by an interrogator that issues an interactive IFF PDU, and the identical number is included in the reply by an *interactive* transponder to an interrogation. The first number assigned is one, and then the Interactive Event ID is incremented by one for each succeeding interactive interrogation for the same IFF System. A transponder that has a capability to issue an *interactive* IFF PDU representing transmission of one or more squitter message(s) shall always set the Interactive Event ID to zero.

j) A transponder-equipped entity that is capable of generating *interactive* IFF PDUs representing the transmission of squitter messages shall be able to turn off the issuance of squitter *interactive* IFF PDUs both prior to and during an exercise. This may be done on a simulation-wide basis or by Entity ID and System Type.

5.7.6.5.3 Issuance rules

The following issuance rules shall apply to an *interactive* interrogator or transponder:

a) An *interactive* interrogator IFF PDU shall only be issued under the following conditions:

   1) The interrogator is in the Interactive Mode.

   2) The entity to be interrogated has been determined to own an Interactive Capable IFF system as indicated by the last received IFF PDU having the Interactive Capable field set to Capable (1).

   3) The IFF system to be interrogated has been determined to have the mode(s) or other element(s) to be interrogated, and they are indicated to be set to On (1) in appropriate On/Off Status fields of the last IFF PDU received.

   4) Once an interactive interrogation is made to an interactive IFF system, the interrogator shall continue to perform interactive interrogations of that IFF system based on the modeling of the specific interrogator system type until either:

      i) An IFF PDU is received for the IFF system with the Interactive Capable field set to Not Capable (0).

      ii) The interactive interrogator is no longer in the Interactive Mode. In this case, an IFF PDU shall be issued with the Simulation Mode field set to Regeneration (0) and the Interactive Capable field set to Not Capable (0).

b) An *interactive* transponder IFF PDU reply shall only be issued if the transponder is in the Interactive Mode and is responding to an interactive interrogation request that is directed to it. The following requirements shall be met:

   1) All layers applicable to the interrogated modes shall be included in the IFF PDU reply. These layers shall contain regeneration data. All reply data that differs from the corresponding regeneration data shall be included in the appropriate IFF Data record in Layer 5.

   2) The Basic Interactive IFF Data record shall be included in Layer 5. This shall always be the first IFF Data record in an *interactive* IFF PDU reply:

i) The Interrogating Entity ID, Interrogated Entity ID, and Interactive Event ID fields shall be set identical to the received values for these fields.

ii) The Transmission Indicator shall be set to Interrogation Reply (2).

iii) The Interrogated Modes field shall only be set to the requested interrogated modes that it is able to actually respond to based on present transponder settings and capabilities. These may differ from the requested interrogated modes.

iv) The Reply Amplification field shall be set to as follows:
- No Statement (0).
- Complete (1). All requested modes/information were responded to as an interactive reply.
- Limited (2). Some, but not all, requested modes/information were able to be responded to as an interactive reply.
- Unable to Respond (3). None of the requested modes/information were able to be responded to as an interactive reply.

3) If this is a reply to an interactive Mode 4 interrogation, see B.5.1.2.2 for additional requirements.

4) If this is a reply to an interactive Mode 5 interrogation, see B.5.1.2.4 for additional requirements.

5) If this is a reply to an interactive Mode S interrogation, the Interactive Basic Mode S IFF Data record shall also be included in Layer 5.

5.7.6.5.4 Receipt rules

The following receipt rules shall apply to an *interactive* interrogator or transponder:

a) If this is an interactive interrogator and it receives an *interactive* IFF PDU reply to one of its interactive interrogations, it shall be processed as follows:

   1) The received *interactive* IFF PDU reply data in Layer 5 that is stored shall be marked as interactive data to distinguish it from received regeneration data for the same transponder.

   2) The received Interrogated Modes field values shall be used to determine what interrogated modes were able to be responded to by the interactive transponder. Some of the reply data may be in Layer 5, and some in other layers. Whenever there is Layer 5 reply data for a field contained in other layers, the Layer 5 data shall be considered the interactive response data.

b) If this is an interactive transponder, and it receives an *interactive* IFF PDU interrogation directed to it, it shall respond as specified in item b) of 5.7.6.5.3 above.

c) Nothing in this standard shall preclude an interactive interrogator or transponder simulation from applying additional criteria (e.g., range, terrain, and interference) to determine whether it would have received an interactive interrogation or have been able to process an interactive reply.

5.7.7 Supplemental Emission/Entity State (SEES) PDU

5.7.7.1 Purpose

The SEES PDU shall be used to communicate supplemental information concerning passive emissions (infrared and radar cross section) and active emissions (acoustic), along with additional information pertaining to propulsion systems contained within the entity.

5.7.7.2 Information contained in the SEES PDU

The SEES PDU shall contain the following information:

a) Identification of the originating entity

b) Infrared signature representation index

c) Acoustic signature representation index

d) Radar cross-section signature representation index

e) Information required to define the entity's propulsion systems, including:

   1) Number of propulsion systems

   2) Number of vectoring nozzle systems

   3) Propulsion system data

   4) Vectoring nozzle system data

5.7.7.3 Issuance of the SEES PDU

Entities that simulate infrared and surface acoustic signatures shall output parameters that, in conjunction with database information, describe their own infrared and acoustic emissions. Also, entities shall output parameters that in conjunction with database information, describe their own radar cross-section information. The SEES PDU shall be issued in the following instances:

a) *State Update*

   1) An Entity State PDU for the associated entity is first issued, and the entity is generating supplemental emission/entity information.

   2) A predetermined length of real-world time has elapsed since issuing the last SEES PDU. This periodic transmission parameter and tolerance shall be identified by the symbolic name HBT_PDU_SEES. (See 6.1.8 for parameter details and default values.)

b) *Changed Data Update*. Propulsion or nozzle system parameters contained in the SEES PDU fields change more than a specified threshold, or the state of the entity's infrared, acoustic, or presented radar cross-section signature changes so that a new representation index shall be issued. A SEES PDU for a changed data update shall contain only data for the parameters that exceed the defined thresholds. The parameters that can initiate a changed data update are:

   1) Representation index change, i.e., any change in the value of the representation index (infrared, acoustic, or radar cross section).

   2) Power setting change. The power setting change parameter shall be identified by the symbolic name SEES_PS_THRSH. (See 6.1.8 for parameter details and default values.)

   3) Engine RPM change. The engine RPM change parameter shall be identified by the symbolic name SEES_RPM_THRSH. (See 6.1.8 for parameter details and default values.)

   4) Nozzle deflection angle (NDA) change. The NDA change parameter shall be identified by the symbolic name SEES_NDA_THRSH. (See 6.1.8 for parameter details and default values.)

The SEES PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.7.7.4 Receipt of the SEES PDU

Upon receipt of a SEES PDU (and the most recently received corresponding Entity State PDU), the receiving SEES simulations shall determine whether the signatures are detectable and use the information in the SEES PDU and predefined databases to appropriately influence infrared, acoustic, and/or radar detection equipment. The receiving simulation shall also be responsible for modeling any environmental and transmission effects among the signal source, entity, and sensor receiver. These environmental conditions shall be provided by the Environmental Process PDU (see 5.11.2.2), Gridded Data PDU (see 5.11.2.3), and static environment.

5.8 Radio and intercom communications

5.8.1 General

The protocol for the simulation of radio communications, including intercom communications, in DIS is described in 5.8.2 through 5.8.7. The Radio Communications Protocol (RCP) supports the simulation of both audio and data transmission by radio and Tactical Data Links (TDLs). In the RCP, the content of the radio communication may be conveyed in its entirety in real time or may be conveyed by reference to a prerecorded database.

The general requirements for any type of radio systems are contained in this clause. The additional requirements based on the specific type of radio are contained in Annex C. Where a conflict exists between the requirements stated here and those listed for a specific radio type in Annex C, the Annex C requirements shall take precedence.

The term *lossless propagation*, as used in this clause, means all of the transmitter power is available at the receiver input regardless of the associated transmitter and receiver antenna positions or any other antenna characteristic. In this case, such things as terrain, distance, atmospheric phenomena, and other affects on the radio signal are not taken into account.

A radio may be used to support radio communications between operational personnel in a DIS exercise and for nonoperational uses such as exercise management and technical support functions. A radio may or may not be attached to an entity or object in a DIS exercise. The Radio Attached Indicator field of the PDU Status record of the PDU Header indicates the radio attachment status associated with a Transmitter, Signal, Receiver, Intercom Control, or Intercom Signal PDU.

Intercom communications may use one of two methods, the Simple Intercom Method (i.e., Transmitter and Signal PDUs) or the Intercom Method (i.e., Intercom Signal and Intercom Control PDUs). Subclauses 5.8.6 and 5.8.7 contain information to assist in deciding on which method to use. When the Simple Intercom Method is used to indicate an intercom communications device, the requirements specified in C.3 shall apply.

5.8.2 Radio communications approach

The following approach shall be followed for radio communications in DIS:

a) Transmitting radio simulation applications shall output a Transmitter PDU and a Signal PDU to represent their state.
b) Receiving radio simulation applications shall reproduce the received signal to the fidelity level required by that particular receiver simulation. This shall be accomplished using the parameters provided in the Transmitter PDU and Signal PDU.
c) As multiple radios, including radios that are the same radio type, may be associated with the same entity, object, or unattached radio, each radio, regardless of its radio type, is assigned a sequential radio number. The Radio Number, when used together with the Radio Reference ID, uniquely identifies the radio of an entity, object, or unattached radio. This unique radio designation is referred to as the Radio Identifier (see 6.2.70).
d) The radio attachment status is indicated by the Radio Attached Indicator of the PDU Status record [see item b1) in 5.8.3.3].
e) Receiving radio simulation applications may issue Receiver PDUs to reflect the receiver state. These PDUs are for use by radio network monitors, data loggers, and similar systems.
f) The Antenna Location field in the Transmitter PDU indicates if a receiving simulation is intended to model propagation effects for the specific transmission represented by a Transmitter PDU and its associated Signal PDU(s).

5.8.3 Transmitter PDU

5.8.3.1 Purpose

The Transmitter PDU shall be used to communicate the state of a particular radio transmitter or simple intercom.

5.8.3.2 Information contained in the Transmitter PDU

The Transmitter PDU shall contain the following information:

a) Identification of the entity or other object that contains the radio transmitter for attached radios, or identification of an unattached radio.
b) Identification of the particular transmitter that is being described.
c) Identification of the type of transmitter that is being described.
d) State of the transmitter (whether it is off, on but not transmitting, or on and transmitting).
e) Source of the radio audio input (whether the pilot, co-pilot, first officer, or intentional jamming, etc.).
f) Location of the radiating portion of the antenna in both world and entity coordinates.
g) Type of representation used for the radiation pattern from the antenna [see item o)].
h) Center frequency for transmission.
i) Bandwidth of the particular transmitter measured between the half-power (–3 dB) points (this value represents total bandwidth, not the deviation from the center frequency).
j) Power. This shall be the power that the radio would be capable of outputting if on and transmitting. This is independent of the actual transmit state of the radio. This value will normally be the power setting of the radio. However, if the power is degraded (e.g., low batteries), then the reduced capability shall be reflected instead. Power shall be defined as the average effective radiated power (ERP) and shall be in units of decibel-milliwatts.
k) Type of modulation used for transmission (this includes the spread-spectrum usage, details on modulation type, and the compatibility of the emissions from the subject transmitting device).
l) Specification of the crypto or secure voice equipment. This includes information that allows applications to verify that the transmitting and receiving crypto gear are using the same crypto key.
m) Modulation parameters that define the details of the radio frequency modulation used. Only one Modulation Parameters (MP) record shall be allowed in a Transmitter PDU. If no MP record is included, then the Length of Modulation Parameters field shall be set to zero.
n) Variable transmitter parameters that define other transmitter data not included elsewhere in the PDU. Multiple Variable Transmitter Parameters (VTP) records shall be able to be included in a Transmitter PDU, and each record may be of a variable length. If no VTP records are included, then the Number of Variable Transmitter Parameters Records field shall be set to zero.
o) The antenna pattern that describes the radiation pattern from the antenna, its orientation in space, and the polarization of the radiation. If no Antenna Pattern record is included, then the Antenna Pattern Length field shall be set to zero.
p) Indication of whether the associated transmission signal is subject to electromagnetic propagation losses.
q) Indication of whether the radio is attached to an entity or object.

5.8.3.3 Issuance of the Transmitter PDU

A Transmitter PDU shall be issued by the representation of an actively transmitting communication device for radios and intercoms as follows:

a) A transmitter PDU shall be issued when any of the following occur:
   1) A predetermined length of reference time has elapsed since issuing the last Transmitter PDU. This time interval shall depend on whether the transmitter is currently moving or stationary. The Transmitter PDU moving and stationary heartbeat timers shall be identified by the symbolic names HBT_PDU_TRANSMITTER and HBT_STATIONARY, respectively. (See 6.1.8 for parameter details and default values.)
   2) Any parameter in the Transmitter PDU has changed other than antenna location, antenna azimuth, or antenna elevation.
   3) The difference between the current antenna location and the antenna location in the last issued Transmitter PDU exceeds a predefined positional threshold. The positional threshold parameter shall be identified by the symbolic name TRANS_POS_THRSH. (See 6.1.8 for parameter details and default values.)
   4) The difference between the current antenna direction (azimuth and elevation) and the antenna direction reported in the last issued Transmitter PDU exceeds a predefined orientation threshold. The orientation threshold parameter shall be identified by the symbolic name TRANS_ORIENT_THRSH. (See 6.1.8 for parameter details and default values.)
   5) Simulated time has been paused.
   6) Simulated time has been restarted after a pause.
   7) Immediately preceding the first Signal PDU of a transmission when a transmission is initiated.
   8) Immediately following the final Signal PDU of a transmission when a transmission is concluded.
b) *Specific Field Requirements*
   1) Radio Attached Indicator. This field within the PDU Status field of the PDU Header record shall be set as follows:
      i) If the Radio Attached Indicator is not implemented to indicate being Unattached (1) or Attached (2), the field shall be set to No Statement (0).
      ii) If the radio is unattached to an entity or object, the field shall be set to Unattached (1).
      iii) If the radio is attached to an entity or object, the field shall be set to Attached (2).
   2) Radio Reference ID. This field shall contain a unique identifier to indicate either the entity or other object that the radio is attached to or an identifier for an unattached radio. This field shall be set as follows:
      i) Radio Attached to an Entity. If the radio is attached to an entity, the Entity Identifier record format shall be used for this field (see 6.2.28).

         NOTE—A radio that is represented as a radio entity is considered attached to itself and the Entity ID of the radio is used.

      ii) Radio Attached to an Object. If the radio is attached to an object, the appropriate Object Identifier record shall be used for this field (see 6.2.63).
      iii) Unattached Radio. If the radio is Unattached to an entity or object, the Unattached Identifier record shall be used for this field (see 6.2.92).
   3) Radio Number. Radio numbers shall be assigned sequentially to the radio(s) associated with an entity or object, starting with the number one irrespective of the radio type. If an entity or object has only one radio, this field shall be set to one. The combination of the Radio Reference ID and Radio Number uniquely identifies a radio, whether attached or unattached, within a simulation exercise and is referred to as the Radio Identifier (see 6.2.70).
   4) Parameter Records Section. There is no general requirement that any parameter records be included in the Transmitter PDU. Such records are included as required for a specific radio system type. When one or more types of parameter records are present, they shall be included in the parameter records section of the Transmitter PDU in the following order: a single

Modulation Parameters (MP) record, if any, followed by a single Antenna Pattern record, if any, followed by Variable Transmitter Parameters (VTP) records, if any.

5) Antenna Pattern. An omnidirectional antenna pattern shall be described by setting the Antenna Pattern Type field and Antenna Pattern Length field to zero. To specify other antenna patterns, the Antenna Pattern Type field shall be set to the appropriate enumeration.

6) Antenna Location. This field shall be set to 0.0, 0.0, 0.0 (all bits set to zero) to indicate a lossless propagation transmission. Otherwise, it shall be set to the world location of the radiating portion of the antenna whether attached or unattached.

7) Relative Antenna Location. This field shall be set to 0.0, 0.0, 0.0 (all bits set to zero) for unattached radios. For attached radios, this field shall be set to the relative offset from the world location of the object to which it is attached.

8) Crypto System. If an encryption system is not available for the radio, the Crypto System field shall be set to No Encryption Device (0). If an encryption system is available for the radio, whether in plain or secure mode, the Crypto System field shall be set to a nonzero value corresponding to the encryption system.

9) Crypto Key ID. To indicate plain communications, the Pseudo Crypto Key field shall be set to zero and the Crypto Mode shall be set to Baseband (0) encryption mode. To indicate secure communications, the Pseudo Crypto Key field shall be set to a nonzero value corresponding to the simulated cryptographic key:

   i) The Crypto Mode shall be set to Baseband (0) to indicate that the crypto equipment is in the Baseband encryption mode. It shall be set to Diphase (1) to indicate that the crypto equipment is in the Diphase encryption mode.

   ii) If the transmitter has no encryption capability, or it is operating in plain communications mode, then the Pseudo Crypto Key shall be set to zero. If the Pseudo Crypto Key of the transmitter and receiver match, they shall be considered to be using the same encryption key. Note that this Pseudo Crypto Key is not an actual crypto key, the nature of which varies among encryption systems and is classified. The Pseudo Crypto Key setting is specified in the exercise agreement.

10) Input Source. To indicate intentional jamming, the Input Source field shall be set to the appropriate enumeration for a jammer.

11) Transmit State. This field shall be set as follows:

   i) Off (0). The radio is powered off, and no effective radiated power is evident at the antenna. This status shall not affect the setting of any other field in the Transmitter PDU.

   ii) On but Not Transmitting (1). The radio is powered on but not transmitting. This status shall be set whenever the radio power is on but no transmission as indicated by Signal PDUs is occurring. This status shall continue to be transmitted whenever a Transmitter PDU is sent unless the status changes to Off (0) or On and Transmitting (2).

   iii) On and Transmitting (2). The radio is powered on and transmitting. This status shall be set in the Transmitter PDU immediately preceding the issuance of the associated Signal PDU.

12) Power. This field shall always indicate the effective radiated power that would be present if the radio were transmitting and shall not change as a function of the Transmit State field.

13) Frequency. If the radio is in a frequency hopping mode, this field shall be set as specified in Annex C for this type of radio. If this type of radio is not included in Annex C and it is in frequency hopping mode, this field may be set to the center of the frequency hopping band currently in use or to some other appropriate value.

The Transmitter PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.8.3.4 Receipt of the Transmitter PDU

Upon receipt of the Transmitter PDU, the receiving radio simulation application shall determine the effects of the transmission on the receiving radio. The effects may include the clear reception of the modulated signal, the addition of noise or jamming effects to signals already being received, or other simulation application-specific effects:

a) A Transmitter PDU shall be processed based on the specific radio system identified in the Radio System field of the Modulation Type record. If the receiving simulation does not implement the specified radio system, it shall discard the Transmitter PDU along with its associated Signal PDUs.

b) If the receiver parameters are such that the radio transmission is at least partially received and demodulated, the receiving radio simulation application shall use this as a cue to begin processing the Signal PDUs from the received transmitter.

c) If all bits of the 192-bit Antenna Location field are equal to zero (the transmitter antenna is at the center of the Earth), the receiving simulation shall consider the associated transmitted signal to be a lossless propagation transmission as defined in 5.8.1.

d) If the receiving application's radio antenna is modeled at world location 0, 0, 0 (receiver antenna is at the center of the Earth), all incoming signals shall be modeled with lossless propagation as defined in 5.8.1.

e) The receiver shall consider that the frequencies match as follows:

    1) Non-Frequency Hopping Radio. If the radio is not capable of frequency hopping (FH) or is not currently in the FH mode, the receiver shall consider that the frequencies match if there is an overlap between the transmitter and receiver frequency ranges as determined by comparing the receiver frequency value (or range of frequency values) with the range computed from the values contained in the Frequency field and the Transmit Frequency Bandwidth field.

    2) Frequency Hopping Radio. If the radio is currently in the FH mode, the receiver shall process the Frequency field value in accordance with the receipt rules specified for this specific radio as defined in Annex C. If this radio is not defined in Annex C, then the frequency value shall be ignored for the purpose of determining whether the radios can communicate.

f) Plain communications shall be indicated if either the Crypto System or the Pseudo Crypto Key, or both, are set to zero. Otherwise, an encrypted transmission is indicated (see Table 15 and Table 16):

    1) Successful secure communications shall be indicated when both the received Crypto System is compatible with the simulated receiver and the Crypto Key ID record matches the data present in the simulated receiver.

    2) Unsuccessful secure communications shall be indicated when either the received Crypto System is incompatible with the simulated receiver or the Crypto Key ID record does not match the data present in the simulated receiver.

    3) The Crypto System field shall always indicate the type of crypto system being used even if the encryption equipment is not keyed.

g) The location of the radio transmitter's antenna in world coordinates shall be determined by either of two methods:

    1) When using the first method, the location of the antenna shall be as specified by the Antenna Location field of the Transmitter PDU. As an optional extension to this method, the antenna location may be extrapolated based on the locations in previously received Transmitter PDUs. The algorithm used to perform this extrapolation shall guarantee that the extrapolated location does not deviate from the location in the last Transmitter PDU by more than the threshold described in item a3) of 5.8.3.3.

    2) When using the second method, the location of the antenna shall be calculated using a combination of the Transmitter PDU's Relative Antenna Location field and the dead reckoned model of the entity's location as computed from the corresponding Entity State PDUs (see 5.3.2.3).

h) When both a Modulation Parameters record and one or more Variable Transmitter Parameters records are included in a Transmitter PDU and they represent conflicting data, the Variable Transmitter Parameters record data shall take precedence.

i) A Transmitter shall be timed out if the associated Transmitter PDU is not received within the timeout value based on HBT_STATIONARY. (See 4.2.7.)

j) Omnidirectional and beam antenna pattern types shall be supported by all receiver simulation applications. Support for other antenna pattern types is optional.

Table 15—Plain and encrypted settings

Crypto-system	Pseudo Crypto Key	Communications mode—encryption capability
No Encryption Device (0)	Any value	Plain communications—has no encryption capability
Nonzero	Zero	Plain communications—has encryption capability
Nonzero	Nonzero	Secure communications—has encryption capability

Table 16—Examples of crypto settings

Case	Transmission		Reception (receiver using system = 2, key = 82)
	Crypto system	Pseudo Crypto Key	
1	0	0	Plain Successful
2	1	0	Plain Successful
3	0	81	Plain Successful
4	1	81	Secure Unsuccessful (System and Key mismatch)
5	2	0	Plain Successful
6	0	82	Plain Successful
7	2	82	Secure Successful

5.8.4 Signal PDU

5.8.4.1 Purpose

The Signal PDU shall be used to convey the audio or digital data carried by the simulated radio or intercom transmission.

5.8.4.2 Information contained in the Signal PDU

The Signal PDU contains the content of a radio transmission. This content may be digitized audio, binary data, or an index into a database that defines the signal.

The Signal PDU shall contain the following information:

a) Identification of the radio that is the source of the transmission.

b) Identification of the particular transmitter that is transmitting.

c) Specification of the encoding scheme utilized. A radio signal encoding type listed in Table 17 shall be required to use one of the sampling rates shown for that radio type. Radio signal encoding types not listed in Table 17 shall use any sampling rate(s) defined by the exercise agreement. The minimum encoding scheme that shall be implemented by all DIS voice communications radio simulation applications is 8-bit μ-law.

d) Specification of the type of TDL message included in the Signal PDU. It is set to Other (0) if this is not a TDL message. When the Signal PDU carries a TDL message, the TDL type is a positive integer that represents the TDL type (e.g., link 4A, link 11, etc.).

e) Sample rate in samples per second for audio data. The data rate in bits per second for digital data.

f) Length of the data fields expressed in bits.

g) Number of individual audio/voice samples.

Table 17—Required sample rates

Value	Radio signal encoding types	Required sample rates
1	8-bit μ-law (ITU-T G.711)	8000 16000
2	CVSD (MIL-STD-188-113)	8000 16000
3	ADPCM (ITU-T G.726)	8000 16000
4	16-bit Linear PCM 2’s complement, Big Endian	8000 16000 22050 32000 44100 48000
5	8-bit Linear PCM, unsigned	8000 16000
6	16-bit Linear PCM 2’s complement, Little Endian	8000 16000 22050 32000 44100 48000

5.8.4.3 Issuance of the Signal PDU

5.8.4.3.1 General issuance requirements

A Signal PDU shall be issued whenever voice or data is being transmitted. Signal PDUs will be issued under the following conditions:

a) Immediately following the Transmitter PDU that indicates the beginning of the transmission.

b) During the transmission using the following criteria:

1) When the Signal PDU contains an encoding class other than the database index, Signal PDUs shall be issued so as to maintain an uninterrupted flow of signal content. The duration of the signal content in each Signal PDU is specified by the Sample Rate field, Data Length field, and Samples field of the PDU.
2) When the Signal PDU contains an encoding class of the database index, a Signal PDU shall be issued after every Transmitter PDU. The Offset and Duration fields of these Signal PDUs shall be updated to correspond to the reference time interval into the transmission so that new simulations entering an exercise can process the transmission.

c) Preceding the Transmitter PDU that indicates the end of the transmission.

The encoding class may change within the Signal PDUs of a single transmission (e.g., to represent mixed voice and data in a TDL transmission). The encoding type used to support the encoding class of encoded audio on a particular simulated radio shall not change during the duration of a simulation exercise.

For simulations that implement the encoding class of the database index, the issuing radio simulation shall reference the same database index throughout a single transmission.

The Signal PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.8.4.3.2 Field-specific requirements

The following field-specific requirements apply:

a) *Encoding Scheme*. This 16-bit record (see Table 177) is composed of a 2-bit field specifying the encoding class and a 14-bit field specifying either the encoding type or the number of TDL messages contained in the Signal PDU as follows:
    1) Encoding Class. The two most significant bits of the Encoding Scheme record shall specify the encoding class.
    2) Encoding Type or Number of TDL Messages
        i) The 14 least significant bits of the Encoding Scheme record shall represent Encoding Type when the Encoding Class is Encoded Audio (0).
        ii) The 14 least significant bits of the Encoding Scheme record shall be zero when the encoding class is not Encoded Audio (0) and the TDL Type is Other (0).
        iii) Otherwise, the 14 least significant bits of the Encoding Scheme record shall represent the number of tactical data link messages contained in the data section of the Signal PDU.
    3) A special requirement applies to the choice of encoding scheme when analog modulated audio communication is simulated. The sample rate is in samples per second for audio data. The bit rate is in bits per second for digital data.

b) *TDL Type*. When the Data field is not representing a TDL Message, this field shall be set to Other (0).

c) *Sample Rate*. This field specifies the sample rate or data rate. The interpretation of this field depends on the value of the Encoding Class field of the Encoding Scheme record as follows:
    1) If the Encoding Class is Encoded Audio (0), this field shall specify the sample rate in samples per second.
    2) If the Encoding Class is Database Index (3), this field shall be zero.
    3) For data transmissions, this field shall specify the data rate in bits per second.

d) *Data Length*. This field specifies the length of the Data field in bits. If the Encoding Class is Database Index (3), the Data Length field shall contain the value 96.

e) *Samples*. If the Encoding Class is not Encoded Audio (0), this field shall be zero.

f) *Data*. The length of the valid data contained in this field shall be the value of the Data Length field. The Data field shall be zero-padded to comply with the overall PDU length requirements (see 6.3.2). The interpretation of this field depends on the value of the TDL Type field and of the Encoding Class field of the Encoding Scheme record as follows:
   1) If the Encoding Class is Encoded Audio (0), the Data field shall be interpreted as containing audio information digitally encoded as specified by the encoding type.
   2) If the Encoding Class is Raw Binary Data (1), the format of the data shall be specified by the TDL Type field.
   3) If the Encoding Class is Application-Specific Data (2), the first 32 bits of the Data field shall specify a User Protocol identification number (see [UID 177]). The remainder of the Data field shall be interpreted as specified by the user protocol.
   4) If the Encoding Class is Database Index (3), the Data field shall be composed of three fields:
      i) A 32-bit unsigned integer (data octets 0 to 3) index into a predefined database of prerecorded data.
      ii) A 32-bit unsigned integer (data octets 4 to 7) specifying the offset in milliseconds from the start of the index data.
      iii) A 32-bit unsigned integer (data octets 8 to 11) indicating the duration of the transmission in milliseconds from the indexed offset.

5.8.4.4 Receipt of the Signal PDU

Upon receipt of a Signal PDU, the receiving simulation application shall determine whether it has received an associated Transmitter PDU. If the Transmitter PDU was received and the simulation application determines that it can detect the transmission, then it shall represent the content of the transmission as specified by the Data field of the PDU [see item i) in 7.7.3].

When the signal contains voice communications audio, the receiving simulation application shall convert the signal into the correct sample rate and compression scheme if supported by the simulation application so that the signal will be properly received. If the simulation application does not support the sample rate or compression scheme, the signal shall not be made audible.

5.8.5 Receiver PDU

5.8.5.1 Purpose

The Receiver PDU may be used to communicate the state of a particular radio receiver. Its primary application is in communicating state information to radio network monitors, data loggers, and similar applications for use in debugging, supervision, and after-action review.

5.8.5.2 Information contained in the Receiver PDU

The Receiver PDU shall contain the following information:

a) Identification of the radio that is controlling the receiver
b) Identification of the particular receiver that is being described
c) State of the receiver (whether it is off, on but not receiving, or on and receiving)
d) Identification of the radio that is controlling the transmitter
e) Identification of the particular transmitter that is being described
f) Average power being received

5.8.5.3 Issuance of the Receiver PDU

A Receiver PDU, when applicable to a simulation, shall be issued by a radio simulation application when:

a) A predetermined length of reference time has elapsed since issuing the last Receiver PDU. The Receiver PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_RECEIVER. (See 6.1.8 for parameter details and default values.)

b) Any parameter in the Receiver PDU has changed.

c) Simulated time has been paused.

d) Simulated time has been restarted after a pause.

The Receiver PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.8.5.4 Receipt of the Receiver PDU

No positive response to a Receiver PDU shall be required of a radio simulation. Applications for which these PDUs are useful may respond appropriately.

5.8.6 Intercom Signal PDU

5.8.6.1 Introduction

The Intercom Signal PDU is one of two PDUs, along with the Intercom Control PDU, that support Intercom functionality when using the Intercom Method. The terms *intercom*, *intercom device*, and *communications device* are used interchangeably when referring to requirements related to the Intercom Method.

5.8.6.2 Purpose

The Intercom Signal PDU shall be used to convey the audio or digital data that is used to communicate between simulated intercom devices. This is one of two methods that may be used to represent an intercom. The other method uses the Transmitter and Signal PDUs (see C.3). The following factors, among others, may be taken into account when performing a cost–benefit analysis to select which intercom method to implement:

a) Signal PDU Intercom Method (Simple Intercom). This method sets certain Transmitter PDU data fields to specific values to indicate that this is a simple intercom transmission. The Signal PDU is used to convey intercom voice or data. Factors that would favor this method include the availability of existing simulated radio equipment that uses the Transmitter and Signal PDU.

b) Intercom Signal PDU Method. This method uses the Intercom Signal and Intercom Control PDUs to provide an intercom system using the DIS protocol. Factors that would favor this method include where the Intercom system is more complex with state information and no simulated radio equipment system is needed that could also provide the capability.

5.8.6.3 Information contained in the Intercom Signal PDU

The Intercom Signal PDU shall contain the content of an intercom transmission. This content may be digitized audio, binary data, or an index into a database that defines the data.

The Intercom Signal PDU shall contain the following information:

a) Identification of the entity, object, or unattached device that is the source of the transmission

b) Identification of the particular device that is transmitting

c) Specification of the encoding scheme utilized. The encoding scheme shall be as specified in item c) in 5.8.4.2
d) Specification of the type of TDL message included in the Intercom Signal PDU. The TDL type shall be as specified in item d) in 5.8.4.2
e) Sample rate, in samples per second, for audio data and/or the data rate, in bits per second, for digital data
f) Length of the data fields expressed in bits
g) Number of individual audio/voice samples
h) Audio/voice data
i) Indication of whether the intercom is attached to an entity or object

5.8.6.4 Issuance of the Intercom Signal PDU

An Intercom Signal PDU shall be issued by the representation of an actively transmitting communication device for intercoms as follows:

a) An Intercom Signal PDU shall be issued whenever intercom audio or data are being transmitted on any line of the simulated intercom device.
b) The encoding class may change within the Intercom Signal PDUs of a single transmission; however, the encoding type for a particular communications device shall not change for the duration of a simulation exercise.
c) Intercom Attached Indicator. This field within the PDU Status field of the PDU Header record shall be set as follows:
    1) If the Intercom Attached Indicator is not implemented to indicate being Unattached (1) or Attached (2), the field shall be set to No Statement (0).
    2) If the intercom is unattached to an entity or object, the field shall be set to Unattached (1).
    3) If the intercom is attached to an entity or object, the field shall be set to Attached (2).

The Intercom Signal PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.8.6.5 Receipt of the Intercom Signal PDU

Upon receipt of the Intercom Signal PDU, the receiving intercom simulation application shall determine whether the issuing simulated intercom device is currently transmitting information on a channel that the receiving simulated intercom device is simulating. If so, the receiving intercom simulation application shall convey the voice or data to the appropriate local destinations.

5.8.7 Intercom Control PDU

5.8.7.1 Purpose

The Intercom Control PDU shall be used to communicate the state of a particular intercom device, request an action of another intercom device, or respond to an action request.

5.8.7.2 Information contained in the Intercom Control PDU

The Intercom Control PDU shall contain the following information:

a) Identification of the type of request being made by the Intercom Control PDU
b) Identification of the type of communications channel that is being simulated
c) Identification of the source of this Intercom Control PDU, including:

    1) Identification of the entity, object, or unattached device associated with the intercom communications device
    2) Identification of the communications device by sequence number
    3) Identification of the line number of the communications device by sequence number
d) Transmit priority of the communications device on the line
e) Transmit line state of the communications device on the line
f) Identification of the type of Intercom Control PDU
g) Identification of the master of the intercom channel, including:
    1) Identification of the entity, object, or unattached device associated with the intercom communications device
    2) Identification of the communications device by sequence number
    3) Identification of this communications device's intercom channel by sequence number

    Together, these three fields provide an intercom channel identification that is unique exercise-wide.
h) Additional parameter information, including destination(s) of the Intercom Control PDU as required
i) Indication of whether the intercom is attached to an entity or object

5.8.7.3 Issuance of the Intercom Control PDU

An Intercom Control PDU shall be issued by an intercom simulation application when:

a) The simulated intercom device has a change in the transmit state or priority on a particular line or has any other change in state of which other simulated intercom devices need to be aware
b) An action of another simulated intercom device is requested
c) A response to a request of another simulated intercom device is made
d) Intercom Attached Indicator. This field within the PDU Status field of the PDU Header record shall be set as follows:
    1) If the Intercom Attached Indicator is not implemented to indicate being Unattached (1) or Attached (2), the field shall be set to No Statement (0).
    2) If the intercom is unattached to an entity or object, the field shall be set to Unattached (1).
    3) If the intercom is attached to an entity or object, the field shall be set to Attached (2).

The following field specific requirements apply:

— *Communications Channel Type*. The most significant bit of the Communications Channel Type field shall enumerate the following communications classes:

    0 = Simulated Communications Channel

    1 = Simulation Support Communications Channel

    The seven least significant bits of the Communications Channel Type field shall enumerate the type of communications channel.
— *Source Intercom Number*. All intercom devices associated with a particular entity or object, or an unattached intercom, shall be numbered to provide a unique identification. The Source Intercom Number shall be assigned so that all intercom devices for an intercom source have a unique Source Intercom Number. This field shall contain the value zero only if the entity or object does not have a simulated intercom device.
— *Source Line ID*. For intercom devices with a single line, this field shall contain the value one.
— *Transmit Priority*. The value zero shall be reserved, the value 1 shall denote the highest priority, and the value 255 shall denote the lowest priority.
— *Command*. When the Control Type of the Intercom Control PDU is request or acknowledge, this field shall specify the detailed type requested. Otherwise this field shall contain the value zero.

— *Master Intercom Number*. This number shall be assigned so that there is a unique channel identification among all other current Master Channel IDs created by this Master Intercom Reference ID and Master Intercom Number pair.

The Intercom Control PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.8.7.4 Receipt of the Intercom Control PDU

Upon receipt of an Intercom Control PDU, the receiving intercom simulation application shall determine the type of Intercom Control PDU. If the PDU contains only transmit state information and the receiving simulated intercom device is currently involved in a simulation of the channel indicated in the Intercom Control PDU, the receiving intercom simulation application shall take appropriate actions to change its internal state based on its internal logical model. If the PDU contains a request, the receiving simulated intercom device shall determine whether it is the intended recipient of the PDU, perform the request if possible, and send an acknowledgment to the requester if required. If the PDU contains an acknowledgment, the receiving simulated intercom device shall determine whether it is the intended recipient of the PDU and take appropriate actions based on the type of acknowledgment and the logic model.

5.9 Entity management

5.9.1 General

Protocols for controlling entity aggregation activities, communicating the state of grouped entities, transferring the ownership of an entity, and requesting a hierarchical linkage of separately hosted simulation entities are defined in 5.9.2 through 5.9.5. The protocols of these four functional capabilities are not explicitly coupled, but rather they provide alternative means for addressing several entity management needs within a DIS exercise.

5.9.2 Aggregation

5.9.2.1 General

The Aggregate State PDU in conjunction with three Simulation Management PDUs provides a capability for aggregating entities and communicating information about these aggregates of entities. It also provides a mechanism for participants in a DIS exercise to request the level at which these entities are represented.

The three Simulation Management PDUs (see 5.6) used with the Aggregate State PDU are the:

a) Action Request PDU
b) Action Response PDU
c) Event Report PDU

When these Simulation Management PDUs are used in the aggregation capability of the Entity Management protocol, they shall specify the Entity Management protocol in the Protocol Family field of the PDU Header.

5.9.2.2 Aggregate State PDU

5.9.2.2.1 Purpose

The Aggregate State PDU shall be used to communicate the state and other pertinent information about an aggregated unit.

5.9.2.2.2 Information contained in the Aggregate State PDU

The Aggregate State PDU shall contain the following information:

a) Identification of the aggregate that issued the PDU
b) Identification of the force to which the aggregate belongs
c) State of the aggregate
d) Type of aggregate
e) Formation of the aggregate
f) Marking of the aggregate
g) Aggregate unit information, including:
    1) Size of the bounding volume
    2) Orientation
    3) Location of the center of mass
    4) Linear velocity
h) Number of Aggregates IDs
i) Number of Entity IDs
j) Number of Silent Aggregate Systems
k) Number of Silent Entity Systems
l) Identification for each subaggregate issuing Aggregate State PDUs
m) Identification for each entity that is a part of the aggregate and that is also issuing Entity State PDUs
n) Information about each type of silent subaggregate (not issuing Aggregate State PDUs), including:
    1) Number of silent subaggregates of this type
    2) Type of aggregate
o) Information about each type of silent entity (not issuing Entity State PDUs), including:
    1) Number of silent entities of this type
    2) Number of appearance records
    3) Type of entity
    4) Appearance records for some of these entities
p) Exercise-specific information

Simulation applications representing the aggregate have to be careful not to exceed MAX_PDU_SIZE_BITS for the Aggregate State PDU.

5.9.2.2.3 Issuance of the Aggregate State PDU

The aggregate shall transmit an Aggregate State PDU when a predetermined length of real-world time has elapsed since the issuing of the last Aggregate State PDU. The Aggregate State PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_AGGREGATE_STATE. (See 6.1.8 for parameter details and default values.) Each nonsilent aggregate shall issue an Aggregate State PDU. A silent aggregate shall not issue an Aggregate State PDU when it is being represented in the Aggregate State PDU of a higher level aggregate. Also, immediately after a unit is aggregated or disaggregated, the aggregate shall transmit an Aggregate State PDU to indicate the new state.

The following field specific requirements apply:

a) *Aggregate IDs*. This is the list of subaggregates that are part of this aggregate and are issuing Aggregate State PDUs. Each subaggregate is also assigned an Aggregate Identifier, and each one issues an Aggregate State PDU with its Aggregate ID. If a subaggregate in this list no longer has an

Aggregate State PDU issued for it, it shall revert to a silent aggregate and its Aggregate ID shall be removed from the list.

b) *Entity IDs.* This is the list of entities for which an Entity State PDU is being issued. Each entity in this list is also transmitted as a separate Entity State PDU. If an entity included in the Entity ID list no longer has an Entity State PDU issued for it, it shall revert to a silent entity and its Entity ID shall be removed from the list.

c) *Number of Appearance Records*. Within the Silent Entity Systems field, the Number of Appearance records field shall be a number between zero and the value of the Number of Entities field. Simulation applications representing the aggregate that do not model entity appearances shall set the Number of Appearance records field to zero. Simulation applications representing the aggregate that model entity appearances shall set this field to the number of entity appearances that deviate from the default appearance.

The Aggregate State PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.9.2.2.4 Receipt of the Aggregate State PDU

Upon receipt of the Aggregate State PDU, a simulation application shall determine whether the PDU contains more current information than that currently being used to model the transmitting aggregate. If so, the simulation application shall use the information contained therein to model the position and orientation of the aggregate that issued the PDU. If the receiving simulation models entity appearance, any entity appearances not specified shall be set to the default appearances. If the PDU does not contain more current information than that currently being used to model the transmitting aggregate, then the PDU shall be discarded.

5.9.2.3 Action Request PDU

5.9.2.3.1 Purpose

The Action Request PDU shall be used in the aggregation capability of the Entity Management protocol to request that a unit be represented at a different level.

5.9.2.3.2 Information contained in the Action Request PDU

The Action Request PDU shall contain the following information:

a) Identification (ID) of the aggregate requested to change state (Receiving Entity ID)
b) Identification of the entity, aggregate, or simulation requesting the change of state (Originating Entity ID)
c) Identification number of the request to change state
d) Identification of the specific action to be taken
e) Data required for performing the requested action as follows:
    1) Requested state of the aggregate
    2) Trigger used to determine this request
    3) Detection range of the entity requesting the change of state
    4) Estimated simulation time that the unit is requested to be in the new state

5.9.2.3.3 Issuance of the Action Request PDU

The Action Request PDU shall be issued by any application that wants a unit to be represented at a different level and shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.9.2.3.4 Receipt of the Action Request PDU

The Action Request PDU shall be received by the aggregate. If the aggregate receives conflicting requests, it is recommended, but not mandatory, that the unit be represented at the lowest level requested.

5.9.2.4 Action Response PDU

5.9.2.4.1 Purpose

The Action Response PDU shall be used in the aggregation capability of the Entity Management protocol to communicate the status of the action requested in the Action Request PDU.

5.9.2.4.2 Information contained in the Action Response PDU

The Action Response PDU shall contain the following information:

a) Identification of the entity, aggregate, or simulation that sent the Action Request PDU (Originating Entity ID)
b) Identification of the aggregate requested to change state (Receiving Entity ID)
c) Identification number of the original request to change state
d) Status of the action request

5.9.2.4.3 Issuance of the Action Response PDU

The Action Response PDU shall be issued by the aggregate in response to an Action Request PDU and shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.9.2.4.4 Receipt of the Action Response PDU

Upon receipt of the Action Response PDU, the application that sent the original request shall note that the Action Request PDU was received and the status of that request.

5.9.2.5 Event Report PDU

5.9.2.5.1 Purpose

The Event Report PDU shall be used in the aggregation capability of the Entity Management protocol to communicate a change of state of an aggregate.

5.9.2.5.2 Information contained in the Event Report PDU

The Event Report PDU shall contain the following information:

a) Identification of the aggregate issuing the PDU
b) Identification of the simulation application to receive the Event Report PDU
c) Identification of the type of change of state that has been requested
d) Identification of the status of the completion of the request

5.9.2.5.3 Issuance of the Event Report PDU

The Event Report PDU shall be issued by the aggregate to warn other simulations of a change state request that may affect them. It may also be issued to warn other simulations that the simulation applications representing the aggregate want to change the state of a unit. The exercise participants have to determine before the exercise which requested states are reported. If a response is desired, then the exercise participants have to also determine how long the aggregate will wait for that response. If no value is specified, the default shall be AGG_RESPONSE_DFLT. Typically, the aggregation of a unit is reported, while the disaggregation of a unit is not reported.

The Event Report PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.9.2.5.4 Receipt of the Event Report PDU

Upon receipt of the Event Report PDU, the application shall determine whether the requested change of state affects them. If the status of the change state request is pending, then other simulations may request another state.

5.9.2.6 Aggregate states

This subclause defines the possible states of an aggregate:

a) *Aggregated.* The state in which a unit is issuing an Aggregate State PDU only for itself. Aggregates in this state shall not issue PDUs for any subparts of the aggregate. Aggregate units are usually simulated at the aggregated level.

b) *Fully Disaggregated.* The state in which a unit is issuing Entity State PDUs for its entities, but it does not consistently issue Aggregate State PDUs for itself. It is possible that some simulations will not know about the aggregate if they enter the exercise after the aggregate stops issuing Aggregate State PDUs. The advantage of this state is that it allows the simulation application representing the aggregate to transfer control of the constituent entities and ignore the unit until it is requested to aggregate. The simulation application representing the aggregate also does not have to correlate the state information of aggregates and their constituent entities. These entities are capable of normal interaction with other entities at the virtual level.

c) *Disaggregated.* The state in which a unit is consistently issuing Entity State PDUs for all of its entities and Aggregate State PDUs for itself. This state requires more of the simulation application representing the aggregate, because it has to maintain the correlation between the aggregate-level and entity-level representations of the aggregate. This state also keeps other simulations updated as to the state of the aggregate. Entities in a disaggregated unit are capable of normal interaction with other entities at the virtual level.

d) *Pseudo-disaggregated.* The state in which a unit is issuing Entity State PDUs for its entities and is issuing Aggregate State PDUs for itself, but the entities in the aggregate are not controlled at the virtual level. These entities are not capable of full interaction with other entities. They may be seen and targeted, but they may not respond normally to events in the virtual world. This state of disaggregation reduces the processing power needed to visually represent aggregate units at the virtual level.

e) *Partially Disaggregated.* The state in which a unit is between aggregated and disaggregated. A partially disaggregated unit issues PDUs for some parts of the aggregate, but it is not issuing Entity State PDUs for all of its entities. A unit having one or more subaggregates that are issuing Aggregate State PDUs is in the partially disaggregated state. A unit having some entities that are issuing Entity State PDUs and other entities that are silent is also in the partially disaggregated state. The unit shall issue its Aggregate State PDU to represent the multiple parts of the aggregate. This state requires the simulation application representing the aggregate to correlate the multiple representations of the unit. (See 5.9.2.9 for correlation details.)

5.9.2.7 Aggregate hierarchy

The aggregation capability of the Entity Management protocol is designed to indicate hierarchical grouping of entities. An aggregate may consist of subaggregates and/or entities. Aggregates should be used to indicate a common characteristic among the entities and subaggregates. Entities and aggregates may be grouped together for a number of reasons. The most common method of grouping entities is by representing their military hierarchy. For example, one Aggregate State PDU could represent one tank battalion that consists of three tank companies. These three tank companies could be represented by three Aggregate State PDUs with 20 tanks in each.

5.9.2.8 Dead reckoning

A simulation that wishes to track an aggregate unit shall use a first-order Dead Reckoning Model (DRM) [see Annex E algorithm DRM (FPW) or DRM (FPB)].

5.9.2.9 Correlation issues

5.9.2.9.1 General

When a unit is in the disaggregate or partially disaggregate state, it is represented at both the aggregate level and the entity level. These two representations need to be correlated in some way. The Entity Management protocol does not specify a specific method to correlate these representations, but it does require that a method of correlation be determined before the exercise starts. The following serves as a guide to users of the Entity Management protocol in determining a correlation method.

5.9.2.9.2 Fields to correlate

The following is a list of the fields in the Aggregate State PDU that might change due to events at the virtual level. These fields may or may not be correlated with the virtual world depending on the pre-exercise correlation specifications:

a) *Position.* As the positions of the entities change, the position of the aggregate should also change.

b) *Velocity.* As the velocities of the entities change, the velocity of the aggregate should also change.

c) *Orientation.* As the orientations of the entities change, the orientation of the aggregate should also change.

d) *Boundaries.* As the positions of the entities change, the boundaries of the aggregate should also change.

e) *Entity IDs.* As entities are destroyed or disabled, they may or may not be removed from the aggregate.

5.9.2.9.3 Timing to correlate

The reference time interval for transmitting the Aggregate State PDU and the accuracy of the calculations are defined in the exercise agreement. For example, the simulation application representing the aggregate could calculate the current position of the aggregate from the latest Entity State PDUs, or it could use Entity State PDUs that were issued 30 s previously.

5.9.2.9.4 Entities to correlate

Which entities are included in the calculations of the new aggregate state information is defined in the exercise agreement. For example, all the entities in the Entity IDs could be included in the calculations, or the calculation could include only those entities that are still active. The decision of whether destroyed or disabled entities will be left in the Entity IDs list is defined in the exercise agreement.

5.9.2.9.5 Example of a correlation method

This example is provided for two reasons: to show the issues that need to be determined by the exercise participants and to provide a method that can be used if the exercise participants cannot agree on another method:

a) Fields to correlate:
   1) Position
   2) Velocity
   3) Orientation
   4) Boundaries
   5) Entity IDs
b) Timing:
   1) When a unit is in the disaggregated state, the Aggregate State PDU should be transmitted at least once every HBT_PDU_AGGREGATE_STATE seconds.
   2) When a unit is in the aggregated state, the Aggregate State PDU should also be transmitted at least once every HBT_PDU_AGGREGATE_STATE seconds.
   3) When a unit is in the disaggregated state, the fields in the Aggregate State PDU should be calculated using entity state information that is less than 2 × HBT_PDU_AGGREGATE_STATE seconds old.
c) Entities to correlate:
   1) Any entity that is destroyed or disabled should be removed from the Entity IDs and shall not be included in the aggregate state information.
   2) All entities referenced in the Entity IDs should be included in the calculations of the aggregate state information.

5.9.2.10 Changing the state of an aggregate (simulation initiated)

The PDUs that shall be sent when a simulation wants a unit to change state are as follows:

a) *Action Request PDU.* The simulation desiring a change of state shall transmit an Action Request PDU.
b) *Action Response PDU.* The aggregate shall respond with an Action Response PDU. If the aggregate is uncertain that it should complete the request, then it shall respond with a pending status.
c) *Event Report PDU.* If the exercise participants specified the requested state as one to be reported, then the aggregate shall issue an Event Report PDU. If the action is pending, then the aggregate shall wait AGG_RESPONSE_DFLT seconds for other simulations to request the current state.
d) *Aggregate State PDU.* If the state change is completed, then the aggregate shall issue an Aggregate State PDU to indicate the new state of the unit.

5.9.2.11 Changing the state of an aggregate (aggregate initiated)

The PDUs that shall be sent when the aggregate wants to change state are as follows:

a) *Event Report PDU.* If the exercise participants specified the new state as one to be reported, then the aggregate shall issue an Event Report PDU. If the action is pending, then the aggregate shall wait AGG_RESPONSE_DFLT seconds for other simulations to request the current state.

b) *Aggregate State PDU.* If the state change is completed, then the aggregate shall issue an Aggregate State PDU to indicate the new state of the unit.

5.9.3 IsGroupOf PDU

5.9.3.1 Purpose

The IsGroupOf PDU shall communicate information about the individual states of a group of entities, including state information that is necessary for the receiving simulation applications to represent the issuing group of entities in the simulation applications' own simulation.

Only entities originated by the simulation that is issuing the IsGroupOf PDU are eligible to be contained in that PDU due to the dependency on a master Site Number and Application Number for all group entities. This means that a local entity with a different Site Number and Application Number from that of the Group Entity Identifier Site Number and Application Number is not eligible to be included in that IsGroupOf PDU. This condition is normally the result of an ownership transfer.

5.9.3.2 Issuers of the IsGroupOf PDU

The IsGroupOf PDU may be issued by:

a) A node server for the purpose of communications bandwidth reduction:
   1) The node server shall receive Entity State PDUs from simulation applications communicating on the local network and generate a single IsGroupOf PDU for each appropriate or logical grouping of entities.
   2) An appropriate or logical grouping of entities may be all entities of a specific category within the local network, entities of a specific category separated by force identification, or entities of a specific category grouped within identified tactical units.

b) A computer-generated forces (CGF) application of either automated or semi-automated force type.

5.9.3.3 Information contained in the IsGroupOf PDU

The IsGroupOf PDU shall contain the following information:

a) Identification of the group of entities represented by the PDU
b) Category of the entity types represented by the PDU
c) Number of entities grouped within the PDU
d) Group reference point
e) Grouped entity descriptions consisting of entity state information, formatted as a function of the category of the grouped entity, including:
   1) Individual entity identification
   2) Individual entity location described in offset from the group reference point
   3) Entity appearance
   4) Entity orientation
   5) Entity speed
   6) Entity weapons or articulated part(s) appropriate to the entity category
   7) (Optional) Logistics status indicators of fuel, ammunition, and maintenance

5.9.3.4 Issuance of the IsGroupOf PDU

5.9.3.4.1 General

A simulation may remove an entity from a group and issue it as a separate Entity State PDU. If this occurs, the entity shall be removed from the IsGroupOf PDU by decrementing the Number of Grouped Entities field and removing the Group Entity Description (GED) record for that entity.

A simulation shall issue an IsGroupOf PDU when any of the following occur:

a) The discrepancy between the actual state of any grouped entity (as determined by its own internal model) and its dead reckoned state (state using specified dead reckoning algorithms) exceeds a predetermined threshold. This threshold includes changes in position/orientation and articulation parameter information.
b) A change in the grouped entity's appearance occurs.
c) A change in the number or identification of the grouped entities occurs.
d) A predetermined length of real-world time has elapsed since the issuing of the last IsGroupOf PDU. The IsGroupOf PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_ISGROUPOF. (See 6.1.8 for parameter details and default values.)

The IsGroupOf PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.9.3.4.2 Issuance of a new IsGroupOf PDU by a node server

When the IsGroupOf PDU is generated by a node server, the new IsGroupOf PDU shall incorporate the best current actual state information possessed by the node server and may include a polling of actual state information from the grouped entities.

5.9.3.4.3 Issuance of a new IsGroupOf PDU by a CGF application

When the IsGroupOf PDU is generated by a CGF application, the new IsGroupOf PDU shall incorporate the current actual state information of all entities extant in the grouping.

5.9.3.5 Receipt of the IsGroupOf PDU

5.9.3.5.1 General

Actions upon receipt of the IsGroupOf PDU are determined by the character of the receiving entity.

5.9.3.5.2 Receipt of the IsGroupOf PDU by a simulation application

Upon receipt of the IsGroupOf PDU, a simulation application shall determine, for each of the entities grouped within the IsGroupOf PDU, whether the PDU contains more current information than is currently being used to model those entities within the group being transmitted. If so, the simulation application shall use the information contained therein to model the position, orientation, and appearance of the entities contained within the IsGroupOf PDU. Otherwise, the PDU shall be discarded.

5.9.3.5.3 Receipt of the IsGroupOf PDU by a node server

Upon receipt of the IsGroupOf PDU, a node server shall determine, for each of the entities grouped within the IsGroupOf PDU, whether the PDU contains more current information than that currently being used to model those entities within the group being transmitted. If so, for each entity where the information is more current, the communications application shall generate, on the local network, an Entity State Update PDU (see 5.3.5). If the newly received entity data do not exceed the parameters of the dead reckoned state of the entity, the communications application shall discard that state information, unless the Simulation Manager (SM) requires that all entity states shall be updated whenever any IsGroupOf PDU is received. If so, an Entity State Update PDU shall be issued. When all appropriate Entity State Update PDUs have been issued to the local network, the IsGroupOf PDU shall be discarded.

5.9.3.6 Dead reckoning

A first-order dead reckoning algorithm [see Annex E algorithm DRM (FPW) or DRM (FPB)] shall be employed to estimate the position and orientation of the group in order to limit the number of PDUs to be issued.

5.9.4 Transfer ownership

5.9.4.1 General

The transfer of an entity and associated attributes from one simulation to another shall be accomplished using the Transfer Ownership process. This requirement applies to constructive, virtual, and live entity transfers and to any combination (e.g., virtual-constructive, constructive-constructive, etc.) unless otherwise indicated. This process is initiated by sending a TO PDU that contains a transfer request. The transfer of an entity may occur as a Push or Pull Transfer. These terms are defined in 3.1. The TO PDU provides a capability to transfer ownership of a complete entity from one simulation to another in an exercise. [For the HLA Real-time Platform Reference Federation Object Model (RPR FOM), this is referred to as the transfer of a base entity object.] *Annex H contains normative requirements for the issuance and receipt of PDUs that support the Transfer Ownership function.*

There is no *best effort* to maintain a certain level of fidelity and interoperability for a transferred entity. The basic fidelity and interoperability requirements to be met are specified in this subclause. Additional fidelity and interoperability requirements, and any restrictions on transfers between simulations, are expected to be addressed in exercise agreements.

A transfer transaction may be in progress at one simulation and not at the other simulation involved in the transfer due to the delay in exchanging messages. There should not be more than a few seconds when the transfer status differs between the two simulations.

Transfer requests, where a simulation simultaneously asks multiple simulations if they want to assume ownership of an entity, are not allowed. Swap transfers, where there is a mutual exchange of ownership of entities, are accomplished using separate transfer transactions, which can be either Push or Pull Transfer requests.

Any kind of entity, including an environmental process entity, but excluding an Aggregate Entity, may be transferred from one simulation to another. A simulation is deemed to have implemented the Transfer Ownership function if it initiates or receives and processes the TO PDU. A simulation that does not implement the Transfer Ownership function is not required to process any of the PDUs unique to this function. The PDUs associated with the Transfer Ownership function are listed in Table 18. The records associated with the Transfer Ownership function are listed in 5.9.4.4.

Table 18—Transfer ownership function PDUs

PDU	Transfer	
	Transaction	Nontransaction
Transfer Ownership	X	
Set Record-R	X	
Record-R	X	X
Acknowledge-R	X	
Entity State	X	
Event Report (Ownership)	X	
Data Query (Ownership)		X
Data (Ownership)		X
Record Query-R		X

5.9.4.2 Transfer ownership rules

5.9.4.2.1 General rules

The following general rules shall be met by a simulation that implements the Transfer Ownership function or requires knowledge of simulation ownership in an exercise that includes transfer ownership capable simulations, or desires to exchange internal state data outside of a transfer transaction:

a) For Simulations that implement the Transfer Ownership function, the following rules apply:
   1) The Entity ID of the transferred entity shall remain the same and shall not change as a result of either the completion or the termination of a transfer transaction.
   2) Both Simulation Management and Simulation Management with Reliability PDUs are used in addition to PDUs that belong to other families. Simulation Management with Reliability PDUs are permissible to use regardless of whether the Acknowledged service level has been requested.
   3) All references to the Entity State PDU shall be applicable to the Entity State Update and the Environmental Process PDUs unless otherwise indicated.
   4) All transfers shall use the Required Reliability Service of Unacknowledged since not all requirements of the Acknowledge Reliability Service are used.
   5) There shall be no limit on the number of cycles an entity can be transferred between simulations including being transferred back to a simulation that previously owned it.
   6) A simulation may implement either *a full or a limited transfer ownership capability* (see 3.1). See also item 11) and item 25).
   7) A simulation that implements transfer ownership shall not base the determination of whether an entity is a local or remote entity solely on the Simulation Address portion (Site Number/Application Number) of the Entity ID for the entity and shall account for the fact that the Simulation Address (Site Number/Application Number) portion of a remote entity may be identical to that of its own Simulation Address.

8) A transfer may be initiated automatically by a simulation or manually by operator action at a simulation. Likewise, the processing of a TO PDU or Acknowledge-R PDU by a simulation may be the result of automatic logic or manual intervention by an operator. Both methods may be available to a simulation and used for different transfer transactions.
9) Simultaneous transfer transactions shall be allowed between simulations so long as different entities are involved in each transfer transaction. The number of simultaneous transfer transactions allowed is beyond the scope of this standard. There is no requirement for a simulation to be able to handle more than one transfer transaction at a time whether initiated by itself or in response to a transfer request.
10) When a transfer is in progress, it is referred to as a transfer transaction. A transfer transaction may be initiated automatically or manually. It may be responded to automatically or manually or a combination of the two methods.
11) A simulation that does not implement the *full or limited Transfer Ownership capability* shall not be required to respond to the TO PDU or any PDUs sent as part of a transfer transaction.
12) The only simulations involved in a transfer transaction shall be the Divesting Simulation and the Acquiring Simulation.
13) A transfer capable simulation shall not initiate a new local entity using an Entity ID that already exists for either a local or a remote entity currently in its database.
14) To avoid potential ownership conflicts during the rejoin of a simulation after exercise start, transfer capable simulations shall implement a means to prevent duplicate Entity IDs.
15) A simulation that implements the transfer ownership function and provides for no operator interaction shall include a capability to automatically terminate a transfer transaction started by the simulation application based on, as a minimum, timeouts. Other criterion may be applied as well.
16) Before the completion of a transfer transaction, either simulation involved in a transfer transaction may terminate the transaction by automatic or manual means.
17) The *Total Record Sets* record shall be required to be implemented for transmission by any simulation whose local data conveyed in any record exceeds the maximum length of the PDU in which it is placed. All simulations that implement a *full* or *limited* transfer ownership capability shall be required to process the *Total Record Sets* record in conjunction with a transfer.
18) The *Launched Munition* record is required to be implemented by any simulation that can either initiate or accept a transfer request where the entity being transferred is a munition. The acquiring simulation shall not accept the transfer of a launched munition when it cannot complete the engagement.
19) The fidelity requirements associated with assuming ownership of an entity are beyond the scope of this standard.
20) Simulations implementing transfer ownership shall implement the PDU Status field – Transferred Entity Indicator subfield in the PDU Header record for the Entity State, Entity State Update, and Environmental Process PDUs, whichever of these PDUs are implemented.
21) PDUs shall be issued in the order shown in Figure 25 and Figure 26. Messages depicted with solid lines are mandatory, and those depicted with dashed lines are optional.
22) A simulation that allows for an operator-initiated transfer request shall provide feedback to the operator on the status of the request including rejection of the request by the simulation's application.
23) An entity shall only be eligible to be concurrently involved in one transfer. Once a transfer transaction is initiated for an entity, no new transfer ownership transaction for it shall be sent to another simulation until the transaction is terminated or completed.
24) Any simulation that implements either a *full* or a *limited* transfer ownership capability shall:

   i) Provide a visual indication to an operator in the form of an always-present or requested display, or access to a data file, that provides information on the current ownership of each entity in its database and on data necessary for transfer transactions. Detailed Human System Interface requirements are beyond the scope of this standard (see Annex H).
   ii) Provide the capability to manually remove from its database individual local and remote entities. This allows ownership conflicts to be resolved if automatic logic fails.
   iii) Reflect the most recent motion, orientation, and positional values received from the previous owner in the *initial* Entity State PDU, taking into account any extrapolation of such data that may occur prior to issuance of the first Entity State PDU. Further automatic changes to these parameters shall be based on logic that takes into account rates of change.
25) If a simulation that assumes ownership of an entity has the capability to continue some or all of the entity attributes (e.g., IFF and emissions data) that were being transmitted by the previous owner, the required PDUs shall be transmitted within one entity heartbeat cycle after assuming ownership. The minimum requirement for the data to be contained in such PDUs shall be to duplicate the non-PDU Header data contained in the last PDU of each type received from the previous owner that was processed prior to assuming ownership.
26) When a transaction is canceled, both simulations shall restore their database and other system status information for the entity to their original condition as it existed before the transaction was initiated. A canceled transaction shall not result in an ownership conflict.
27) No entity that is part of an aggregation shall be eligible to be transferred as a single entity unless it is removed from the aggregation to which it belongs and established as a separate, standalone entity.

b) For simulations that do not implement the Transfer Ownership function, the following rules apply:
   1) A simulation that does not implement the transfer ownership capability, but needs to determine ownership of entities during an exercise, shall implement the following:
      i) Receive and process the *ownership* Event Report PDU.
      ii) Issue the *ownership* Data Query PDU.
      iii) Receive and process the *ownership* Data PDU in response, as a minimum, to an *ownership* Data Query PDU it originated.
      iv) Have the ability to process the Transferred Entity Indicator subfield of the PDU Status field in the PDU Header record.
      v) A simulation may also receive and process an *ownership* Data PDU that is not in response to an *ownership* Data Query PDU that it originated for the purpose of maintaining its ownership data base.
      vi) The simulation shall comply with the rules set forth for the use of these PDUs in 5.9.4.
   2) If a simulation does not process ownership information as defined in item b1) above, it shall recognize the existence of the PDU Status field in the PDU Header record as a legitimate field and not as a Padding field, unless it does not perform a validity check that a Padding field is set to zero.

c) If a simulation implements a *limited* transfer ownership capability, it shall:
   1) Respond to any received TO PDU that initiates a Push or Pull Transfer that it does not implement. The response shall be an Acknowledge-R PDU indicating Unable to Comply (2).
   2) Comply with all rules associated with the type of transfer it can originate or respond to.

d) *Transfer Initiation Rules*
   1) A transfer transaction is initiated by sending a TO PDU that contains a request to transfer an entity. The valid Transfer Types and associated (enumerations) are as follows:
      i) Push Transfer. Push Transfer—Entity (1), Push Transfer—Environmental Process (4).

   ii) Pull Transfer. Automatic Pull Transfer—Entity (2), Automatic Pull Transfer—Environmental Process (5), Manual Pull Transfer—Entity (8), and Manual Pull Transfer—Environmental Process (9).

2) Only an Active entity shall be eligible to be transferred as indicated by the Entity State PDU Entity Appearance record State field (bit 23) being set to Active (0).

e) *Transfer Transaction Completion Rules*

1) The transfer transaction for an entity shall be considered completed by the *Acquiring Simulation* when it sends the *initial* Entity State PDU for the entity for which it assumed ownership. The *ownership* Event Report PDU for the entity shall be sent by the Acquiring Simulation immediately following the transmission of the *initial* Entity State PDU and before the TO_AUTO_RESPONSE_TIMER expires. The transfer transaction shall be considered completed at the *Divesting Simulation* upon receipt of the *initial* Entity State PDU from the Acquiring Simulation for this transaction.

2) The divesting simulation shall continue to have ownership, process interactions, and output information on the entity until it receives the *initial* Entity State PDU transaction from the Acquiring Simulation involved in this transfer transaction.

3) When an acquiring simulation assumes ownership of an entity, it shall output all required PDUs within one heartbeat period following the receipt of all necessary information from the divesting simulation. Subsequent PDUs for the entity shall be output as required by Entity State PDU issuance rules (see 5.3.2.4).

4) The values contained in the initial PDUs output by the acquiring simulation shall reflect the latest information received from the previous owner for those attributes and capabilities, as a minimum, that are required to be continued as specified in item 7) below.

5) Whenever the acquiring simulation assumes ownership of an entity, the PDU Status record Transferred Entity Indicator field (bit 0) shall be set as follows for the Entity State, Entity State Update, and Environmental Process PDUs only:

   i) If the Site Number/Application Number of the Entity ID is the same as its own Site Number/Application Number, it shall be set to No Difference (0).

   ii) If the Site Number/Application Number of the Entity ID is not the same as its own Site Number/Application Number, it shall be set to Difference (1).

6) Regardless of which PDUs are generated by the acquiring simulation, once the transfer transaction is completed, the divesting simulation shall cease transmitting all PDUs where the entity was the primary Entity ID in the PDUs (e.g., Entity State, IFF, and EE PDUs). No final PDUs of any kind shall be issued for the transferred entity.

7) The following data fields of the Entity State PDU shall be maintained by the new owner using the last values received from the divesting owner and shall not be subsequently changed for the life of the entity:

   i) Entity ID.

   ii) Entity Type and Alternate Entity Type.

   iii) Markings.

f) *Transfer Termination Rules*

A transfer transaction may be terminated before being completed. A transfer transaction that is in progress may be canceled by either the Acquiring Simulation or the Divesting Simulation. The simulation initiating the *Cancel* TO PDU, i.e., Transfer Type set to Cancel Transfer (7), shall terminate the transaction and disregard any further transfer-related PDUs received from the other simulation for the canceled transaction. The simulation receiving a valid *Cancel* TO PDU from the other simulation shall also terminate the transaction and cease transmitting any further transfer-related PDUs for the transaction. A cancellation of a transfer transaction due to any of the following conditions shall occur immediately upon detection of the condition. The following rules apply to the simulations and the associated entity that is involved in a specific transfer transaction that is terminated:

1) A transfer transaction shall be automatically canceled if any of the following conditions exist:
   i) A valid *Cancel* TO PDU is received from the other simulation involved in a transfer.
   ii) An appropriate Acknowledge-R PDU is issued by a simulation involved in a transfer transaction indicating Unable to Comply (2). The issuing simulation cancels the transfer transaction when the PDU is issued and the receiving simulation cancels the transfer transaction when the PDU is processed.
   iii) A timer expires without receiving an appropriate message from the other simulation. In this case, the *Cancel* TO PDU shall be sent by the simulation whose timer expired.
   iv) The required number of Set Record-R PDUs are not received. In this case, the *Cancel* TO PDU shall be issued by the acquiring simulation.
   v) A message is received out of sequence.
   vi) A value is received in a data field of either a PDU or a record inside a PDU that is outside the valid range of values.
2) If a transfer transaction is terminated, the *Cancel* TO PDU shall be sent by the simulation initiating the termination.

g) *Ownership Conflict Rules*

An ownership conflict exists when two or more simulations are transmitting an entity with the same Entity ID. Ownership conflicts generally occur due to either transfer ownership anomalies or when two or more simulations have the same Site Number and Application Number. A simulation that implements either the *full* or *limited* transfer ownership shall be capable of detecting an ownership conflict between itself and another simulation. See 3.1 and Annex H. Ownership conflicts may be resolved by automatic logic or manual operator action. All transfer-capable simulations shall have the capability to manually deactivate a local entity without sending out a *final* Entity State PDU and to process a received Remove Entity TO PDU:

1) Ownership Conflict Detection. Ownership conflict detection for transfer-capable simulations shall be an automatic process. An ownership conflict shall be detected if any of the following messages are received with an Entity ID that is a duplicate of a locally held, active entity and the entity is currently not involved in a transfer transaction:
   i) An Entity State or Entity State Update PDU.
   ii) An *ownership* Event Report with Ownership Status = Ownership Conflict (3).
   iii) An *ownership* Event Report with Ownership Status = New Owner (1).
   iv) If an ownership conflict is detected and the simulation detecting the conflict does not resolve it automatically, the simulation shall send an *ownership* Event Report with Ownership Status = Ownership Conflict (3).
2) Ownership Conflict Resolution. An ownership conflict may be resolved by any of the following methods:
   i) Automatic conflict resolution based on the rules described in item 3) below.
   ii) Manual conflict resolution based on the rules described in item 5) below.
   iii) Automatic conflict resolution based on processing a Remove Entity TO PDU as described in item 6) below.
3) Automatic Ownership Conflict Resolution. An ownership conflict may be automatically resolved only by the divesting simulation from a previous transfer in which the transfer was terminated by the divesting simulation prior to normal completion [see item f) in 5.9.4.2.1, Transfer Termination Rules]. The following rules shall apply if a simulation elects to implement this option and remove its local entity:

i) The entity shall be removed as an active, local entity in the database. If retained in the database, it shall be considered a deactivated local entity.

ii) No *final* Entity State or Entity State Update PDU shall be sent [i.e., no PDU with the Entity Appearance record State field (bit 23) set to Deactivated (1)].

iii) Any associated PDUs shall cease to be transmitted without sending a final PDU for them.

iv) An *ownership* Event Report PDU with Ownership Status = Local Entity Canceled – Auto Resolve Conflict (4) shall be issued by the simulation.

4) Ownership Conflict Dissemination. If a transfer-capable simulation detects an ownership conflict with one of its local active entities and it does not resolve the conflict automatically, an *ownership* Event Report PDU with Ownership Status = Ownership Conflict (3) shall be issued by the simulation.

5) Manual Ownership Conflict Resolution. An ownership conflict may be manually resolved by operator action. The operator may resolve the conflict by coordination with another simulation to have it drop its local entity or by manually dropping its own local entity. If the conflict is manually resolved, the simulation that canceled its local entity shall send an *ownership* Event Report PDU with Ownership Status = Local Entity Canceled – Manual Resolve Conflict (5). No *final* Entity State PDU shall be sent in this case.

6) Remove Entity TO PDU. The Remove Entity TO PDU may be used by a simulation to cause a transfer-capable simulation to automatically remove, or deactivate, its local entity from the exercise. The use of the Remove Entity TO PDU shall be governed by the exercise agreement. If the conflict is resolved in this manner, the simulation that canceled its local entity shall send an *ownership* Event Report PDU with Ownership Status = Local Entity Canceled – Remove Entity TO Received (6). No *final* Entity State PDU shall be sent in this case.

h) *Munition Transfer Rules*

The transfer of a munition entity shall be accomplished using the following additional rules:

1) An acquiring simulation shall only take ownership of a munition entity that has been fired from another entity if it has an engagement capability, including generation of the Detonation PDU, that would allow it to continue the engagement for this type of munition.

2) The divesting simulation shall send the Launched Munition record for the entity that the acquiring simulation shall process, store, and use the data either in the eventual Detonation PDU or in a new Launched Munition record if the entity is transferred to someone else before it detonates.

3) The acquiring simulation shall be able to continue the engagement of a specific target entity if one is contained in the Launched Munition record. However, the engagement may be subsequently broken if its internal engagement logic determines that the intercept is not feasible.

i) *Transfer Gateway Rules*

A transfer gateway may be used to provide some or all of a transfer ownership capability to a host simulation where the host simulation is not able to completely incorporate this functionality. If a transfer gateway is used, it shall manage the DIS traffic to and from the host simulation so that it is in compliance with the applicable rules contained in 5.9.4.

j) *Transmission of Internal State Data Outside of a Transfer Transaction*

The transmission of internal state data for an entity that is related to a decision whether to transfer the entity, or in preparation for such a transfer, shall be accomplished using the Record Query-R PDU and Record-R PDU. All other transmissions of internal state data for reasons not related to transfer ownership may be accomplished using any of the PDUs that have a capability to request or transmit internal state data. There is no requirement for a simulation that implements transfer ownership to initiate or respond to a Record Query-R PDU.

k) *Automatic and Manual Transfer Mode*

A simulation shall be considered in the Automatic Transfer mode if there is no operator manual intervention in the decision to comply or not comply with a transfer request. A simulation shall be considered in the Manual Transfer mode if there is the capability for an operator to make a decision whether to comply or not comply with a transfer request:

1) Regardless of which transfer mode a simulation is operating in, when it receives a transfer request, the initial Acknowledge-R PDU shall be automatically generated. If the simulation is in the Automatic Transfer mode, it shall set the Response Flag to Able to Comply (1) or Unable to Comply (2). If it is in the Manual Transfer mode, it may also respond with the Response Flag set to Able to Comply (1) or Unable to Comply (2) if it has initial automatic validity checks that would preclude it from complying with the request. Otherwise, it shall set the Response Flag to Pending Operator Action (3).
2) Manual operator actions to initiate a transfer request or cancel one may occur in both the automatic and manual transfer modes.
3) A simulation operating in the Manual Transfer mode shall alert an operator to a transfer request for which a Pending Operator action response was returned to the originator of the request. This alert and any decision support information shall be made available to the operator such that a final response can be sent within the time limits imposed by the Manual Response timer.

5.9.4.2.2 Push Transfer

The rules and associated message sequence for a Push Transfer are specified in this subclause. A Push Transfer is where the present simulation who owns an entity desires to transfer the entity to another simulation. A request is sent from the present owner (divesting simulation) to the potential new owner (acquiring simulation). This request includes any internal state data. The acquiring simulation then responds with an Acknowledge-R PDU indicating whether it can take the entity or not. It may initially respond to indicate that an operator will make the decision and then follow up with the final decision. Once the acquiring simulation has indicated that it can take the entity from the divesting simulation, it shall send the *initial* Entity State PDU, *ownership* Event Report PDU, and any other PDUs that would normally be sent for a new local entity. Some optional PDUs are also involved in the process. Timers are set by each simulation at different points in the transfer process to allow the transfer to be automatically terminated if applicable:

a) *Push Transfer Rules*

A Push Transfer shall comply with the general transfer rules specified in 5.9.4.2.1, 5.9.4.3, and 5.9.4.4 in addition to the specific Push Transfer rules listed in item b) below.

b) *Push Transfer Message Sequence*

The message sequence and associated rules shall be as defined in this paragraph and depicted in Figure 25. The steps listed here correlate with those shown in Figure 25. If there is a conflict between this paragraph and the content of Figure 25, the requirements defined here shall be considered superseding:

1) Step 1. The Divesting Simulation transmits a TO PDU that indicates a transfer request [see item d1ii) in 5.9.4.2.1] with the TO PDU set as specified in 5.9.4.3 to a specific Acquiring Simulation. Required internal state data inside Record Sets shall be included, if applicable. If there is no internal state data to be transmitted in the TO PDU, then the PDU shall be issued with the Number of Record Sets field set to zero.
2) Step 2. If the number of Record Sets needed to send all the internal state data exceeds the maximum length of the TO PDU, the remaining Record Sets shall be included in one or more Set Record-R PDUs sent with the same timestamp and following the TO PDU.

3) Step 3. If the Acquiring Simulation is in the manual mode, an Acknowledge-R PDU with the Response Flag set to Pending Operator Action (3) shall be automatically sent to the Divesting Simulation, and the Acquiring Simulation operator shall be alerted to the transfer request. Otherwise, this step is skipped.

4) Step 4. If an Acknowledge-R PDU with the Response Flag set to Able to Comply (1) is sent, the transfer transaction shall be continued and step 5) through step 7) shall be executed. If an Acknowledge-R PDU with the Response Flag set to Unable to Comply (2) is sent, the transfer transaction shall be canceled by both parties.

5) Step 5. (Optional) One or more Record-R PDUs may be optionally sent by the Acquiring Simulation to reflect the content and exact values of the record(s) received from the Divesting Simulation. The records may be placed in the Record-R PDU in any order. A Divesting Simulation shall not be required to process the Record-R PDU when received as part of a transfer transaction.

6) Step 6. The Acquiring Simulation shall send the *initial* Entity State PDU before the TO_AUTO_RESPONSE_TIMER expires. If there are additional PDUs related to the entity, they shall be sent in accordance with the simulation's requirements for sending such PDUs for a newly initiated local entity. Upon receipt of the *initial* Entity State PDU from the Acquiring Simulation, the Divesting Simulation shall cease transmission of any PDUs for the transferred entity. At this point, the Divesting Simulation no longer owns the transferred entity. No *final* Entity State PDU with the Entity Appearance record State field (bit 23) set to Deactivated (1) shall be transmitted by the Divesting Simulation for a transferred entity. The exact process whereby a Divesting Simulation's local entity is transitioned to a remote entity and an Acquiring Simulation's remote entity is transitioned to a local entity is beyond the scope of this standard.

7) Step 7. The Acquiring Simulation shall also send the *ownership* Event Report PDU before the TO_AUTO_RESPONSE_TIMER expires. This PDU shall follow the *initial* Entity State PDU but may be interspersed with other PDUs so long as it is sent before the timer expires.

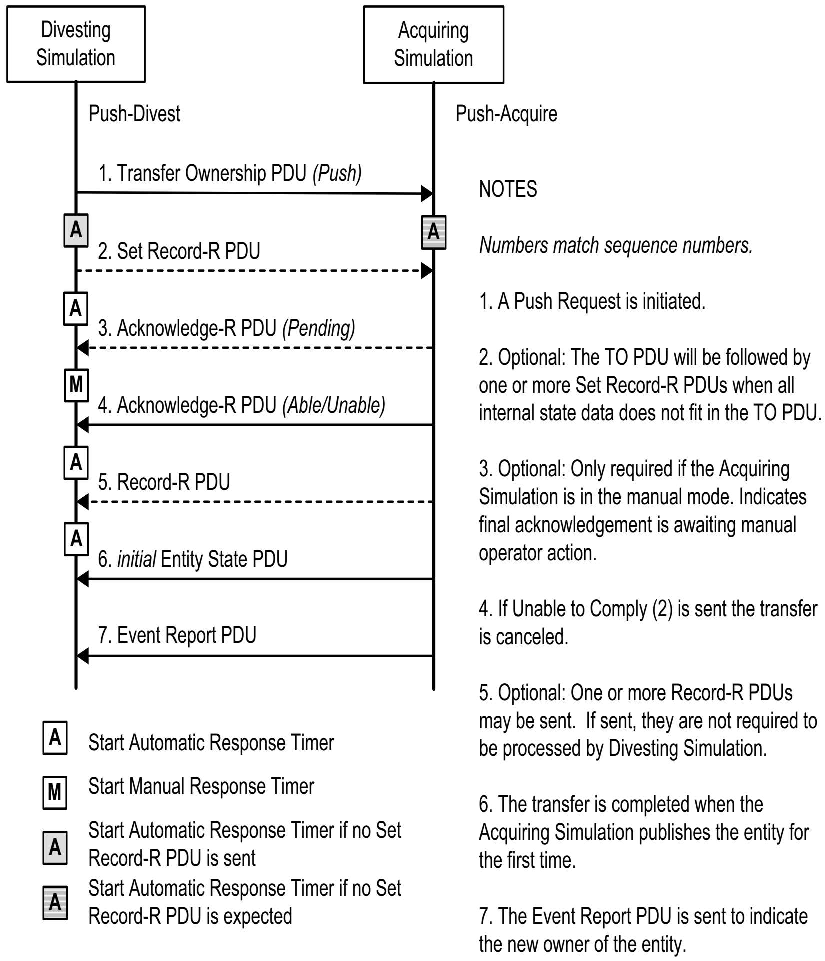


Figure 25—The Push Transfer

5.9.4.2.3 Pull Transfer

The rules and associated message sequence for a Pull Transfer are specified in this section. A Pull Transfer is where a simulation asks to take ownership of someone else's entity. A request is sent from the potential new owner (Acquiring Simulation) to the present owner (Divesting Simulation). The divesting simulation then responds with an Acknowledge-R PDU indicating whether or not it will relinquish ownership of the entity. It may initially respond to indicate that an operator will make the decision and then follow up with the final decision. Once the divesting simulation indicates that it will relinquish the ownership entity, it will send any applicable internal state data. The acquiring simulation then sends the *initial* Entity State PDU, *ownership* Event Report PDU, and any other PDUs that would normally be sent for a new local entity. Some optional PDUs are also involved in the process. Timers are set by each simulation at different points in the transfer process to allow the transfer to be automatically terminated if applicable PDUs are not received within timeout:

a) *Pull Transfer Rules*

   1) A Pull Transfer shall comply with the general transfer rules specified in 5.9.4.2.1, 5.9.4.3, and 5.9.4.4 in addition to the specific Pull Transfer rules listed in item b) below.

2) A simulation shall not send a *Pull* Transfer Request if, as a minimum, any of the following conditions exists:
   i) It is at entity capacity such that it could not store the transferred entity when the transfer request is made.
   ii) It cannot maintain the entity with the same entity type after the transfer is completed.
   iii) It does not currently hold the entity to be transferred as an external (remote) entity in its entity database.
3) A simulation shall not be required to both initiate and respond to a Pull Transfer request. It may do one or the other, or both.
4) If a Pull TO PDU is addressed to a transfer ownership capable simulation that is not the owner of the entity, an automatically generated Acknowledge-R PDU with Response set to Unable to Comply (2) shall be returned to the originator of the Pull TO PDU.
5) The simulation receiving a Pull Transfer Request directed to it shall verify, as a minimum, that all the following conditions are met before responding with an Acknowledge-R PDU with the Response field set to Able to Comply (1). If any one of the required criteria fails to be met, an Acknowledge-R PDU shall be sent with the Response field set to Unable to Comply (2):
   i) The entity to be transferred is held in its entity database as an internal (local) entity when the TO PDU is received.
   ii) The entity is not currently involved in a transfer transaction.
   iii) Internal criteria have been met for relinquishing ownership. Specific criteria are beyond the scope of this standard.
   iv) Eligibility checks to verify that the simulation requesting ownership is capable of maintaining the entity are beyond the scope of this standard. Such checks should be in accordance with the exercise agreement.

b) *Pull Transfer Message Sequence*

The message sequence and associated rules shall be as defined in this paragraph and depicted in Figure 26. The steps listed here correlate with those shown in Figure 26. If there is a conflict between this paragraph and the content of Figure 26, the requirements defined here shall be considered superseding:

1) Step 1. The Acquiring Simulation transmits a TO PDU that indicates a transfer request [see item d1ii) in 5.9.4.2.1] with the TO PDU set as specified in 5.9.4.3 to a specific Divesting Simulation.
2) Step 2. If the Divesting Simulation is in the manual mode, an Acknowledge-R PDU with the Response Flag set to Pending Operator Action (3) shall be automatically sent to the Acquiring Simulation. The Divesting Simulation operator alerted to the transfer request. Otherwise, this step is skipped.
3) Step 3. If an Acknowledge-R PDU with the Response Flag set to Able to Comply (1) is sent, the transfer transaction shall be continued and step 5) through step 7) shall be executed. If an Acknowledge-R PDU with the Response Flag set to Unable to Comply (2) is sent, the transfer transaction shall be canceled by both parties.
4) Step 4. If an *Able To Comply* Acknowledge PDU is sent, any applicable internal state data shall be sent in one or more Set Record-R PDUs immediately following the Acknowledge PDU. If there is no internal state data to be transmitted in the Set Record-R PDU, then the PDU shall be issued with the Number of Record Sets field set to zero.
5) Step 5 (Optional) One or more Record-R PDUs may be optionally sent by the Acquiring Simulation to reflect the content and exact values of the record(s) received from the Divesting Simulation. The records may be placed in the Record-R PDU in any order. A Divesting

Simulation shall not be required to process the Record-R PDU when received as part of a transfer transaction.

6) Step 6. The Acquiring Simulation shall send the *initial* Entity State PDU before the TO_AUTO_RESPONSE_TIMER expires. If there are additional PDUs related to the entity, they shall be sent in accordance with the simulation's requirements for sending such PDUs for a newly initiated local entity. Upon receipt of the *initial* Entity State PDU from the Acquiring Simulation, the Divesting Simulation shall cease transmission of any PDUs for the transferred entity. At this point the Divesting Simulation no longer owns the transferred entity. No *final* Entity State PDU with the Entity Appearance record State field (bit 23) set to Deactivated (1) shall be transmitted by the Divesting Simulation for a transferred entity. The exact process whereby a Divesting Simulation's local entity is transitioned to a remote entity and an Acquiring Simulation's remote entity is transitioned to a local entity is beyond the scope of this standard.

7) Step 7. The Acquiring Simulation shall also send the *ownership* Event Report PDU before the TO_AUTO_RESPONSE_TIMER expires. This PDU shall follow the *initial* Entity State PDU but may be interspersed with other PDUs so long as it is sent before the timer expires.

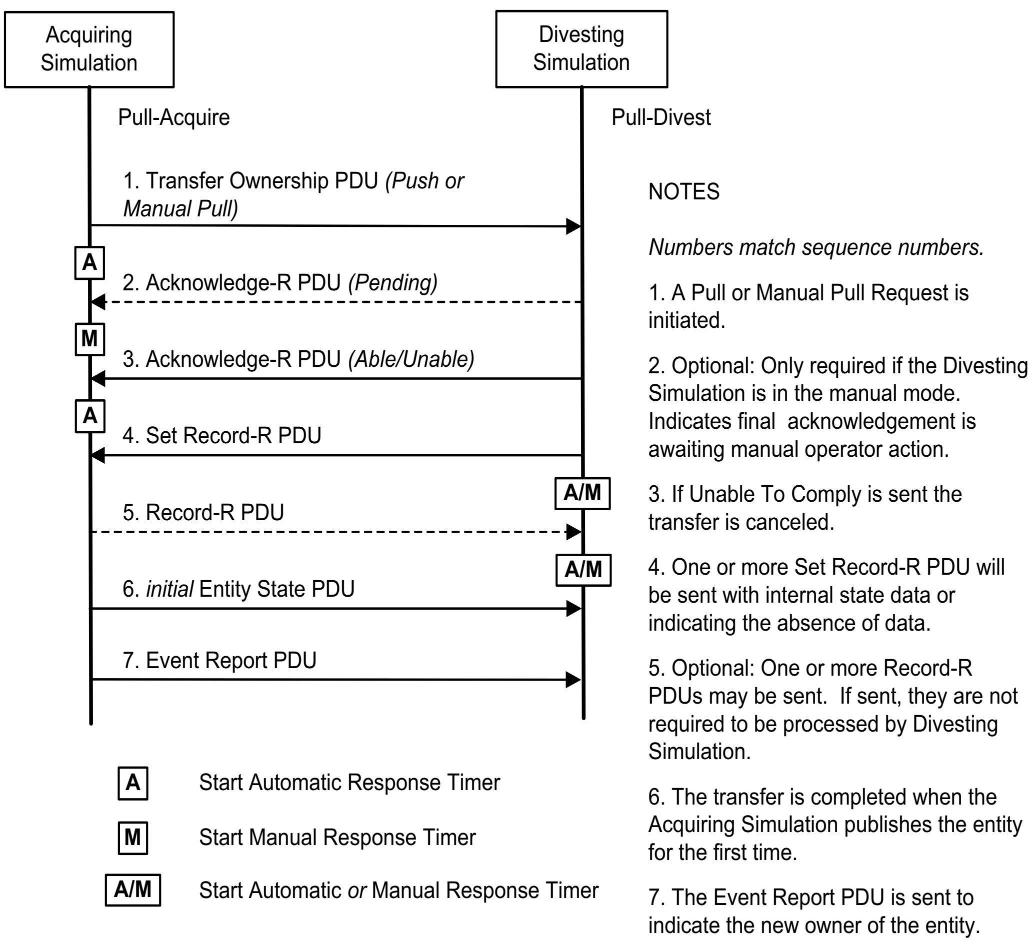


Figure 26—The Pull Transfer

5.9.4.2.4 Transfer ownership timers

The timers associated with the Transfer Ownership function are shown in Table 19.

Table 19—Transfer ownership timers

Symbolic name	Definition	Numeric range[a]	Default value
TO_AUTO_RESPONSE_TIMER	Transfer Ownership Automatic Response Timer This timer shall be set as specified in Figure 25 and Figure 26. If a valid, expected PDU is not received in the allotted time from the other simulation involved in a transfer transaction, the transaction shall be terminated and a *Cancel* TO PDU shall be sent.	(1,1 to 600) s	5 s
TO_MAN_RESPONSE_TIMER	Transfer Ownership Manual Response Timer This timer shall be set as specified in Figure 25 and Figure 26. This timer shall be set (by an Acquiring Simulation or Divesting Simulation) upon receipt of a *Pending Operator Action* Acknowledge-R PDU, or during a pull-divest when the Acquiring Simulation is in the manual transfer mode and the Divesting Simulation is expecting a Record-R PDU or an *initial* Entity State PDU. If a valid, expected PDU is not received in the allotted time from the other simulation involved in a transfer transaction, the transaction shall be terminated and a *Cancel* TO PDU shall be sent.	(1,1 to 600) s	120 s

[a]Numeric range (n1, n2, to n3), where n1 is the increment that the value can be specified in (e.g., 1 means 1 s intervals, 2 means 2 s intervals. If the increment is 2, then the timer value over the range can be 2, 4, 6, etc.). n2 is the lowest possible value, and n3 is the highest possible value.

5.9.4.3 Related PDUs

The detailed requirements related to the PDUs that directly support the transfer ownership function are provided in Annex H.

5.9.4.4 Related records

The records related to the Transfer Ownership function are listed in Table 20. These records shall be used to pass the type of internal state data indicated by each record when required by provisions of 5.9.4 and by an exercise agreement. The record formats and rules are described in 6.2, as shown in Table 20. The General Rules described in 5.9.4.2.1 indicate the conditions under which implementation of specific records are mandatory. Implementation of other records is optional. See [UID 66] for the enumerations used to identify these records in the Record ID field in the TO PDU and the Datum ID field in the Event Report and Data PDUs.

Table 20—Transfer ownership related records

Record name	Subclause
PDU Status record	6.2.67
Munition record	6.2.60
Munition Reload record	6.2.61
Engine Fuel record	6.2.24
Engine Fuel Reload record	6.2.25
Storage Fuel record	6.2.84
Storage Fuel Reload record	6.2.85
Expendable record	6.2.35
Expendable Reload record	6.2.36
Total Record Sets record[a]	6.2.89
Launched Munition record[a]	6.2.50
Association record	6.2.9
Sensor record	6.2.77
Ownership Status record	6.2.65

[a]Mandatory implementation provisions are described in 5.9.4.2.1.

5.9.5 IsPartOf PDU

5.9.5.1 Purpose

The IsPartOf PDU shall be used to request hierarchical linkage of separately hosted simulation entities where the originating simulation application requests that the receiving simulation application honor a respective host/part relationship. An entity that becomes a part of another entity is referred to as a part entity from the simulation time it joins the host entity until it leaves the host entity. When it leaves the host entity, it is again referred to as an entity. The originating entity is also referred to as the host entity.

5.9.5.2 Information contained in the IsPartOf PDU

The IsPartOf PDU shall contain the following information:

a) Standard Simulation Management PDU Header where:
   1) Originating Entity is the entity (host entity) that requests the Receiving Entity to become a part of the host entity
   2) Receiving Entity is the entity requested to become a part of the Originating Entity
b) Relationship between the host entity and part entity
c) Location of the part in the host entity's coordinate system

d) Identification of the part's station with respect to the host entity

e) Attached part's entity type

5.9.5.3 Issuance of the IsPartOf PDU

The IsPartOf PDU shall be issued by a simulation application to request that an entity from a remotely hosted simulation application be treated as a constituent part of the Originating Entity.

The following field specific requirements apply:

a) *Part Location*. The semantics of the part's location data in the Part Location field shall be specified by the Station Name field of the Named Location Identification record in the IsPartOf PDU. When the Station Name enumeration indicates On Station Range and Bearing (15), the three location data fields shall represent the Range (in meters) (Component A), the Bearing (in radians) (Component B), and the third field shall contain zero (Component C). When the Station Name enumeration indicates On Station *xyz* (16), the three location data fields shall represent the *x*, *y*, and *z* coordinates of the part (in meters) (Components A, B, and C, respectively). When the Station Name enumeration is other than On Station Range and Bearing (15) or On Station *xyz* (16), the three fields of the Part Location record shall contain the value zero.

b) There are no additional specific field requirements identified at this time.

The IsPartOf PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.9.5.4 Receipt of the IsPartOf PDU

5.9.5.4.1 General

Upon receipt of an IsPartOf PDU, the receiving simulation application shall make a determination as to the feasibility of honoring the IsPartOf request and shall issue an Acknowledge PDU.

If the requested simulation application is able to comply with the request made in the IsPartOf PDU, the receiving simulation application shall cease issuing Entity State PDUs for its entity (which is now a part of the Originating Entity). The simulation application issuing the IsPartOf PDU shall be responsible for updating the location and orientation information of the part until the part separates from the host entity through deployment, debarkation, or firing events.

Engagement space destruction of the host entity shall imply similar destruction of all of its constituent part entities.

5.9.5.4.2 Munition parts

Upon firing, a part entity that is a munition shall no longer be considered a constituent part of the firing platform. See 5.4.3.2 for the requirements addressing tracked and nontracked fired munitions.

5.9.5.5 Acknowledgment of the IsPartOf PDU

The IsPartOf PDU receiving application shall acknowledge receipt of the IsPartOf PDU by transmitting an Acknowledge PDU (see 5.6.5.6). The Originating Entity ID and Receiving Entity ID fields in the Acknowledge PDU shall be the exact reverse of the corresponding fields in the IsPartOf PDU. When used in the IsPartOf capability of the Entity Management protocol, the Acknowledge PDU shall specify the Entity Management protocol in the Protocol field of the PDU Header.

The Request ID field of the Acknowledge PDU sent in reply to an IsPartOf PDU shall contain the value zero.

5.9.6 Entity separation

5.9.6.1 General

The requirements related to the separation of one entity from another are contained in this subclause. The requirements related to the portrayal of the entity prior to separation are also included. Examples include a pilot life form entity created at the time the pilot would have ejected from a fighter aircraft entity, a landing craft leaving a ship, and the stages of a multistage missile falling away. For accountability, it is important to know the relationships between entities involved in a separation. The term "parent" is used to denote the entity from which another entity separates. Except for the case of a requirement that an *initial* Entity State PDU be issued for a new entity, reference to an Entity State PDU shall also be taken to mean an Entity State Update PDU.

An indication that two entities have physically separated supports higher fidelity representations of separations, as well as data analysis and other functions.

When requirements are contained elsewhere in this standard related to specific instances of the physical separation of entities (e.g., submunitions and multistage missiles), those requirements shall apply and take precedence over the requirements contained in this subclause. Otherwise, if it is desired to communicate that a physical separation has occurred between two entities, the requirements specified herein shall be applicable.

Both the Separation VP record (6.2.94.6) and the Entity Association VP record (6.2.94.4) may indicate the physical separation of two entities. In the latter case, the physical association must be indicated to have existed prior to the separation by the presence of one or more Entity Association VP records in the Entity State PDUs being issued for both entities. Once an association between two entities is established using the Entity Association VP record, that specific association shall be included in all subsequent issuances of the Entity State PDUs for the two associated entities until the specific physical association is broken as indicated by a *final* Entity Association VP record for a specific physical association.

NOTE—The Entity Association VP record is also used to indicate functional associations between entities, or entities and objects, and also the physical association of an entity with a nonentity object such as a point object. Requirements related to those capabilities are contained in 6.2.94.4.

There are two types of separation events. These are referred to as sequential and nonsequential separations:

a) *Sequential Separation*. The separation of one entity from another in a fixed, sequential pattern such as a multistage missile where each stages separates in a fixed sequence.
b) *Nonsequential Separation*. The separation of one entity from another in a nonsequential pattern such as munitions launched from a fighter aircraft.

The Entity Type field of a parent Entity State PDU shall not be changed to reflect a change in the physical appearance of the parent entity due to a separation event.

The term "component" as used in this subclause shall mean any part of an entity that may separate in the normal course of the entity's life. For a multistage missile, this includes the stages, as well as the internal components that will separate such as multiple warheads and decoys.

5.9.6.2 Component representation

The following requirements shall apply to a parent entity for which a new entity is created that represents a separation of a part of the parent entity. These requirements are to support visual and sensor model detection of a parent entity whose physical appearance will change once a component separates from it such that the Entity Type record of the Entity State PDU will no longer reflect the entity's true physical appearance:

a) *Sequential Separations*. If a parent entity part that will eventually separate as a new entity was externally visible prior to the separation, Attached Part VP record(s) will be included in the parent Entity State PDU to represent each part prior to its separation if required as specified in item a3) in A.2.2.3.3.

b) *Nonsequential Separations*. If a parent entity part that was externally visible prior to the separation and that may nonsequentially separate as a new entity, an Attached Part VP record may be included in the parent Entity State PDU to represent the part prior to its separation. Examples include munitions, such as air-to-air and air-to-ground missiles and external fuel tanks on fighter aircraft. High-fidelity flight simulators may represent them as Attached Part VP records prior to their release. Constructive simulations typically do not have that capability.

c) *Association with a Launcher Entity*. If an object is associated with a launcher entity prior to being launched, the following requirements shall apply:

   1) If the weapon is represented as an attached part of the launcher, the launcher entity shall include an Attached Part VP record to represent the missile. When the object is fired or launched, the appropriate processing as specified in item b) in 5.9.6.3 shall apply.

   2) If the object is represented as an entity on the launcher for which Entity State PDUs are being issued, the launcher entity shall not include an Attached Part VP record to represent the missile. However, when the object is fired or launched, the Fire PDU shall indicate the launcher as the Firing Entity ID.

5.9.6.3 Nonweapon separation

When a nonweapon entity physically separates from another entity, and it is desired to communicate that a separation has occurred, the following rules shall apply:

a) A Separation VP record may be included once in the Entity State PDU for the separating entity at the time it is determined that the entity has physically separated from a parent entity. If the two entities had one or more physical connections between them as indicated by the presence of Entity Association VP record(s), all such physical associations shall be broken either before, or at the time, the Separation VP record is sent.

b) If the separated entity was previously represented as an Attached Part VP record in the parent entity, an Entity State PDU update shall be issued for the parent entity when separation occurs with the Detached Indicator set to Detached (1) in the applicable Attached Part VP record. The applicable Attached Part VP record shall be removed from the parent entity in the next Entity State PDU that is issued for the parent in the normal course of events.

c) If the nonweapon entity is a multicomponent manned spacecraft or unmanned rocket such as those that are launched by National Aeronautics and Space Administration (NASA), applicable requirements for multistage missiles shall apply as specified in Annex A.

5.9.6.4 Weapon separation

When a weapon entity initially separates from a parent (launch platform) entity, the requirements specified in 5.4 and Annex A shall apply.

5.10 Minefield

5.10.1 General

The protocol for communicating information associated with the location, appearance, and other pertinent details of mines and minefields is described in 5.10.2 through 5.10.5. The Minefield protocol is composed of the following four PDUs:

a) Minefield State PDU
b) Minefield Query PDU
c) Minefield Data PDU
d) Minefield Response NACK PDU

These PDUs can be used in two protocol modes. One mode called the Minefield Query Response Protocol (QRP) mode uses all four PDUs. The second mode, heartbeat mode, uses only the Minefield State PDU and the Minefield Data PDU. The protocol mode is chosen by the minefield simulation. This choice should be determined prior to the start of an exercise.

The Minefield State PDU provides information defining an aggregate minefield object that consists of a set of individual mines. A Minefield Data PDU is utilized to describe deaggregated mines contained within a minefield.

The QRP mode provides an alternative to the heartbeat-based transmission of the Minefield Data PDU. If the QRP mode is selected, the Minefield Data PDU shall only be transmitted in response to a Minefield Query PDU. The Minefield Query PDU is sent to request information about deaggregated mines within a region. The Minefield Response NACK PDU is used to request retransmission of single Minefield Data PDUs that are not received by the querying simulation. QRP mode (i.e., the use of the Minefield Query and Minefield Response NACK PDUs along with the Minefield State and Minefield Data PDUs) is recommended for DIS exercises with large quantities of mines.

5.10.2 Minefield State PDU

5.10.2.1 Purpose

The Minefield State PDU shall communicate information about the minefield, including the location, perimeter, and types of mines contained within it.

5.10.2.2 Information contained in the Minefield State PDU

The Minefield State PDU shall contain the following information:

a) Identification of the minefield issuing the PDU
b) A minefield sequence number
c) Identification of the force to which the minefield belongs
d) Number of perimeter points defining the extent of the minefield
e) Minefield type
f) Number of types of mines in the minefield
g) Location of the minefield in the simulated world
h) Orientation of the minefield in the simulated world
i) Appearance information for displaying the symbology of the minefield as a doctrinal minefield graphic
j) Protocol mode that the minefield simulation is using to communicate the minefield data

k) Coordinates of the perimeter points

l) Identification of the types of mines in the minefield

5.10.2.3 Issuance of the Minefield State PDU

A minefield simulation shall issue a Minefield State PDU when any of the following occur:

a) A change has occurred in the force ID, location, orientation, perimeter, mine types, or appearance of the minefield.

b) A change has occurred in the state of a mine in the minefield and a Minefield State PDU has not been sent since the last Minefield Query PDU for this minefield was answered by the minefield simulation. If a Minefield State PDU has been issued since the last Minefield Query PDU was satisfied, a Minefield State PDU shall not be sent when the state of a mine in a minefield changes [unless the state change causes a corresponding change in the minefield covered under item a) above].

c) A predetermined length of real-world time has elapsed since the issuing of the last Minefield State PDU. The Minefield State PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_MINEFIELD_STATE. (See 6.1.8 for parameter details and default values.)

The following field-specific requirements apply:

— *Minefield Sequence Number*. This number shall start with one and shall be incremented sequentially when the state of any of the mines contained within the minefield change, or when any of the following fields of the Minefield State PDU change: Force ID, Number of Perimeter Points, Minefield Type, Number of Mine Types, Minefield Location, Minefield Orientation, Appearance, Protocol Mode, Perimeter Point Coordinates, or Mine Type. The Minefield Sequence Number is limited to a maximum value of 32 767. When all possible values are exhausted, the numbers shall be reused beginning at one.

— There are no additional specific field requirements identified at this time.

The Minefield State PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.10.2.4 Receipt of the Minefield State PDU

Upon receipt of a Minefield State PDU, the receiving simulation application operating in QRP mode shall examine the sequence number to determine whether any of the information regarding the minefield has changed. If the sequence number has incremented since receipt of the previous Minefield State PDU, then the new information about the minefield shall be used by the simulation. Otherwise, the PDU shall be discarded. Upon receipt of a Minefield State PDU, the receiving simulation application operating in heartbeat mode shall use the new information about the minefield.

If the appearance information of the minefield indicates that the minefield is in the Deactivated state or the countdown timer for the minefield expires, then all simulations shall remove the minefield from the exercise. This reference time interval shall be established at exercise start, although it may be changed during the exercise. If no value for this time interval is established, then the default value for this time shall be the value of the Minefield State PDU issuance interval [item c) in 5.10.2.3] multiplied by HBT_TIMEOUT_MPLIER.

5.10.3 Minefield Query PDU

5.10.3.1 Purpose

The Minefield Query PDU shall contain information about the requesting entity and the region and mine types of interest to the requesting entity. The Minefield Query PDU shall only be used if the QRP mode is selected.

5.10.3.2 Information contained in the Minefield Query PDU

The Minefield Query PDU shall contain the following information:

a) Identification of the minefield
b) Identification of the requesting entity
c) Request identification of the query
d) Number of perimeter points for the requested area
e) Number of requesting sensors
f) Data filter
g) Request mine type filter
h) Coordinates of the perimeter points of the requested area
i) Sensor types

5.10.3.3 Issuance of the Minefield Query PDU

The Minefield Query PDU shall be issued to request additional mine information from the minefield simulation when the following conditions exist:

a) Use of the Minefield Query PDU (i.e., QRP mode) is selected.
b) Based on a Minefield State PDU, an object (mine type) of interest exists.
c) A minefield is within or near the sensor's range of the vehicle.

The following field-specific requirements apply:

— *Number of Sensor Types*. A zero in this field shall indicate that the requesting simulation is not a detection system.
— *Requested Mine Type Filter*. If a subfield within the Requested Mine Type Filter record (e.g., the country field) contains the value –1 (all bits set), then all subfield values shall match the filter (e.g., all countries are allowed). If a subfield of the Requested Mine Type Filter record contains a value other than –1, a mine type shall match this value to match the filter.

The Minefield Query PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.10.3.4 Receipt of the Minefield Query PDU

Upon receipt of a Minefield Query PDU, a minefield simulation shall utilize the information in the Minefield Query PDU to provide a tailored response with only the requested information provided. In particular, the minefield simulation shall use the following criteria to determine which mine(s) are sent in a Minefield Data PDU:

a) Is the mine of the requested mine type?
b) Is the mine within the perimeter of the requested area?

A Minefield Data PDU shall always be sent even if no mines match the query. If the resulting Minefield Data PDU exceeds the allowable PDU size limit (see IEEE Std 1278.2), the mine information shall be distributed in multiple Minefield Data PDUs.

5.10.4 Minefield Data PDU

5.10.4.1 Purpose

Information about the location and status of a collection of mines in a minefield is conveyed through the Minefield Data PDU on an individual mine basis.

Only mine entities originated by the simulation that is issuing the Minefield State PDU are eligible to be contained in the Minefield Data PDU due to the dependency on a master Site Number and Application Number for all mine entities. This means that a local mine entity with a different Site Number and Application Number from that of the Minefield Identifier Site Number and Application Number is not eligible to be included in the Minefield Data PDU. This condition is normally the result of an ownership transfer.

If the QRP mode is selected, the Minefield Data PDU shall supply only mine data that satisfy conditions of the requesting sensor types and the mine type and geographic filters provided in the Minefield Query PDU. If heartbeat mode is selected, the Minefield Data PDU shall supply mine data for all the mines in the minefield or for mines that have changed (see 5.10.4.3.2).

5.10.4.2 Information contained in the Minefield Data PDU

The Minefield Data PDU shall contain the following information:

a) Identification of the minefield
b) Identification of the requesting entity
c) Minefield sequence number
d) Request identification associated with the query being processed
e) PDU sequence number
f) Number of PDUs in response set
g) Number of mines
h) Number of requesting sensors
i) Data filter that indicates which of the optional data are present for the mine(s) (see 6.2.16)
j) Type of mine contained in the PDU
k) Requesting sensor type(s)
l) Location of the mine(s)
m) Mine Entity Identifier

5.10.4.3 Issuance of the Minefield Data PDU

5.10.4.3.1 General

The Minefield Data PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

The conditions for issuance of the Minefield Data PDU vary depending on the selection of the protocol mode.

5.10.4.3.2 Heartbeat mode issuance criteria

The minefield simulation shall issue a Minefield Data PDU under the following conditions when the heartbeat mode is selected:

a) A change has occurred in the minefield ID associated with a mine; whether a Minefield data PDU is issued immediately to reflect the change or whether the change is included in the next periodic Minefield Data PDU heartbeat depends on the value of the Minefield Data Change parameter [see item d) below]. The Minefield Data PDU change parameter shall be identified by the symbolic name HBT_PDU_MINEFIELD_DATA. (See 6.1.8 for parameter details and default values.)

b) A change has occurred in the state of a mine.

c) A predetermined length of real-world time has elapsed since the issuing of the last Minefield Data PDU describing a specific mine. The Minefield Data PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_MINEFIELD_DATA. (See 6.1.8 for parameter details and default values.)

d) The Minefield Data PDU change parameter shall be identified by the symbolic name MINEFIELD_CHANGE. The MINEFIELD_CHANGE value shall be a number between zero and up to and including the HBT_PDU_MINEFIELD_DATA value. (See 6.1.8 for parameter details and default values in addition to those specified herein.). The sending of a change based on conditions specified in item a) and item b) above shall be determined as follows:

   1) If the MINEFIELD_CHANGE value is set to zero, the Minefield Data PDU is sent immediately with the change specified in item a) and item b).

   2) If the MINEFIELD_CHANGE value is set to the HBT_PDU_MINEFIELD_DATA value, the change specified in item a) and item b) is only sent when the next heartbeat Minefield Data PDU is sent.

   3) If the MINEFIELD_CHANGE value is set to a number between zero and the HBT_PDU_MINEFIELD_DATA value, it represents a predetermined length of real-world time that has elapsed since the last Minefield Data PDU would have been sent for a change if any changes existed to be sent. In this case, any changes specified in item a) and item b) shall be sent together between heartbeats at the indicated rate of MINEFIELD_CHANGE.

A Minefield Data PDU shall contain mine information for only a single type of mine. A Minefield Data PDU may contain data for a single mine or multiple mines. The minefield simulation should make a best effort to combine data for multiple mines into a single Minefield Data PDU in order to reduce network loading. The PDU sequence number is not used when operating in the heartbeat mode.

5.10.4.3.3 QRP mode issuance criteria

A minefield simulation shall issue a Minefield Data PDU when operating in the QRP mode only in response to the receipt of a Minefield Query PDU.

A Minefield Data PDU shall contain mine information for a single type of mine. Multiple Minefield Data PDUs shall be issued if multiple mine types match a query or if the mine information for a single type of mine does not fit in a single PDU. The set of Minefield Data PDUs necessary to contain all of the requested mine information for the requested mine type and within the requested perimeter shall be sent in response to a Minefield Query PDU. The Number of PDUs field shall indicate how many Minefield Data PDUs are being sent in the response set of Minefield Data PDUs. The PDU sequence number shall specify an individual Minefield Data PDU within this set.

In QRP mode, the minefield simulation shall retransmit individual Minefield Data PDUs in response to a Minefield Response NACK PDU.

5.10.4.4 Receipt of the Minefield Data PDU

5.10.4.4.1 Receipt of the Minefield Data PDU in heartbeat mode

Upon receipt of a Minefield Data PDU, the receiving simulation application operating in heartbeat mode shall use the new information about the minefield.

If the countdown timer for a given mine expires, then the simulation application shall remove the model of that mine from its simulation. This value and associated tolerance may be established at the start of an exercise or during the exercise by means of Simulation Management PDUs. The Minefield Data PDU timeout value shall be computed as specified in NOTE 2 in Table 25.

5.10.4.4.2 Receipt of the Minefield Data PDU in QRP Mode

Upon receipt of a Minefield Data PDU, the receiving simulation application operating in QRP mode shall examine the minefield sequence number to determine whether any of the information about the mines in the minefield has changed (the sequence number could be incremented for reasons other than changes in individual mines). If the sequence number has incremented since receipt of the previous Minefield Data PDU, then the new information about the minefield shall be used by the simulation. Otherwise, the PDU shall be discarded.

When operating in QRP mode, the receiving simulation application shall remove the model of a mine if the minefield referenced by the Minefield ID is deleted. The simulation application shall internally represent a mine as obsolete if a Minefield Data PDU with the same Minefield ID, the same Mine Type, and a newer sequence number is received that does not contain the mine. This step is due to the fact that a simulation cannot be sure of the existence and state (including location) of a particular mine that it previously acquired, since the Minefield Data PDU shall contain only data for the specified query region. If a Minefield Data PDU is subsequently received that provides current data for the obsolete mine, the old state information shall be replaced with the new state information and the state of the mine updated from obsolete to current. The simulation application shall not utilize obsolete mine data but may maintain previously generated internal state information for obsolete mines indefinitely. A possible use of internal state information is to provide consistent detection probabilities across acquisitions should the mine be updated to current.

5.10.4.5 Minefield data order

The multidimensional fields within the Minefield Data PDU shall be ordered specifically as shown below:

1) Scalar Detection Coefficient—All of the Scalar Detection Coefficient values (per sensor type) for one mine shall be provided before the next mine. There are M values per mine and N mines. Let S_{nm} be the Scalar Detection Coefficient for mine n and sensor type m. The ordering is thus:

   $S_{11}, S_{12}, \ldots, S_{1M}$

   $S_{21}, S_{22}, \ldots, S_{2M}$

   ...

   $S_{N1}, S_{N2}, \ldots, S_{NM}$

2) Number of Vertices—All of the Number of Vertices values (per wire) for one mine shall be provided before the next mine. There are $I(n)$ wires per mine, where n = 1 to N, and N mines. Let U_{nj} be the Number of Vertices for mine n and wire j. The ordering is thus:

   $U_{11}, U_{12}, \ldots, U_{1I(n)}$

   $U_{21}, U_{22}, \ldots, U_{2I(n)}$

...

$U_{N1}, U_{N2}, \ldots, U_{NI(n)}$

3) Vertex—All of the Vertex values for one wire shall be provided before the next wire. All of the Vertex values for the wires of one mine shall be provided before the next mine. There are *I(n)* wires per mine, where *n* = 1 to *N*, *J(i)* vertices per wire, where *i* = 1 to *I(n)*, and *N* mines. Let V_{nji} be the Vertex for mine *n* and wire *j* and vertex *i*. The ordering is thus:

$V_{111}, V_{112}, \ldots, V_{11J(i)}$

$V_{121}, V_{122}, \ldots, V_{12J(i)}$

...

$V_{1I(n)1}, V_{1I(n)2}, \ldots, V_{1I(n)J(i)}$

...

$V_{NI(n)1}, V_{NI(n)2}, \ldots, V_{1I(n)J(i)}$

5.10.5 Minefield Response Negative Acknowledgment (NACK) PDU

5.10.5.1 Purpose

The Minefield Response NACK PDU shall contain information about the requesting entity and the PDU(s) that were not received in response to a query. The Minefield Response NACK PDU shall be used only if the QRP mode is selected.

5.10.5.2 Information contained in the Minefield Response NACK PDU

The Minefield Response NACK PDU shall contain the following information:

a) Identification of the minefield

b) Identification of the requesting entity

c) Request identification of the corresponding query

d) Number of missing PDU(s)

e) Missing PDU sequence number(s)

5.10.5.3 Issuance of the Minefield Response NACK PDU

The Minefield Response NACK PDU shall be issued to request Minefield Data PDU(s) that were not received by the requesting simulation in response to a Minefield Query PDU. The Minefield Response NACK PDU shall be issued after a predetermined length of real-world time has elapsed since the receipt of a Minefield Data PDU in response to a query if the number of PDU(s) indicated have not all been received. The Minefield response NACK PDU timer parameter and tolerance shall be identified by the symbolic name MINEFIELD_RESPONSE_TIMER. (See 6.1.8 for parameter details and default values.)

The Minefield Response NACK PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.10.5.4 Receipt of the Minefield Response NACK PDU

Upon receipt of a Minefield Response NACK PDU, a minefield simulation operating in QRP mode shall utilize the information in the Minefield Response NACK PDU to retransmit the requested PDU(s).

5.11 Synthetic Environment

5.11.1 General

The Synthetic Environment protocol supports the simulation of environmental conditions that change during the life of a distributed simulation exercise. This protocol contains the following PDUs:

a) *Environmental Process PDU*. Communicates simple environment variables, small-scale environmental updates, and embedded processes.

b) *Gridded Data PDU*. Transmits information about large-scale or high-fidelity spatially and temporally varying ambient fields and about environmental processes and features.

c) *Point Object State PDU*. Transmits specific subset of information about environment objects described by a point.

d) *Linear Object State PDU*. Transmits specific subset of information about environment objects described by lines.

e) *Areal Object State PDU*. Transmits specific subset of information about environment objects described by areas.

The Synthetic Environment protocol provides two approaches for the simulation of environmental conditions. One approach is a process approach, and the second is an object approach. These approaches and the associated protocols are described in 5.11.2 and 5.11.3, respectively.

5.11.2 Process approach to synthetic environment simulations

5.11.2.1 General

The following rules shall be followed when utilizing the process approach to the Synthetic Environment protocol in DIS:

a) The application responsible for simulating a particular environmental condition or conditions shall output Environmental Process and Gridded Data PDUs to represent the current state and geometry of the environmental condition or conditions.

b) Receiving applications that are affected by the environmental condition(s) shall use the parametric physical data in the Environmental Process and Gridded Data PDUs to drive the appropriate effects models within their simulations.

c) An application shall issue Environmental Process and Gridded Data PDUs at process instantiation, process termination, and periodically. The period between updates shall depend on simulation models and shall be established at the start of an exercise for each model. This period may be established by means of Simulation Management PDUs.

5.11.2.2 Environmental Process PDU

5.11.2.2.1 Purpose

The Environmental Process PDU shall communicate information about the environment, including simple environment variables, small-scale environmental updates, and embedded processes. An embedded process is an environmental effect instigated by the action of an entity that may continue after that entity has left the battlespace. Examples include contrails, smoke, obscurants, chaff, dust clouds, and toxic chemicals. The environmental information communicated shall be sufficient for receiving applications to represent the issued environment in the application's own simulation.

5.11.2.2.2 Information contained in the Environmental Process PDU

The Environmental Process PDU shall contain the following information:

a) Identification of the environmental process that is responsible for the environmental condition.
b) Identification of the type of environmental condition.
c) Identification of the environmental model being used.
d) Status of the environmental condition.
e) Number of environmental records.
f) Identification of the PDU sequence number.
g) Information that describes the state of the environmental condition, i.e., source, intensity, speed, etc. The following optional state information may also be included:
    1) Extrapolation method to be used for the state information.
    2) Associated Entity ID that identifies an entity that may be related to or causing the environmental condition.
h) Information that describes the geometry of the environmental condition(s). Optional geometry information includes the extrapolation method to be used for the geometry information. When more than one geometry record are included in the PDU, dimensions of the sphere that enclose the subsequent geometries shall be included as the first Environment record in the PDU.

5.11.2.2.3 Dead reckoning

Both the state and the geometry of an environmental process shall employ dead reckoning algorithms to limit the frequency at which Environmental Process PDUs are issued. Each simulation application shall maintain two models of each process it is representing. One model shall be an internal model of the process. The other model shall be a dead reckoned model of the process. Certain thresholds shall be established as criteria for determining whether the process's actual state and geometry have varied by an allowable amount from the dead reckoned values. When any of the process's actual state or geometry values have deviated from the dead reckoned state or geometry values by more than a threshold value, the simulation application shall issue an Environmental Process PDU to communicate the process's actual state and geometry values to other simulation applications. The simulation application shall also use the same information communicated to other simulation applications to update its dead reckoning model of the process.

The default method of a dead reckoning error calculation shall be based on threshold changes in geometry or state values. The thresholds for geometries shall be as follows:

a) Changes greater than the environmental process dimension threshold in any dimension. The environmental process dimension threshold shall be identified by the symbolic name EP_DIMENSION_THRSH. (See 6.1.8 for parameter details and default values.)
b) Changes greater than the environmental process location threshold. The environmental process location threshold shall be identified by the symbolic name EP_POS_THRSH. (See 6.1.8 for parameter details and default values.)
c) Changes greater than the environmental process state threshold difference in the actual versus expected state values. The environmental process state threshold shall be identified by the symbolic name EP_STATE_THRSH. (See 6.1.8 for parameter details and default values.)

5.11.2.2.4 Issuance of the Environmental Process PDU

The Environmental Process PDU shall be issued by the simulation application responsible for the environmental process when one or more of the following conditions exist:

a) A predetermined length of real-world time has elapsed since the issuance of the last Environmental Process PDU. The Environmental Process PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_ENVIRONMENTAL_PROCESS. (See 6.1.8 for parameter details and default values.) This periodic transmission is not necessary when a process is in the quiescent state. An environmental process is in the quiescent state when the environmental effect managed by the process is quiescent. The Exercise Manager shall provide a mechanism to allow latecomers to receive the information contained in the Environmental Process PDU if the quiescent state capability is used during an exercise.

b) Discrepancy between the process's actual state and geometry values (as determined by its own internal model) and its dead reckoned model state and geometry values (using specified dead reckoning algorithms) exceeds a predetermined threshold (see 5.11.2.2.3 concerning threshold values).

c) Change in the status of the process occurs.

The Environmental Process PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

The following field specific requirements apply:

— *Sequence Number*. When PDU sequencing is required, the Sequence Number field shall be set to zero for each exercise and sequentially incremented by one when a PDU is issued by an environmental process. When all possible values are exhausted, the numbers shall be reused beginning at one. If PDU sequencing is not required, then this field shall contain EP_NO_SEQUENCE.

— There are no additional specific field requirements identified at this time.

5.11.2.2.5 Receipt of the Environmental Process PDU

Upon receipt of an Environmental Process PDU, a simulation application shall determine whether the PDU contains newer information than that currently being used to model the conditions described by the received PDU. If so, the simulation application shall use the information contained therein to model the conditions described in the PDU. Otherwise, the PDU shall be discarded.

If the countdown timer for a given process expires and the Environment Status field of the last Environmental Process PDU did not indicate the process is in a quiescent state, then all simulation applications shall remove that process instance from the simulation. The Environmental Process object timeout parameter shall be calculated in accordance with the requirements set out in 4.2.7.2. If the Environmental Status field of the Environmental Process PDU indicates that the current Environmental Process PDU is the last PDU for the process, the instance of the process shall be deleted from the simulation. If an Environmental Process PDU is received with the Environmental Status field indicating that the process is in the quiescent state, then all simulation applications shall maintain that process instance in the state described by the information contained within the PDU. Simulation applications should not dead reckon environmental processes in the quiescent state; the simulation applications should maintain the process until a subsequent Environmental Process PDU is received indicating that the process is no longer in the quiescent state. This status allows for a change in the terrain without needing to transmit this change for the duration of the exercise. The Exercise Manager shall provide a mechanism to allow for latecomers to learn about changes to the quiescent environmental conditions when the quiescent state capability is used.

5.11.2.2.6 Initiation and hand-off of environmental processes

Any application may initiate an environmental process by assigning a new Entity ID to the process and sending an Environmental Process PDU with the Is Active field (bit 1) of the Environment Status record set to Active (1).

The hand-off of an environmental process shall be accomplished using the Transfer Ownership capability of the Entity Management protocol (see 5.9.4).

5.11.2.3 Gridded Data PDU

5.11.2.3.1 Purpose

The Gridded Data PDU shall transmit information about large-scale or high-fidelity spatially and temporally varying ambient fields and about environmental features and processes. Multiple PDUs may be required to transfer all data necessary to describe an environmental state variable. The environmental state information transmitted shall be sufficient for receiving applications to represent the issued environment within the application's own simulation.

5.11.2.3.2 Information contained in the Gridded Data PDU

The Gridded Data PDU shall contain the following information:

a) Identification of the environmental simulation application distributing the environmental state variables
b) Field number for each environmental variable
c) PDU index number within the total set of PDUs
d) Number of Gridded Data PDUs required to contain the environmental data
e) Identification of the coordinate system of the environmental state data
f) Number of grid axes
g) Identification of whether the domain grid axes have changed
h) Identification of the type of environmental state data
i) Information on the angular orientation of the grid
j) Identification of the simulation time at which the environmental state data are to be applied to the simulation
k) Identification of the total number of data values for the environmental sample
l) Identification of the number of data values stored at each grid location
m) Identification of the dimensionality of the grid
n) Environmental state data at the specified grid location

5.11.2.3.3 Dead reckoning

Dead reckoning algorithms shall not be employed to extrapolate the global environmental data contained in the Gridded Data PDU. The current environmental state shall be used until an update of the environmental state variable is received. The frequency of the update process is governed by the availability of environmental data. An interleaf sampling method is provided so that data voids can be filled through standard interpolation methods.

5.11.2.3.4 Grid axis and point order

In any Gridded Data PDU, the Grid Axis Descriptor records shall appear in the order specified by the Coordinate System field. For instance, if the Coordinate System is specified as a Right-Hand Cartesian system (East, North, Up), then the first Grid Axis Descriptor record in the Gridded Data PDU is for the East axis, followed by the Grid Axis Descriptor record for the North axis, and finally the Grid Axis Descriptor record for the Up axis.

The Grid Data records shall appear from lowest index to the highest index in the following order: first the vector dimension, then the first axis that appears in the Gridded Data PDU, then the second axis, continuing until the last axis. For example, using a three-dimensional grid with a vector dimension of three, representing the *u*, *v,* and *w* wind components, let $G_{x\ i\ j\ k}$ represent the Grid Data record for the Vector component *x* at the *i*th element along the first axis, the *j*th element along the second axis, and the *k*th element along the third axis. Then the order of Grid Data records is:

$$G_{u111}\ G_{v111}\ G_{w111}\ G_{u211}\ G_{v211}\ \dots\ G_{wN_1 11}$$
$$G_{u121}\ G_{v121}\ G_{w121}\ G_{u221}\ G_{v221}\ \dots\ G_{wN_1 21}$$
$$\dots$$
$$G_{u1N_2 1}\ G_{v1N_2 1}\ G_{w1N_2 1}\ G_{u2N_2 1}\ G_{v2N_2 1}\ \dots\ G_{wN_1 N_2 1}$$
$$G_{u112}\ G_{v112}\ G_{w112}\ G_{u212}\ G_{v212}\ \dots\ G_{wN_1 12}$$
$$\dots$$
$$G_{u1N_2N_3}\ G_{v1N_2N_3}\ G_{w1N_2N_3}\ G_{u2N_2N_3}\ G_{v2N_2N_3}\ \dots\ G_{wN_1N_2N_3}$$

where N_i is the number of points in the *i*th axis.

5.11.2.3.5 Issuance of the Gridded Data PDU

The Gridded Data PDU shall be issued by the environmental simulation application responsible for the environmental state variables when one or more of the following conditions exist:

a) A predetermined length of real-world time has elapsed since the issuance of the last Gridded Data PDU. The Environmental Process PDU heartbeat timer parameter and tolerance shall be identified by the symbolic name HBT_PDU_GRIDDED_DATA. (See 6.1.8 for parameter details and default values.)

b) Discrepancy between the actual state (GD_STATE_CHANGE) and geometry (GD_GEOMETRY_CHANGE) values (as determined by its own internal model), and its dead reckoned model state and geometry values (using specified standard interpolation algorithms) exceed a predetermined threshold. (See 6.1.8 for parameter details and default values.)

The Gridded Data PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.11.2.3.6 Receipt of the Gridded Data PDU

Upon receipt of the Gridded Data PDU, a simulation application shall determine the type of environmental state information and valid simulation time for the data contained therein. If the valid simulation time of the data is greater than that of previously received state information and the current simulation time is greater than or equal to the valid simulation time of the data, the environmental state variable is recovered from the received PDU(s) and used by the application.

5.11.3 Object approach to synthetic environment simulations

5.11.3.1 General

An alternative approach for distributing environmental data is to use the environment object PDUs. Point, linear, and areal objects that are attached to the terrain are addressed by a specific class of Synthetic Environment PDUs. The Point, Linear, and Areal Object State PDUs were developed for a specific architecture that requires three applications (one to send, one to receive, and one to manage) to be involved in changes to the environment. The process-based approach (see 5.11.2) can be accomplished with only two applications (one to send and one to receive).

5.11.3.2 Point Object State PDU

5.11.3.2.1 Purpose

The Point Object State PDU shall communicate the addition/modification of a synthetic environment object that is geometrically anchored to the terrain with a single point.

5.11.3.2.2 Information contained in the Point Object State PDU

The Point Object State PDU shall contain the following information:

a) Identification of the synthetic environment object.

b) Identification of the synthetic environment object with which this point object is associated or NO_SPECIFIC_ENTITY if the point object is not associated with any other synthetic environment object.

   NOTE—In most cases, the point object will not be associated with any other synthetic environment object.

c) Unique update number to identify each state transition of an individual object.

d) Identification of the force that created/modified the object.

e) Indication of whether modifications have been made to location and orientation of the point object.

f) Definition of the type of synthetic environment object.

g) Location in world coordinates of the synthetic environment object. This location corresponds to the origin of the point object model.

h) Orientation of the synthetic environment object, where the origin of the object is at the Object Location.

i) Information required to represent the appearance attributes of the point object.

j) Identification of the simulation application that is sending or has sent the Point Object State PDU to the Environment Manager.

k) Identification of the simulation application that is to receive the Point Object State PDU. This field shall be set to the Environment Manager by an application that is sending a Point Object State PDU to the Environment Manager. The Environment Manager shall set this field to ALL_SITES:ALL_APPLIC when sending Point Object State PDUs to all applications in the exercise.

5.11.3.2.3 Issuance of the Point Object State PDU

A Point Object State PDU shall be issued by:

a) A simulation application to the Environment Manager to create, modify, or delete a point object

b) The Environment Manager to transmit the effect of the requested point object creation, modification, or deletion on the environment

The Point Object State PDU shall be issued by a simulation application using a unicast communication service when transmitting to the Environment Manager. The Environment Manager shall issue the Point Object State PDU using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.11.3.2.4 Receipt of the Point Object State PDU

Upon receipt of the Point Object State PDU from a simulation application, the Environment Manager shall determine the effect the creation, modification, or deletion of the object has on the environment. Based on this determination, the Environment Manager shall issue the Point Object State PDU to all other simulation applications in the exercise. The Environment Manager shall be able to modify the object based on the environment before transmittal to all simulation applications in the exercise.

Upon receipt of the Point Object State PDU from the Environment Manager, the simulation application shall update the local representation of the environment.

5.11.3.3 Linear Object State PDU

5.11.3.3.1 Purpose

The Linear Object State PDU shall communicate the addition/modification of a synthetic environment object that is geometrically anchored to the terrain with one point and has a segment size and orientation.

5.11.3.3.2 Information contained in the Linear Object State PDU

The Linear Object State PDU shall contain the following information:

a) Identification of the synthetic environment object in an exercise.
b) Identification of the synthetic environment object with which this linear object is associated or NO_SPECIFIC_ENTITY if the linear object is not associated with any other synthetic environment object.

   NOTE—In most cases, the linear object will not be associated with any other synthetic environment object.

c) Unique update number to identify each state transition of an individual object.
d) Identification of the force that created/modified the object.
e) Identification of the number of segment records in the linear object.
f) Identification of the simulation application that is sending or has sent the Linear Object State PDU to the Environment Manager.
g) Identification of the simulation application that is to receive the Linear Object State PDU. This field shall be set to the Environment Manager by an application that is sending a Linear Object State PDU to the Environment Manager. The Environment Manager shall set this field to ALL_SITES:ALL_APPLIC when sending Linear Object State PDUs to all applications in the exercise.
h) Definition of the type of synthetic environment object.
i) Identification of the individual segment number of the linear segment.
j) Identification of whether modifications have been made to the location and orientation of the linear segment.
k) Information required to represent the appearance attributes of the linear segment.
l) Location in world coordinates of the linear segment. See 1.6.3.4 for a description of the linear segment coordinate system.
m) Orientation of the linear segment about the segment location.
n) Length of the linear segment, where the length extends in the positive *x* direction.

o) Width of the linear segment, where 1/2 width extends in the positive *y* direction and 1/2 width extends in the negative *y* direction.

p) Height of the linear segment above ground level.

q) Depth of the linear segment below ground level.

5.11.3.3.3 Issuance of the Linear Object State PDU

A Linear Object State PDU shall be issued by:

a) A simulation application to the Environment Manager to create, modify, or delete a linear object

b) The Environment Manager to transmit the effect of the requested linear object creation, modification, or deletion on the environment

The Linear Object State PDU shall be issued by a simulation application using a unicast communication service when transmitting to the Environment Manager. The Environment Manager shall issue the Linear Object State PDU using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.11.3.3.4 Receipt of the Linear Object State PDU

Upon receipt of the Linear Object State PDU from a simulation application, the Environment Manager shall determine the effect that the creation, modification, or deletion of the object has on the environment. Based on this determination, the Environment Manager shall issue the Linear Object State PDU to all other simulation applications in the exercise. The Environment Manager shall be able to modify the object based on the environment before transmittal to all simulation applications in the exercise.

Upon receipt of the Linear Object State PDU from the Environment Manager, the simulation application shall update the local representation of the environment.

5.11.3.4 Areal Object State PDU

5.11.3.4.1 Purpose

The Areal Object State PDU shall communicate the addition/modification of a synthetic environment object that is geometrically anchored to the terrain with a set of points (at least three points that come to a closure).

5.11.3.4.2 Information contained in the Areal Object State PDU

The Areal Object State PDU shall contain the following information:

a) Identification of the synthetic environment object.

b) Identification of the synthetic environment object with which this areal object is associated or NO_SPECIFIC_ENTITY if the areal object is not associated with any other synthetic environment object.

   NOTE—In most cases, the areal object will not be associated with any other synthetic environment object.

c) Unique update number to identify each state transition of an individual object.

d) Identification of the force that created/modified the object.

e) Identification of whether modifications have been made to any locations of the areal object.

f) Definition of the type of synthetic environment object.

g) Information required to represent the appearance attributes of the areal object.

h) Identification of the total number of points making up the areal object.

i) Identification of the simulation application that is sending or has sent the Areal Object State PDU to the Environment Manager.

j) Identification of the simulation application that is to receive the Areal Object State PDU. This field shall be set to the Environment Manager by an application that is sending an Areal Object State PDU to the Environment Manager. The Environment Manager shall set this field to ALL_SITES:ALL_APPLIC when sending Areal Object State PDUs to all applications in the exercise.

k) Location in world coordinates of each point making up the areal object.

5.11.3.4.3 Issuance of the Areal Object State PDU

An Areal Object State PDU shall be issued by:

a) A simulation application to the Environment Manager to create, modify, or delete an areal object

b) The Environment Manager to transmit the effect of the requested areal object creation, modification, or deletion on the environment

The Areal Object State PDU shall be issued by a simulation application using a unicast communication service when transmitting to the Environment Manager. The Environment Manager shall issue the Areal Object State PDU using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.11.3.4.4 Receipt of the Areal Object State PDU

Upon receipt of the Areal Object State PDU from a simulation application, the Environment Manager shall determine the effect that the creation, modification, or deletion of the object has on the environment. Based on this determination, the Environment Manager shall issue the Areal Object State PDU to all other simulation applications. The Environment Manager shall be able to modify the object based on the environment before transmittal to all simulation applications.

Upon receipt of the Areal Object State PDU from the Environment Manager, the simulation application shall update the local representation of the environment.

5.12 Simulation management with reliability

5.12.1 General

The content and the procedure for use of Simulation Management with Reliability PDUs with reliability service levels in a DIS exercise are established in 5.12.2 through 5.12.6.4. These PDUs are an alternative to the Simulation Management PDUs specified in 5.6.2 through 5.6.7.4. The Simulation Management with Reliability protocol may or may not be required for participation in an DIS exercise, depending on the requirements of each exercise.

The Simulation Management with Reliability protocol contains the following PDUs:

a) Create Entity-R

b) Remove Entity-R

c) Start/Resume-R

d) Stop/Freeze-R

e) Acknowledge-R

f) Action Request-R

g) Action Response-R

h) Data Query-R

i) Set Data-R
j) Data-R
k) Event Report-R
l) Comment-R
m) Record Query-R
n) Set Record-R
o) Record-R

All references to Simulation Management PDUs in 5.12.2 through 5.12.6.4 and in 7.11.2 through 7.11.16 shall refer to the above listed Simulation Management with Reliability protocol PDUs.

5.12.2 Simulation management transactions

The process of performing an individual simulation management function is termed a simulation management transaction. The application that initiates the transaction is called a Transaction Initiator (TI), and an application or entity to which the simulation management transaction is addressed is called a Transaction Respondent (TR). A simulation management transaction starts when the TI application issues a Simulation Management PDU and ends when all Simulation Management PDU(s) sent in reply by the TR(s) have been received by the TI. Note that applications may be engaged in more than one transaction simultaneously and may be simultaneously acting as both a TI and a TR in separate transactions.

A Simulation Management PDU sent from a TI to a TR is termed a TI PDU, whereas a Simulation Management PDU sent from a TR to a TI is termed a TR PDU. The following PDUs are classed as TI PDUs:

a) Create Entity-R
b) Remove Entity-R
c) Start/Resume-R
d) Stop/Freeze-R
e) Action Request-R
f) Data Query-R
g) Set Data-R
h) Record Query-R
i) Set Record-R

The following PDUs are classed as TR PDUs:

- Acknowledge-R
- Action Response-R
- Data-R
- Event Report-R
- Comment-R
- Record-R

5.12.3 Simulation management reliability levels

5.12.3.1 General

The TI shall determine the level of reliability service associated with each simulation management transaction. There are two reliability levels of service defined for use with the Simulation Management with Reliability protocol: an acknowledged service and an unacknowledged service. Simulation applications shall be able to interact with TIs at both reliability levels of service. The default reliability service for those simulation applications supporting the Simulation Management with Reliability protocol is the acknowledged service. In addition to the use of the reliability services, the TI may reduce the probability of a transmission failure affecting the simulation by issuing multiple copies of the initiating Simulation Management PDU. The Retry Counter parameter shall be identified by the symbolic name SM_REL_RETRY_CNT. The repeated PDUs should be issued at a predetermined real-world time interval. This interval shall be identified by the symbolic name SM_REL_RETRY_DELAY. (See 6.1.8 for parameter details and default values for these two parameters.) Each retransmitted PDU should contain the same data, including the same request ID. A TR can determine when it receives a retransmission by checking the request ID. If acknowledgments are required for this transaction, then the TR shall acknowledge every PDU it receives from the SM, including retransmitted PDUs. The only action the TR should take on receipt of a retransmitted PDU is to acknowledge it.

5.12.3.2 Procedure for acknowledged simulation management service

5.12.3.2.1 General

The following subclauses describe the different states and transitions for the acknowledged simulation management service.

5.12.3.2.2 Tracking transaction respondents

If the acknowledged simulation management service is used, the TI shall track the status of the transaction for each addressed TR. The status of the transaction for a particular TR shall be tracked if the TI receives a valid TR PDU from the TR, or if the TI expects to receive TR PDUs from the TR. The transaction status may be one of the following:

a) *Unknown.* The TI has not yet received a valid TR PDU from the TR.
b) *Incomplete.* The TI has received a valid TR PDU from the TR with the Response Flag/Request Status denoting incomplete.
c) *Complete.* The TI has received a valid TR PDU from the TR with the Response Flag/Request Status denoting complete.
d) *Timed Out.* The TI has determined that either the TR is not responding or the communications path from the TR has failed.

The TI shall also keep a count of the number of retries sent to each TR.

5.12.3.2.3 Acknowledged simulation management service timers

The procedures associated with the acknowledged simulation management service require the definition of a number of timers. The value of these timers shall be predetermined. If no values are established, the following default values for these timers shall be used:

a) TI Timer 1: TI_TIMER1_DFLT
b) TI Timer 2: TI_TIMER2_DFLT
c) TR Timer 1: TR_TIMER1_DFLT
d) TR Timer 2: TR_TIMER2_DFLT

5.12.3.2.4 TI behavior during acknowledged service

For each transaction that the TI is processing, the transaction is in one of three states:

a) *Ready State.* The transaction is in the Ready state when it is not in either the Requesting state or the Monitoring state.

b) *Requesting State.* The transaction is in the Requesting state when multiple initiating TI PDUs are to be sent but not all of them have been transmitted.

c) *Monitoring State.* The transaction is in the Monitoring state when all the initiating TI PDUs have been sent but end of transaction TR PDUs from all addressed TRs have not been received.

The behavior of the TI for each transaction during the acknowledged service is shown in Figure 27. The associated state transitions for the initiator are as shown in Table 21.

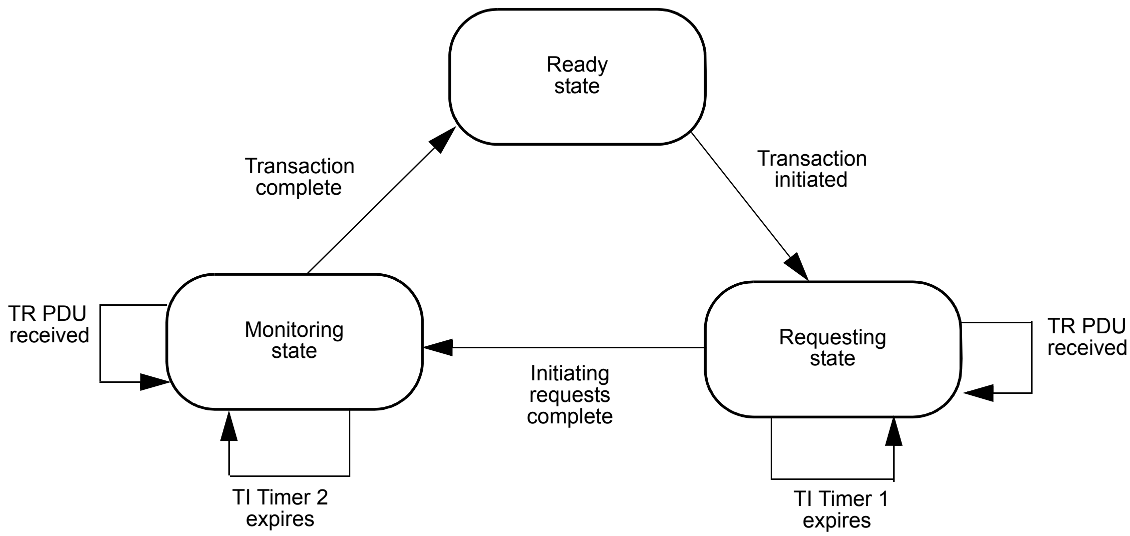


Figure 27—TI behavior during acknowledged simulation management service

Table 21—TI initiator state transitions for acknowledge service

Transition	Condition and actions
Transaction initiated	The TI shall issue a TI PDU (see 5.12.2). TI Timer 1 for this transaction shall be set. The TI shall proceed from the Ready state to the Requesting state for this transaction.
TR PDU received	When a TR PDU is received, the TI shall update the status of the transaction for the TR that sent the TR PDU (see 5.12.2). The retry count for this TR shall be set to zero.
TI Timer 1 expires	When TI Timer 1 expires, the TI shall reissue the initiating TI PDU. The count of initiating TI PDUs sent for this transaction shall be incremented, and TI Timer 1 for this transaction shall be set.

Table 21—TI initiator state transitions for acknowledge service *(continued)*

Transition	Condition and actions
Initiating requests complete	When the required number of initiating TI PDUs for this transaction has been sent, the transaction shall proceed from the Requesting state to the Monitoring state. The TI Timer 2 for this transaction shall be set, and TI Timer 1 shall be canceled.
TI Timer 2 expires	When TI Timer 2 expires, then the TI shall check the status of all TRs. If the status of any TR is either unknown or incomplete and a TR PDU has not been received from that TR during the TI Timer 2 period, then the retry count for that TR shall be incremented. If the retry count is less than a predetermined value, then the TI shall retransmit the TI PDU to the associated TR. If the retry count is equal to the predetermined value, then the transaction shall be marked as timed out. TI Timer 2 shall be set.
Transaction complete	When the status of the transaction for all TRs is either complete or timed out, then the transaction shall proceed from the Monitoring state to the Ready state. TI Timer 2 shall be canceled.

5.12.3.2.5 TR behavior during acknowledged service

For each transaction that the TR is processing, the transaction is in one of three states:

a) *Ready State.* A transaction is in the Ready state when it is not in either the Responding state or the Finished state.

b) *Responding State.* A transaction is in the Responding state when it has received a TI PDU but has not yet completed the requested operation.

c) *Finished State.* A transaction is in the Finished state when it has completed the requested operation and is waiting to see whether the TI requires confirmation of the results of the operation.

The behavior of the simulation application during the acknowledged service is shown in Figure 28. The associated state transitions for the TR are as shown in Table 22.

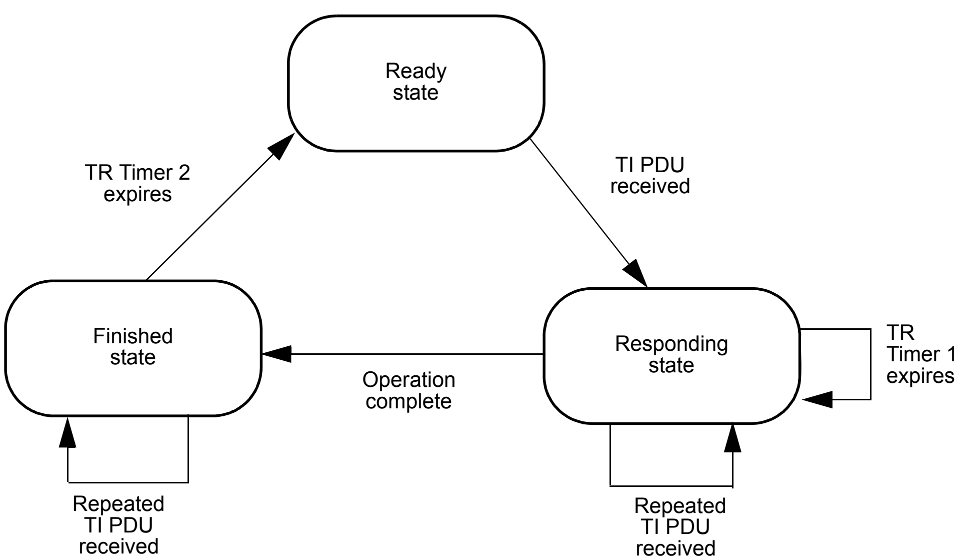


Figure 28—TR behavior during acknowledged simulation management service

Table 22—TR state transitions for acknowledge service

Transition	Condition and actions
TI PDU received	When a TI PDU is received, the TR shall determine whether it can complete the request immediately. If the TR can complete the request immediately, then it shall issue an appropriate TR PDU with the Response Flag/Request Status field indicating that it can comply with the request. If the TR cannot complete the request immediately, then it shall issue an appropriate TR PDU with the Response Flag/Request Status field indicating whether the request is pending or in progress. TR Timer 1 shall be set. The transaction shall proceed from the Ready state to the Responding state.
Repeated TI PDU received	When a TI PDU is received with the same Request ID, the TR will reissue the TR PDU with the Response Flag/Request Status field indicating the current status of the transaction. If the transaction is in the Responding state, then TR Timer 1 shall be set. If the transaction is in the Finished state, then TR Timer 2 shall be set.
TR Timer 1 expires	When TR Timer 1 expires, the TR PDU shall be reissued with the Response Flag/Request Status field indicating the current status of the transaction. TR Timer 1 shall be set.
Operation complete	When the TR completes the request, it shall reissue the TR PDU with the Response Flag/Request Status field indicating whether it has complied with the request (and a reason if it could not). TR Timer 1 shall be canceled, and TR Timer 2 shall be set. The transaction shall proceed from the Responding state to the Finished state.
TR Timer 2 expires	When TR Timer 2 expires, the transaction shall proceed from the Finished state to the Ready state.

5.12.3.3 Procedure for unacknowledged simulation management service

If the TI requires the unacknowledged simulation management service to be used, the TR shall not issue any TR PDUs. Since the TI may choose to issue multiple copies of the TI PDU, the TR should check each TI PDU received to see whether it has been received before. Only the first request received by a TR shall be processed.

5.12.4 Simulation Management with Reliability PDUs

5.12.4.1 General

The following PDUs shall be used to perform simulation management functions with reliability service.

5.12.4.2 Create Entity-R PDU

5.12.4.2.1 Purpose

The Create Entity-R PDU serves the same function as the Create Entity PDU (see 5.6.5.2.1) but with the addition of reliability service levels (see 5.12.3).

5.12.4.2.2 Information contained in the Create Entity-R PDU

The Create Entity-R PDU shall contain the same information as found in the Create Entity PDU (see 5.6.5.2.2) with the addition of the level of reliability service to be used for the creation action being requested.

5.12.4.2.3 Issuance of the Create Entity-R PDU

In addition to the issuance rules in 5.6.5.2.3, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.2.4 Receipt of the Create Entity-R PDU

Upon receipt of the Create Entity-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Create Entity-R PDU is a new request and the receiving simulation can comply with the Create Entity-R PDU request, then:
    1) It shall assign an entity identification number to the new entity as requested by the SM.
    2) If the Create Entity-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.
b) If the Create Entity-R PDU is a retransmission of a previously received Create Entity-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.

5.12.4.3 Remove Entity-R PDU

5.12.4.3.1 Purpose

The Remove Entity-R PDU serves the same function as the Remove Entity PDU (see 5.6.5.3.1) but with the addition of reliability service levels (see 5.12.3).

5.12.4.3.2 Information contained in the Remove Entity-R PDU

The Remove Entity-R PDU shall contain the same information as found in the Remove Entity PDU (see 5.6.5.3.2) with the addition of the level of reliability service to be used for the removal action being requested.

5.12.4.3.3 Issuance of the Remove Entity-R PDU

In addition to the issuance rules in 5.6.5.3.3, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.3.4 Receipt of the Remove Entity-R PDU

Upon receipt of the Remove Entity-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Remove Entity-R PDU is a new request and the receiving simulation can comply with the Remove Entity-R PDU request, then:
    1) The receiving simulation shall immediately cease simulating the entity identified in the Remove Entity-R PDU and shall remove it from the simulation exercise as quickly as possible.

2) The simulation entity that is being removed from the exercise shall issue at least one more Entity State PDU following the acknowledgment of the Remove Entity-R PDU. The last Entity State PDU issued by the deactivated entity shall have the Entity Appearance record State field (bit 23) set to Deactivated (1).

3) If the Remove Entity-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.

b) If the Remove Entity-R PDU is a retransmission of a previously received Remove Entity-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.

NOTE—Entities interacting with the removed entity will be able to recognize that the removed entity has left the exercise through the reception of the Entity State or Entity State Update PDU from the removed entity with the Entity Appearance record State field (bit 23) set to Deactivated (1) and the absence of further Entity State or Entity State Update PDUs for the removed entity.

5.12.4.4 Start/Resume-R PDU

5.12.4.4.1 Purpose

The Start/Resume-R PDU serves the same function as the Start/Resume PDU (see 5.6.5.4.1) but with the addition of reliability service levels (see 5.12.3).

5.12.4.4.2 Information contained in the Start/Resume-R PDU

The Start/Resume-R PDU shall contain the same information as found in the Start/Resume PDU (see 5.6.5.4.2) with the addition of the level of reliability service to be used for the start/resume being requested.

5.12.4.4.3 Issuance of the Start/Resume-R PDU

In addition to the issuance rules in 5.6.5.4.3, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.4.4 Receipt of the Start/Resume-R PDU

Upon receipt of the Start/Resume-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Start/Resume-R PDU is a new request, then:

1) A simulation may restrict the receipt of Start/Resume-R PDUs to those PDUs that originated from one or more designated SMs, or it may discard received Start/Resume-R PDUs if it is in the overall simulation management role.

2) A simulation shall discard a Start/Resume-R PDU if it is currently active in an exercise, not paused, and is issuing or receiving PDUs.

3) If the Start/Resume-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.

4) When an exercise is using Relative Time, any simulation that is joining or rejoining the exercise shall not start transmitting PDUs and interacting with data until either (i) it receives an appropriate Start/Resume-R PDU or (ii) there is coordination between the SM and the

simulation to provide the necessary Relative Time value and a manual procedure to activate the simulation at the specified Relative Time that matches a Real-World Time.

5) Each simulation shall protect itself from a Start/Resume-R PDU that is not applicable to it or that is received after it is already active in the exercise.

6) A simulation shall set its local simulation time to the time specified in the Simulation Time field when the local host System Clock reaches the time specified in the Real-World Time field.

7) If the Real-World Time field contains a time that is earlier than the present real-world time, the receiving application shall start or resume immediately. This condition may be considered an error that may be reported with an Event Report-R PDU.

b) If the Start/Resume-R PDU is a retransmission of a previously received Start/Resume-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.

5.12.4.5 Stop/Freeze-R PDU

5.12.4.5.1 Purpose

The Stop/Freeze-R PDU serves the same function as the Stop/Freeze PDU (see 5.6.5.5.1) but with the addition of reliability service levels (see 5.12.3).

5.12.4.5.2 Information contained in the Stop/Freeze-R PDU

The Stop/Freeze-R PDU shall contain the same information as found in the Stop/Freeze PDU (see 5.6.5.5.2) with the addition of the level of reliability service to be used for the stop/freeze action being requested.

5.12.4.5.3 Internal state of a frozen entity

See 5.6.5.5.3.

5.12.4.5.4 Issuance of the Stop/Freeze-R PDU

In addition to the issuance rules in 5.6.5.5.4, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.5.5 Receipt of the Stop/Freeze-R PDU

Upon receipt of the Stop/Freeze-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Stop/Freeze-R PDU is a new request, then:

1) If the Stop/Freeze-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.

2) The receiving simulation(s) or the specified entity shall leave the Simulating state at the real-world time indicated in the PDU.

3) The stop/freeze condition may be temporary or indefinite based on the reason for stop/freeze.

4) If the Real-World Time field contains a time that is earlier than the present real-world time, the receiving application shall stop or freeze immediately. This condition may be considered an error that may be reported with an Event Report-R PDU.

b) If the Stop/Freeze-R PDU is a retransmission of a previously received Stop/Freeze-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Acknowledge-R PDU.

5.12.4.6 Acknowledge-R PDU

5.12.4.6.1 Purpose

The Acknowledge-R PDU serves the same function as the Acknowledge PDU (see 5.6.5.6.1) but is used to acknowledge the receipt of a Create Entity-R PDU, a Remove Entity-R PDU, a Start/Resume-R PDU, or a Stop/Freeze-R PDU.

5.12.4.6.2 Information contained in the Acknowledge-R PDU

The Acknowledge-R PDU shall contain the same information as the Acknowledge PDU (see 5.6.5.6.2).

5.12.4.6.3 Issuance of the Acknowledge-R PDU

The Acknowledge-R PDU shall be issued by a simulation application in response to a Create Entity-R PDU, a Remove Entity-R PDU, a Start/Resume-R PDU, or a Stop/Freeze-R PDU that has indicated that the acknowledged reliability service should be used. In this case, the issuance rules in 5.6.5.6.3 apply.

5.12.4.6.4 Receipt of the Acknowledge-R PDU

Upon receipt of the Acknowledge-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving SM shall note that the simulation has successfully received the previous PDU transmission.

5.12.4.7 Action Request-R PDU

5.12.4.7.1 Purpose

The Action Request-R PDU serves the same function as the Action Request PDU (see 5.6.5.7.1) but with the addition of reliability service levels (see 5.12.3).

5.12.4.7.2 Information contained in the Action Request-R PDU

The Action Request-R PDU shall contain the same information as found in the Action Request PDU (see 5.6.5.7.2) with the addition of the level of reliability service to be used for the requested action.

5.12.4.7.3 Issuance of the Action Request-R PDU

In addition to the issuance rules in 5.6.5.7.3, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.7.4 Receipt of the Action Request-R PDU

Upon receipt of the Action Request-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Action Request-R PDU is a new request, then:
   1) The receiving simulation(s) or the specified entity/entities shall implement the request action if possible.
   2) If the Action Request-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Action Response-R PDU.
b) If the Action Request-R PDU is a retransmission of a previously received Action Request-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing an Action Response-R PDU.

5.12.4.8 Action Response-R PDU

5.12.4.8.1 Purpose

The Action Response-R PDU serves the same function as the Action Response PDU (see 5.6.5.8.1) but is used to acknowledge the receipt of an Action Request-R PDU.

5.12.4.8.2 Information contained in the Action Response-R PDU

The Action Response-R PDU shall contain the same information as the Action Response PDU (see 5.6.5.8.2).

5.12.4.8.3 Issuance of the Action Response-R PDU

The Action Response-R PDU shall be issued in response to an Action Request-R PDU that has indicated that the acknowledged reliability service should be used. In this case, the issuance rules in 5.6.5.8.3 apply.

5.12.4.8.4 Receipt of the Action Response-R PDU

Upon receipt of the Action Response-R PDU the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the originator of the Action Request-R PDU shall note that the action request was received and the status of that request.

5.12.4.9 Data Query-R PDU

5.12.4.9.1 Purpose

The Data Query-R PDU serves the same function as the Data Query PDU (see 5.6.5.9.1) but with the addition of reliability service levels (see 5.12.3).

5.12.4.9.2 Information contained in the Data Query-R PDU

The Data Query-R PDU shall contain the same information as found in the Data Query PDU (see 5.6.5.9.2) with the addition of the level of reliability service to be used for the data being requested.

5.12.4.9.3 Issuance of the Data Query-R PDU

In addition to the issuance rules in 5.6.5.9.3, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.9.4 Receipt of the Data Query-R PDU

Upon receipt of the Data Query-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU and the Data Query-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Data-R PDU containing the value or values (if any) of the datum requested in the Data Query-R PDU to which the simulation application is capable of replying.

NOTE—The use of the unacknowledged service with this PDU is not recommended because no Data-R PDU will be issued by the receiving simulation application.

5.12.4.10 Set Data-R PDU

5.12.4.10.1 Purpose

The Set Data-R PDU serves the same function as the Set Data PDU (see 5.6.5.10.1) but with the addition of reliability service levels (see 5.12.3).

5.12.4.10.2 Information contained in the Set Data-R PDU

The Set Data-R PDU shall contain the same information as found in the Set Data PDU (see 5.6.5.10.2) with the addition of the level of reliability service to be used.

5.12.4.10.3 Issuance of the Set Data-R PDU

In addition to the issuance rules in 5.6.5.10.3, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.10.4 Receipt of the Set Data-R PDU

Upon receipt of the Set Data-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Set Data-R PDU is a new request, then:
   1) The addressed simulation or entity shall set the appropriate parameters as specified in the Set Data-R PDU. It shall be up to the receiver to determine which (if any) parameters described in the Set Data-R PDU can be set.
   2) If the Set Data-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Data-R PDU.

b) If the Set Data-R PDU is a retransmission of a previously received Set Data-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Data-R PDU.

5.12.4.11 Data-R PDU

5.12.4.11.1 Purpose

The Data-R PDU serves the same function as the Data PDU (see 5.6.5.11.1) but with the addition of reliability service levels (see 5.12.3) and is used in response to a Data Query-R PDU, a Data-R PDU, or a Set Data-R PDU. It is used to provide information requested in a Data Query-R PDU, to confirm information received in a Set Data-R PDU, and to confirm the receipt of a periodic or unsolicited Data-R PDU when the acknowledged reliability service level is requested.

5.12.4.11.2 Information contained in the Data-R PDU

The Data-R PDU shall contain the same information as found in the Data PDU (see 5.6.5.11.2) with the addition of the level of reliability service to be used.

5.12.4.11.3 Issuance of the Data-R PDU

The Data-R PDU shall be issued in response to a Data Query-R PDU or a Set Data-R PDU that has indicated that the acknowledged reliability service should be used. In this case, the issuance rules in 5.6.5.11.3 apply.

The Data-R PDU shall also be used to confirm the receipt of a periodic or unsolicited Data-R PDU that has indicated that the acknowledged reliability service should be used.

When issuing a Data-R PDU, the Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

5.12.4.11.4 Receipt of the Data-R PDU

Upon receipt of the Data-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Data-R PDU is a new request, then:
   1) The receiving simulation shall record the information for simulation management purposes.
   2) If the Data-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Data-R PDU with the same request ID and the requested reliability service set to Unacknowledged (1). The reply need not contain any parameter values.

b) If the Data-R PDU is a retransmission of a previously received Data-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Data-R PDU with the same request ID and the requested reliability service set to Unacknowledged (1). The reply need not contain any parameter values.

5.12.4.12 Event Report-R PDU

5.12.4.12.1 Purpose

The Event Report-R PDU serves the same function as the Event Report PDU (see 5.6.5.12.1).

5.12.4.12.2 Information contained in the Event Report-R PDU

The Event Report-R PDU shall contain the same information as found in the Event Report PDU (see 5.6.5.12.2).

5.12.4.12.3 Issuance of the Event Report-R PDU

The Event Report-R PDU shall be issued according to the Event Report PDU issuance rules (see 5.6.5.12.3).

5.12.4.12.4 Receipt of the Event Report-R PDU

Upon receipt of the Event Report-R PDU the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving simulation may process, record, or ignore the PDU.

5.12.4.13 Comment-R PDU

5.12.4.13.1 Purpose

The Comment-R PDU serves the same function as the Comment PDU (see 5.6.5.13.1).

5.12.4.13.2 Information contained in the Comment-R PDU

The Comment-R PDU shall contain the same information as found in the Comment PDU (see 5.6.5.13.2).

5.12.4.13.3 Issuance of the Comment-R PDU

The Comment-R PDU shall be issued according to the Comment PDU issuance rules (see 5.6.5.13.3).

5.12.4.13.4 Receipt of the Comment-R PDU

Upon receipt of the Comment-R PDU the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the receiving simulation may process, record, or ignore the PDU.

5.12.4.14 Record Query-R PDU

5.12.4.14.1 Purpose

The Record Query-R PDU shall be used to communicate a request for data in record format.

5.12.4.14.2 Information contained in the Record Query-R PDU

The Record Query-R PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82)
b) Request identification for the records being requested
c) Level of reliability service to be used for the records being requested

d) Event type used to trigger the issuance of the PDU
e) Simulation time or simulation time interval between issues of requested information
f) Quantity and types of records requested

5.12.4.14.3 Issuance of the Record Query-R PDU

The following field specific requirements apply:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).
d) *Event Type*. A value of zero in this field shall indicate that the reporting shall be periodic based on the interval specified in the Time field. A non-zero value shall identify the type of event to be used as the trigger for the issuance of the PDU.
e) *Time*. A non-zero value in this field shall specify the simulation time interval between issues of Record-R PDUs or may represent a specific simulation time for time-based reporting. A value of zero in this field shall mean that the requested data should be sent once and not at any previously specified simulation time interval.

The Record Query-R PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.12.4.14.4 Receipt of the Record Query-R PDU

Upon receipt of the Record Query-R PDU the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU and the Record Query-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Record-R PDU containing the parameter value or values (if any) of the requested record to which the simulation application is capable of replying.

NOTE—The use of the unacknowledged service with this PDU is not recommended since no Record-R PDU will be issued by the receiving simulation application.

5.12.4.15 Set Record-R PDU

5.12.4.15.1 Purpose

The Set Record-R PDU shall be used to set or change certain parameter values. These parameter values are contained within a record format as compared to the datum format used in the Set Data-R PDU (see 5.12.4.10).

5.12.4.15.2 Information contained in the Set Record-R PDU

The Set Record-R PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82)
b) Request identification for the data issued
c) Level of reliability service to be used for the records being requested

d) Number of records, the record identifications, and the specific parameter values

5.12.4.15.3 Issuance of the Set Record-R PDU

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

The Set Record-R PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.12.4.15.4 Receipt of the Set Record-R PDU

Upon receipt of the Set Record-R PDU the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Set Record-R PDU is a new request, then:
    1) The addressed simulation or entity shall set the appropriate parameter values as specified in the Set Record-R PDU. It shall be up to the receiver to determine which (if any) parameter value or values described in the Set Record-R PDU can be set.
    2) If the Set Record-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Record-R PDU.
b) If the Set Record-R PDU is a retransmission of a previously received Set Record-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Record-R PDU.

5.12.4.16 Record-R PDU

5.12.4.16.1 Purpose

The Record-R PDU shall be used to respond to a Record Query-R PDU or a Set Record-R PDU. It is used to provide information requested in a Record Query-R PDU, to confirm the information received in a Set Record-R PDU, and to confirm the receipt of a periodic or unsolicited Record-R PDU when the acknowledged service level is used.

Record-R PDUs may be issued at a periodic rate or based on an event trigger. This rate and event trigger can be set in the Record Query-R PDU (see 5.12.4.14). The simulation or entity sending such Record-R PDUs shall decide which reliability service to use.

Record-R PDUs may be issued without first receiving a Set Record-R or Record Query-R PDU. The contents of the Record-R PDU in this case can be determined by the issuing simulation or entity. Error reporting shall be implemented in the Event-Report-R PDU (see 5.12.4.12).

5.12.4.16.2 Information contained in the Record-R PDU

The Record-R PDU shall contain the following information:

a) Simulation Management PDU Header (see 6.2.82)
b) Request identification for the records being sent
c) Level of reliability service to be used for the records being requested
d) Event type being reported
e) Serial number of the response
f) Number of record sets, record identifications, and specific record values

5.12.4.16.3 Issuance of the Record-R PDU

The Record-R PDU may be issued at regular intervals or when internal conditions are met. If a Record Query-R PDU or a Set Record-R PDU indicates that the acknowledged reliability service shall be used for a transaction, a Record-R PDU shall be issued on receipt of the PDU.

When the Record-R PDU is issued in response to a Record Query-R PDU, it shall contain the parameter value or values (if any) of the requested records to which the simulation or entity is capable of replying.

When the Record-R PDU is issued in response to the Set Record-R PDU, it shall verify the receipt of the Set Record-R PDU by returning the parameter values that were set in response to the Set Record-R PDU. Parameter values that were set in the simulation to the same values as in the Set Record-R PDU shall be set to those values in the Record-R PDU. Parameter values that were set to different values in the simulation than requested in the Set Record-R PDU shall be set to their actual values in the Record-R PDU. Parameters to which the receiving simulation or entity cannot comply shall not be included in the Record-R PDU.

The following specific requirements shall be met:

a) The Originating ID field shall be set to the ID of the simulation or entity that is issuing the PDU (see 5.6.2.3).
b) The Receiving ID field shall identify the intended recipient or recipients of the PDU (see 5.6.2.4).
c) The Required Reliability Service field shall be set to the level of reliability service to be used for this transaction (see 5.12.3).

The Record-R PDU shall be issued using a best effort unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

5.12.4.16.4 Receipt of the Record-R PDU

Upon receipt of the Record-R PDU, the receiving simulation shall determine whether it is the intended recipient (see 5.6.2.5). If it is not the intended recipient of the PDU, then the receiving simulation shall ignore the PDU.

If it is the intended recipient of the PDU, then the following rules shall apply:

a) If the Record-R PDU is a new request, then:
    1) The receiving simulation shall record the information for simulation management purposes.
    2) If the Record-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Record-R PDU with the same request ID and the requested reliability service set to Unacknowledged (1). The reply need not contain any parameter values.

b) If the Record-R PDU is a retransmission of a previously received Record-R PDU and indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond by issuing a Record-R PDU with the same request ID and the requested reliability service set to Unacknowledged (1). The reply need not contain any parameter values.

5.12.5 Entity/exercise management

5.12.5.1 General

Management of an entity or exercise includes the capability to create new entities, initialize or change entity or exercise parameters, start or stop an entity or exercise, request an entity to perform a specific action, and record significant entity data or exercise events. These actions are described in 5.12.5.2 through 5.12.5.7.

5.12.5.2 Creating a new entity

5.12.5.2.1 General

The Simulation Management with Reliability protocol provides three ways to create a new entity. The first method allows the SM to establish the identification (using Create Entity-R PDU) of the new entity, query for data about the new entity (using Data Query-R PDU), and set initial parameters for the new entity (using Set Data-R PDU). The second method is similar to the first method except the SM does not query for data. The third method is even more streamlined, requiring only the Create Entity-R PDU; however, the third method requires certain database information to be established in advance of the exercise start. In contrast, the first method allows the entire creation and initialization process to proceed with little information established in advance. The three methods of entity creation are described in 5.12.5.2.2 through 5.12.5.2.4.

Entities may always enter an exercise without being created through the Simulation Management PDUs. An entity enters an exercise by exchanging PDUs (Entity State PDUs, for example) with other entities in the DIS exercise.

5.12.5.2.2 (First method) Entity creation, query, and initialization

To create, query, and initialize a new entity, either for an existing exercise or for a new exercise, the SM shall begin by issuing a Create Entity-R PDU to the simulation application that will be controlling the simulation entity. If the Create Entity-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond with an Acknowledge-R PDU. These actions simply assign an identification number to a new entity.

The SM then shall request that certain data be issued by the simulation application controlling the new entity. This request is accomplished by issuing a Data Query-R PDU. The Data Query-R PDU shall use an acknowledged reliability service. The simulation application shall respond by sending the requested data using a Data-R PDU.

The SM then shall send necessary initialization information to the new entity using a Set Data-R PDU. If the Set Data-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the entity shall indicate the receipt of the Set Data-R PDU by issuing a Data-R PDU.

This process of entity creation, query, and initialization is illustrated in Figure 29.

NOTE—In Figure 29 through Figure 38, a PDU transmission that is outlined in a dashed line denotes a transmission that is optional based on the reliability service requested in the immediately preceding PDU.

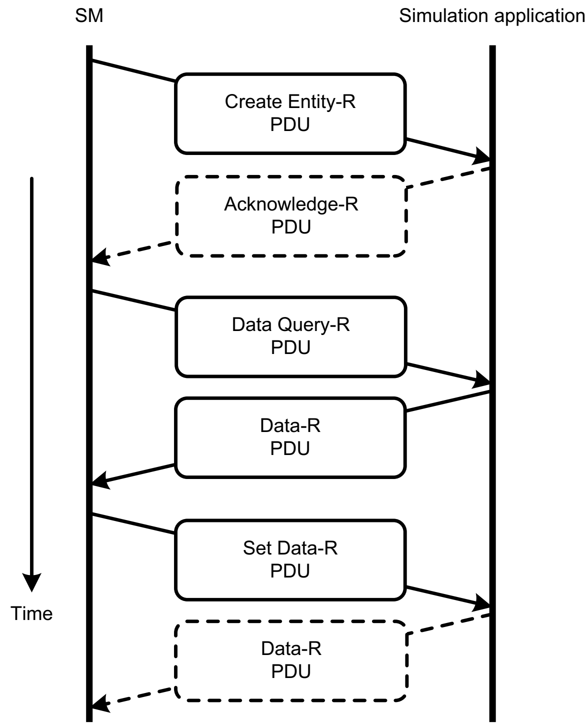


Figure 29—Entity creation, query, and initialization

5.12.5.2.3 (Second method) Entity creation and initialization

To create and initialize a new entity, the SM shall begin by issuing a Create Entity-R PDU to the simulation application that will be controlling the simulation entity. If the Create Entity-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond with an Acknowledge-R PDU. These actions simply assign an identification number to a new entity.

It is assumed that the SM has all the necessary data it needs for the new entity (e.g., type of entity and characteristics). This information shall be established offline and prior to the entity creation. The SM then shall initialize the new entity by issuing Set Data-R PDU to the simulation application controlling the new entity. If the Set Data-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the simulation application shall indicate the receipt of the Set Data-R PDU by issuing a Data-R PDU.

This process of entity creation and initialization is illustrated in Figure 30.

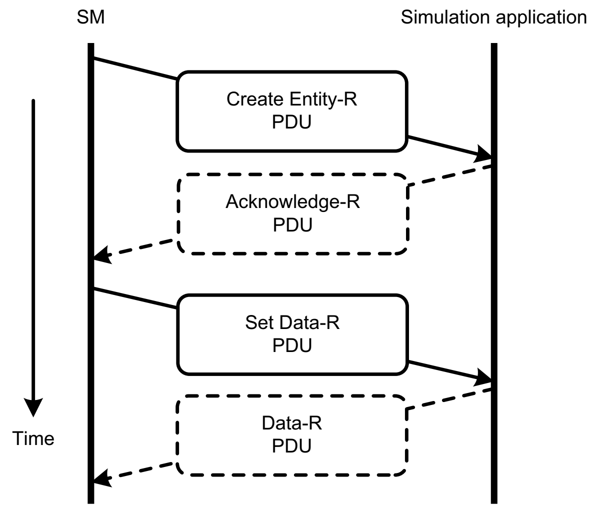


Figure 30—Entity creation and initialization

5.12.5.2.4 (Third method) Entity creation

When necessary entity data and initialization data have already been established offline and prior to the exercise, it is possible to create a new entity by assigning a particular entity identification. To create a new entity, the SM shall issue a Create Entity-R PDU to the simulation application that will be controlling the simulation entity. If the Create Entity-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving simulation application shall respond with an Acknowledge-R PDU.

This process of entity creation is illustrated in Figure 31.

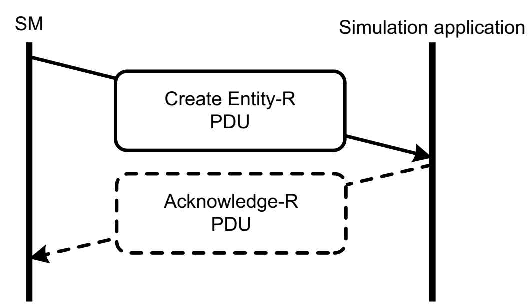


Figure 31—Entity creation

5.12.5.3 Changing entity parameters

Parameters within a particular entity shall be changed by the SM by the issue of a Set Data-R PDU. For example, during initialization, the Set Data-R PDU requests that an entity set (or change) certain parameters of its internal state to specified values. As in initialization, the receiving entity shall respond by issuing a Data-R PDU if the Set Data-R PDU request indicates that the acknowledged reliability service shall be used for this transaction. The receipt of a Data-R PDU will serve as an acknowledgment to the SM that the correct changes were made. Another Set Data-R PDU shall be issued by the SM if the first was incorrectly received.

5.12.5.4 Starting or stopping an entity

5.12.5.4.1 General

An entity, or the simulation application supporting the entity, shall be in one of three states—Wait, Stopped/Frozen, or Simulating. An entity does not participate in the exercise (i.e., does not transmit Entity State PDUs) while in the Wait state, although it is ready to be created and then initialized by the simulation application controlling it. The entity may also be in the Stopped/Frozen state when it is not simulating, but it is able to be started. Finally, an entity may be in the Simulating state; in which case, it is actually participating in an exercise.

These states are represented in Figure 32.

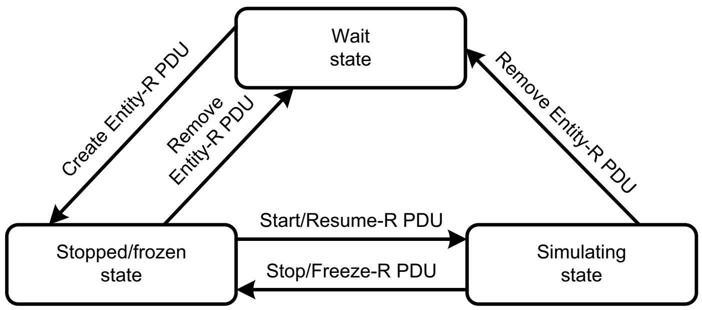


Figure 32—Entity states in simulation management

5.12.5.4.2 Change of frozen behavior

An entity shall change its Frozen behavior based on the Frozen Behavior field of the latest received Stop/Freeze-R PDU as indicated by the Request ID field. The Stop/Freeze-R PDU shall cause an entity to transmit or not transmit PDUs, update or freeze its internal simulation clock, and react to or ignore received PDUs.

An entity is in the Wait state until it is created and initialized (see 5.12.5.2). At this point the entity is in the Stopped/Frozen state. To start an entity, an SM shall issue a Start/Resume-R PDU to the entity to be started. If the Start/Resume-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond with an Acknowledge-R PDU. Similarly, to stop or freeze a simulating entity, an SM shall issue a Stop/Freeze-R PDU to the entity to be stopped. If the Stop/Freeze-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond with an Acknowledge-R PDU.

These actions are represented in Figure 33.

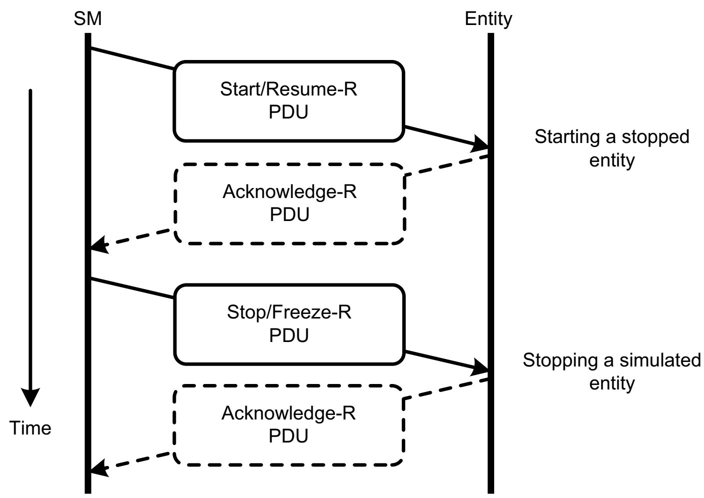


Figure 33—Starting/stopping an entity

5.12.5.5 Removing an entity from an exercise

An entity may be removed from an exercise by an SM. To remove the entity, an SM shall issue a Remove Entity-R PDU to the entity to be removed. The simulation application controlling the entity shall cease simulating the entity. If the Remove Entity-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the simulation application controlling the entity shall respond with an Acknowledge-R PDU. A removed entity shall issue one last Entity State PDU with the Entity Appearance record State field (bit 23) set to Deactivated (1).

These actions are represented in Figure 34.

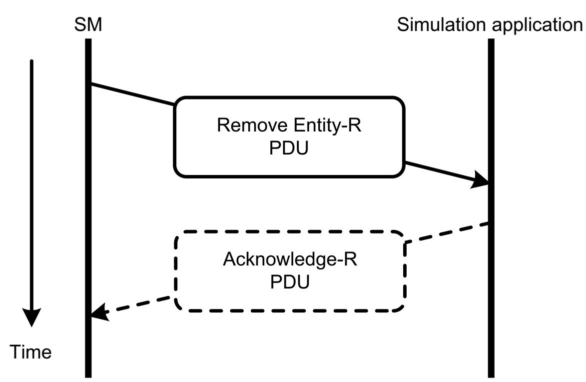


Figure 34—Removing an entity

5.12.5.6 Requesting an entity to perform an action

An entity may be requested to perform a specific action by the SM. In addition to requesting the action, the SM may provide needed information for performance of the requested action.

To request an action, the SM shall issue an Action Request-R PDU to the entity to perform the action. Upon receipt of the Action Request-R PDU, the receiving entity shall act on the request. If the Action Request-R PDU indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond with an Action Response-R PDU.

This interaction is represented in Figure 35.

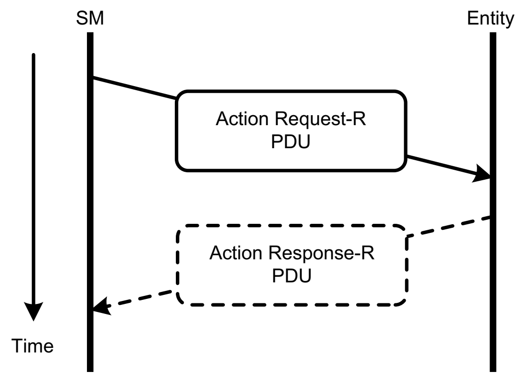


Figure 35—Action request/response

5.12.5.7 Recording an event

To keep a record of certain key events that may occur during the course of a DIS exercise, Event Report-R PDUs shall be used in the reporting of such events. The SM may set certain parameters so that a particular event will initiate the issuance of an Event Report-R PDU. When that event occurs, the entity involved issues an Event Report-R PDU to report the event.

5.12.6 Data management

5.12.6.1 General

In addition to managing entities and the exercise, data management may be accomplished using the Simulation Management with Reliability PDUs.

5.12.6.2 Requesting data

An SM shall request data concerning the internal state of an entity by issuing a Data Query-R PDU or a Record Query-R PDU. If the Data Query-R PDU or the Record Query-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond by providing the requested data in a Data-R PDU.

These actions are represented in Figure 36.

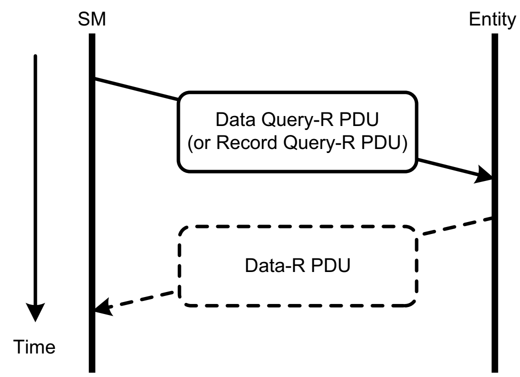


Figure 36—Requesting entity data

5.12.6.3 Setting or changing internal state values

Internal state information can be set or modified by the SM. To set or modify this information, the SM shall issue a Set Data-R PDU indicating the data to be changed or set. The receiving entity shall determine which (if any) parameters described in the Set Data-R PDU it can set. If the Set Data-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond by issuing a Data-R PDU. The Data-R PDU shall verify the receipt of the Set Data-R PDU by returning the parameter values that were set in response to the Set Data-R PDU. Parameters that were set to the same values as in the Set Data-R PDU shall be set to those values in the Data-R PDU. Parameter values that were set to different values than requested shall be set to their actual values in the Data-R PDU. Parameters to which the receiving entity cannot comply shall not be included in the Data-R PDU response. This Data-R PDU serves as an acknowledgment to the SM. If the SM determines that the original request was incorrectly received, another Set Data-R PDU may be issued by the SM. The assurance that the data values contained in the Set Data-R PDU are within the valid range for the specified variable, are of the correct data structure, are properly named, etc., is outside the scope of this standard.

These actions are represented in Figure 37.

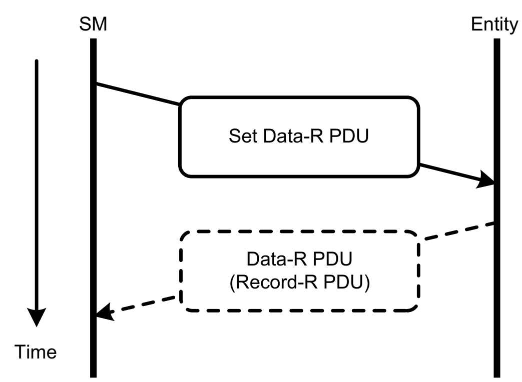


Figure 37—Setting/changing internal state values

The Set Record-R PDU can also be used to set or change internal state values in a similar manner to the Set Data-R PDU as discussed above. The Record-R PDU would be used to respond to the Set Record-R PDU in place of the Data-R PDU in the above discussion.

5.12.6.4 Reconstituting an entity

A killed or damaged entity shall allow its reconstitution by an SM when requested. If the same entity (in the exercise) is to be reconstituted, then the unique entity identification shall remain the same during the process. If the same entity is to be reconstituted as a different entity in the exercise, then the entity shall not be reconstituted, but rather, it shall be removed, and a new and different entity shall be created following 5.12.5.2.

To reconstitute an entity, an SM shall issue a Stop/Freeze-R PDU. If the Stop/Freeze-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond with an Acknowledge-R PDU. The entity shall not be removed but shall be in the Stopped/Frozen state as in Figure 32. The SM shall then issue a Set Data-R PDU to reset the entity parameters. If the Set Data-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond with a Data-R PDU. The SM shall then issue a Start/Resume-R PDU for the receiving entity to rejoin the exercise. If the Start/Resume-R PDU request indicates that the acknowledged reliability service shall be used for this transaction, then the receiving entity shall respond with an Acknowledge-R PDU.

These actions are represented in Figure 38.

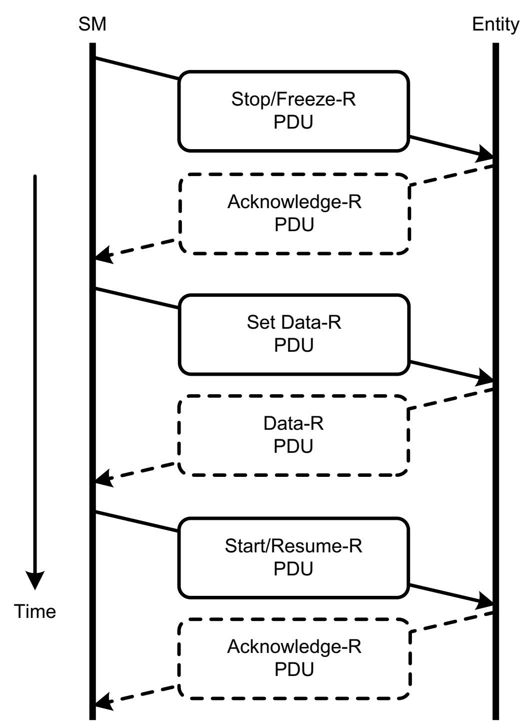


Figure 38—Reconstituting an entity

5.13 Information operations

5.13.1 General

Information associated with the representation of information operations (IO) are described in this subclause. Information operations (IO) are the integrated employment of electronic warfare (EW), computer network operations (CNO), psychological operations (PSYOP), military deception (MILDEC), and operations security (OPSEC), along with specific supporting capabilities, to influence, disrupt, corrupt, or otherwise affect enemy information and decision making while protecting friendly information operations. The IO Action and IO Report PDUs, and the use of IO records within those PDUs, are intended to support all types of information operations although the current IO records defined in 6.2.48 focus on IO attacks on communications nodes and networks. Additional IO records, issuance and receipt rules, and enumerations are expected to be developed as the capabilities of IO simulations evolve.

5.13.2 General requirements

The following general requirements shall apply:

a) *IO Attacker*. Simulations that are capable of carrying out IO attacks shall issue the appropriate PDUs to represent IO effects:
   1) An IO attack shall be associated with a specific IO attacker entity. This entity may be generated by the simulation performing the IO attack, or the IO simulation may attach itself to an entity owned by another simulation. In either case, the IO simulation shall monitor the entity for damage to its capability to launch an IO attack and shall cease such attacks or reduce its capability based on any damage sustained.
   2) Some PDUs are specific to IO, while others support other DIS areas such as an EE PDU used to represent jamming.
   3) Some IO weapons systems communicate the IO effect on a specific entity representing a node of a communications network or other IO targets and provide that calculated IO effect information to the simulation that is modeling the target of an IO attack.
   4) An IO Weapons System shall monitor the damage status of entities representing a target of an IO attack, including entity nodes and communications links associated with a node. It shall take into account the present damage status when performing an IO attack.

b) *IO Target*. Entities and their associated equipment, including communications networks, may be affected by Information Operations attacks. Simulations that own such entities may assess IO effects on themselves or may rely on another simulation to provide it with IO effects information:
   1) Some simulations may be required to accept the IO effects received from another simulation, and some may use the information along with its own internal IO effects assessment model to calculate IO effects on its entities and models.
   2) A simulation that owns an entity that is vulnerable to IO attack, such as an entity representing a node of a communications network, shall, as a minimum, process the Detonation PDU, to assess non-IO attack damage to the entity such as could be caused by a munition.

5.13.3 Information Operations (IO) Action PDU

5.13.3.1 General

The IO Action PDU shall be used to communicate an IO attack or the effects of an IO attack on one or more target entities. Targets include, but are not limited to, operational and communications centers, communication nodes such as microwave towers and relay facilities, and other associated equipment and links of communications networks. The information contained in the IO Action PDU is used by a receiving simulation to model the effects of the IO attack on its entities.

5.13.3.2 Information contained in the IO Action PDU

The IO Action PDU shall contain the following information:

a) The identifier of the simulation that is generating the IO attack and providing IO effects assessments.
b) The identifier of the simulation to which the IO Action PDU is addressed, if applicable.
c) The Request ID to allow acknowledgment of an IO action.
d) The IO warfare type.
e) The IO simulation source to indicate the model name that is issuing this PDU.
f) IO action type.
g) The phase of the IO attack.
h) The Entity ID of the entity that is carrying out the IO attack.
i) The primary entity ID of the target of the IO attack.
j) IO action records containing information on the IO attack and IO effects assessment, if appropriate.

5.13.3.3 Issuance of the IO Action PDU

A simulation shall issue an IO Action PDU as follows:

a) An IO Action PDU shall be issued when any of the following conditions occur:
   1) An IO attack has been initiated against a target entity.
   2) When a change occurs in IO attack profile data or one or more IO effects after an initial IO Action PDU has been sent. Only the IO Action records that have changed data need to be included.
   3) For other conditions determined by a simulation that issues an IO Action PDU.
b) The issued IO Action PDU shall comply with the following requirements:
   1) The Receiving Simulation ID shall be set to one of the following values:
      i) If addressed to a specific simulation, it shall contain the Simulation ID for that simulation.
      ii) If not addressed to a specific simulation, it shall be addressed to all simulations (All_SITES, ALL_APPLIC, NO_REF_NUMBER). See Table 3 and Table 25.
   2) The Request ID shall be set based on the Receiving Simulation ID field value as follows:
      i) If the Receiving Simulation ID field contains a Simulation ID, then this field shall contain a Request ID (see 6.2.75).
      ii) If the Receiving Simulation ID field is set to all simulations, then this field shall be set to zero.
   3) The IO attack profile data that is needed by the target being attacked in order for the target to determine the effect on its IO capability.
   4) The calculated IO effects on the target as determined by the attacker simulation. These may be reported initially and during the attack or only reported at the end of the attack.
   5) Appropriate IO records shall be included.

5.13.3.4 Receipt of the IO Action PDU

Upon receipt of an IO Action PDU, a simulation shall use the information therein to determine the effects on its IO capability or, if the IO effects are provided, may use them in lieu of making the assessment itself:

a) An Acknowledge PDU shall be issued by the simulation responsible for maintaining IO effects for the primary target entity contained in the IO Action PDU. The fields of the Acknowledge PDU shall be set as follows:

1) PDU Header. Set appropriately.
2) Originating ID. Set to the simulation that is responding to this IO Action PDU.
3) Receiving ID. Set to the simulation contained in the Originating ID field of the corresponding IO Action PDU.
4) Acknowledge Flag. Set to IO Action PDU (6).
5) Response Flag. Set to Able to Comply (1) or Unable to Comply (2) as appropriate.
6) Request ID. Set to the Request ID contained in the corresponding IO Action PDU.

b) The absence of an IO record in a subsequent IO Action PDU shall indicate that there has been no change in the data.

5.13.4 Information Operations (IO) Report PDU

5.13.4.1 General

The IO Report PDU shall be used to communicate the effects of an IO attack on one or more target entities. Targets include, but are not limited to, operational and communications centers, communication nodes such as microwave towers and relay facilities, and other associated equipment and links of communications networks. The information contained in the IO Report PDU is used by a receiving simulation to determine whether to continue an IO attack and to change attack parameters. It is also used by data analysts to identify IO interoperability problems and for IO data analysis.

5.13.4.2 Information contained in the IO Report PDU

The IO Report PDU shall contain the following information:

a) The identifier of the simulation that is generating the IO report.
b) The Entity ID of the entity or entities that are under an IO attack or which have been attacked.
c) The status of the effects of the IO attack on the entity or entities.

5.13.4.3 Issuance of the IO Report PDU

A simulation shall issue an IO Report PDU as follows:

a) An IO Report PDU shall be issued when any of the following conditions occurs:
   1) A target entity affected by an IO attack has determined the effect(s).
   2) When there is a change in one or more IO effects for an entity after an initial IO Report PDU has been issued.
   3) For other conditions determined by a simulation that issues an IO Report PDU.
b) The issued IO Report PDU shall comply with the following requirements:
   1) The calculated IO effects may be sent at the beginning of an IO attack after initial effects have been calculated or after the IO attack has ceased.
   2) A single IO Report PDU may include information on multiple entities comprising a portion of a communications network.
   3) Any physical damage to an entity for which an IO Report PDU is being issued and that affects IO data shall be reflected in the IO Report PDU.
   4) An Entity ID shall not be included in an IO Report PDU if the entity has been deactivated and is no longer in the exercise.

5.13.4.4 Receipt of the IO Report PDU

Upon receipt of an IO Report PDU, a simulation involved in IO simulation shall use the information therein as applicable to its simulation.

6 Detailed requirements

6.1 Representation of data

6.1.1 General

Octet ordering, enumeration representation, number representation, and symbolic names used in this standard are described in 6.1.2 through 6.1.8.

6.1.2 Octet ordering

The order of transmission of the data described in this standard shall be resolved to the octet level. In Figure 39, the order of the transmission of octets is illustrated. The octets in Figure 39 shall be transmitted in the order in which they are numbered.

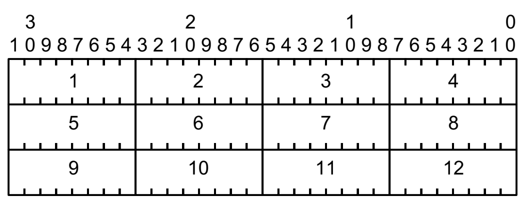


Figure 39—Transmission of octets

Whenever an octet represents a numeric quantity, the left-most bit in the diagram shall represent the high order or most significant bit. That is, the bit labeled 7 is the most significant bit. For example, Figure 40 represents the decimal value 170.

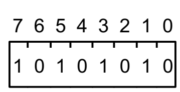


Figure 40—Significance of bits

Similarly, whenever a multi-octet field represents a single numeric quantity or is a bit field record, then the left-most bit of the entire multi-octet field shall be the most significant bit. When a multi-octet quantity is transmitted, the most significant octet (the one containing the most significant bit) shall be transmitted first. Multi-octet fields shall be 16, 32, or 64 bits in length.

Table 23 provides an example record to illustrate how octets are ordered when transmitted on the wire. The bit field record C is further defined in Table 24.

Table 23—Example record to illustrate octet ordering

Field size (bits)	Field name	Data type
48	A	A1—32-bit unsigned integer
		A2—16-bit unsigned integer
16	Padding1	16 bits unused
32	F	32-bit floating point
32	B	B1—8-bit enumeration
		Padding2—8 bits unused
		B2—16-bit unsigned integer
32	C	32-bit record (bit field)

Table 24—Example bit field record to illustrate octet ordering

Field name	Bits	Data type
C1	0–2	3-bit enumeration
Padding3	3–15	13 bits unused
C2	16–31	16-bit unsigned integer

This record is transmitted on the wire according to Figure 41. Note that in the transmission of bit field record C, its individual fields (C1, Padding3, and C2) are not transmitted in the order they exist within the record. Compare the ordering of the octets for record C with those of the fields in record B. As shown in Figure 41, octets 17 and 18 correspond to C2 of the bit field record C. Note that B1 is transmitted before B2, but C1 is not transmitted before C2.

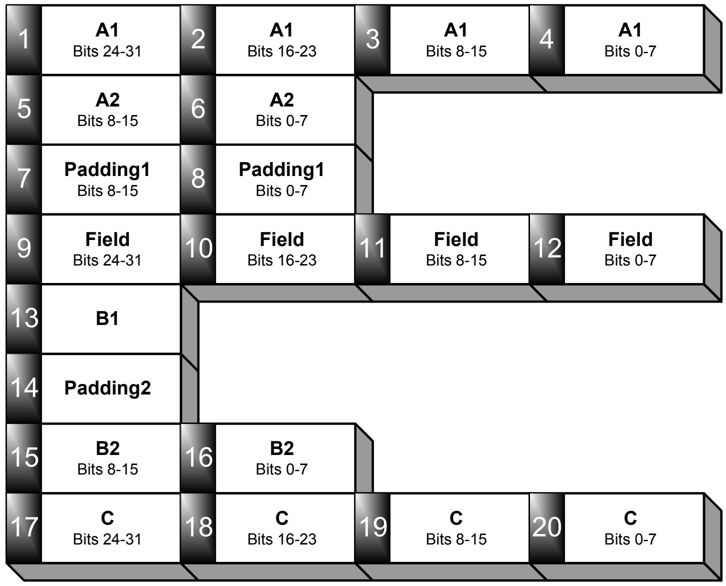


Figure 41—Octet ordering of the example record

6.1.3 Enumeration representation

All enumerations shall be unsigned integers and shall begin with zero for the first element of the enumeration. Enumerations not contained in records shall have a size of 8, 16, or 32 bits. Enumerations contained in records may have any size up to 32 bits.

6.1.4 Booleans

A Boolean data type shall be represented as a single bit representing a true-false value. This bit shall represent an enumeration type of one bit, where the value 0 is interpreted as false, and the value 1 as true.

6.1.5 Number representation

6.1.5.1 General

Numbers shall be represented as either floating point numbers or integers.

6.1.5.2 Floating point numbers

Single- and double-precision floating point numbers shall adhere to IEEE Std 754.

6.1.5.3 Integers

Integers shall be represented as signed or unsigned. Signed integers shall be represented in two's-complement form, where the most significant bit shall designate the sign bit. This bit shall have a value of zero for positive numbers and one for negative numbers. Integers not contained in records shall have a size of 8, 16, 32, or 64 bits. Integers contained in records may have any size up to 64 bits. Any integer that is not 8, 16, 32, or 64 bits shall be unsigned.

6.1.6 Bit field records

Bit field records have a field structure that is defined by explicit bit numbering. For example, a record consisting of three fields defined as bits 0–2, bits 3–7, and bits 8–15 is a bit field record. Bit field records shall be 8, 16, 32, or 64 bits in total length.

6.1.7 Padding

All padding fields shall be set to zero. The receipt of a PDU with nonzero padding field(s) shall not be sufficient to reject the PDU as invalid.

6.1.8 Symbolic names

The symbolic names contained in Table 25 define the fixed and variable parameters required by this standard except for timeout variable parameters associated with remote entities, other objects, and supplemental data, which are specified in item a) of 4.2.7.2. The requirement for each fixed and variable parameter, including associated tolerance values, are specified in applicable sections of this standard for individual PDU types. Fixed and variable parameters are defined prior to an exercise in an exercise agreement. These parameters may be exchanged via Simulation Management PDUs or other means at the direction of exercise management. A simulation shall only be required to implement the fixed and variable parameters that are associated with the PDU types that it implements.

The symbolic names are associated with numeric values. All numeric values are in base ten except for those followed by the suffix (H), which indicates a hexadecimal number. Some symbolic names are fixed parameters, and others are variable parameters that may have a value, including a fractional value. Simulations may desire to make variable parameters part of an initialization process such that a change in the value will not require a change to the simulation application.

All fixed parameters shall use the value shown in Table 25. All variable parameter values shall be positive numbers unless the default value indicates it may be either a positive or a negative number. If the default value indicates it may be a positive or negative number, that shall apply to any number value specified in an exercise agreement as well. All fixed parameters may be stored in any format that can denote the fixed numeric value. All variable parameters can have a value that may change from one exercise to another. The default value shown in Table 25 shall be used if the exercise agreement does not specify another value. All variable parameters shall be stored as 32-bit floating point numbers in order to accommodate a wide range of values including fractional values. The units associated with a variable parameter (e.g., s for seconds) shown in Table 25 shall apply whether it is a default value or a value defined in an exercise agreement.

See the Legend at the end of Table 25 for an explanation of the notations used.

Table 25—Symbolic names

Symbolic name	Fixed or variable	Fixed and default parameter values
AGG_RESPONSE_DFLT	Variable	Default: 10 s
ALL_AGGS	Fixed	FFFF (H)
ALL_APPLIC	Fixed	FFFF (H)
ALL_BEAMS	Fixed	FF (H)
ALL_EMITTERS	Fixed	FF (H)

Table 25—Symbolic names *(continued)*

Symbolic name	Fixed or variable	Fixed and default parameter values
ALL_ENTITIES	Fixed	FFFF (H)
ALL_OBJECTS	Fixed	FFFF (H)
ALL_SITES	Fixed	FFFF (H)
COLLISION_ELASTIC_TIMEOUT	Variable	Default: 5 s
COLLISION_THRSH	Variable	Default: 0.1 m/s
DE_AREA_AIMING_THRSH	Variable	Default: 10°
DE_ENERGY_THRSH	Variable	Default: 1.0%
DE_PRECISION_AIMING_THRSH	Variable	Default: 0.5 m
DRA_ORIENT_THRSH	Variable	Default: 3°
DRA_POS_THRSH	Variable	Default: 1 m
D-SPOT_NO_ENTITY	Fixed	NO_SITE: NO_APPLIC: NO_ENTITY
EE_AD_PULRAT_THRSH	Variable	Default: 0.017 rad/s
EE_AD_PULACC_THRSH	Variable	Default: 0.017 rad/s^2
EE_AZ_THRSH	Variable	Default: 1°
EE_EL_THRSH	Variable	Default: 1°
EE_ERP_THRSH	Variable	Default: 1.0 dBm
EE_FREQ_THRSH	Variable	Default: 1 Hz
EE_FRNG_THRSH	Variable	Default: 1 Hz
EE_FT_VEL_THRSH	Variable	Default: 1.0 m/s
EE_FT_ACC_THRSH	Variable	Default: 1.0 m/s^2
EE_FT_MWD_THRSH	Variable	Default: 10000 m
EE_FT_KT_THRSH	Variable	Default: 10 s
EE_FT_ESP_THRSH	Variable	Default: 10 m
EE_HIGH_DENSITY_THRSH	Variable	Default: 10 entities/beam
EE_PRF_THRSH	Variable	Defauilt: 1 Hz
EE_PW_THRSH	Variable	Default: 1 µs
ENTITY_ID_UNKNOWN	Fixed	NO_SITE: NO_APPLIC: NO_ENTITY
EP_DIMENSION_THRSH	Variable	Default: 1 m
EP_NO_SEQUENCE	Fixed	FFFF (H)
EP_POS_THRSH	Variable	Default: 1 m shift

Table 25—Symbolic names *(continued)*

Symbolic name	Fixed or variable	Fixed and default parameter values
EP_STATE_THRSH	Variable	User defined Default: ±10%
GD_GEOMETRY_CHANGE	Variable	User defined Default: ±10%
GD_STATE_CHANGE	Variable	User defined Default: ±10%
HBT_DAMAGE_TIMEOUT_MPLIER (see NOTE 3)	Variable	Default: 2.4
HBT_ESPDU_KIND_CULTURAL_FEATURE	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_KIND_ENVIRONMENTAL	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_KIND_EXPENDABLE	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_KIND_LIFE_FORM	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_KIND_MUNITION	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_KIND_RADIO	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_KIND_SENSOR/EMITTER	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_KIND_SUPPLY	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_PLATFORM_AIR	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_PLATFORM_LAND	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_PLATFORM_SPACE	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_PLATFORM_SUBSURFACE	Variable	Default: 5 s Tolerance: ±10%
HBT_ESPDU_PLATFORM_SURFACE	Variable	Default: 5 s Tolerance: ±10%
HBT_PDU_AGGREGATE_STATE	Variable	Default: 30 s Tolerance: ±10%
HBT_PDU_APPEARANCE	Variable	Default: 60 s Tolerance: ±10%
HBT_PDU_DE_FIRE	Variable	Default: 0.5s Tolerance: ±10%
HBT_PDU_DESIGNATOR	Variable	Default: 5 s Tolerance: ±10%

Table 25—Symbolic names *(continued)*

Symbolic name	Fixed or variable	Fixed and default parameter values
HBT_PDU_EE	Variable	Default: 5 s Tolerance: ±10%
HBT_PDU_ENTITY_DAMAGE	Variable	Default: 10 s Tolerance: ±10%
HBT_PDU_ENVIRONMENTAL_PROCESS	Variable	Default: 15 s Tolerance: ±10%
HBT_PDU_GRIDDED_DATA	Variable	Default: 15 min Tolerance: ±10%
HBT_PDU_IFF	Variable	Default: 10 s Tolerance: ±10%
HBT_PDU_ISGROUPOF	Variable	Default: 5 s Tolerance: ±10%
HBT_PDU_MINEFIELD_DATA	Variable	Default: 5 s Tolerance: ±10%
HBT_PDU_MINEFIELD_STATE	Variable	Default: 5 s Tolerance: ±10%
HBT_PDU_RECEIVER	Variable	Default: 5 s Tolerance: ±10%
HBT_PDU_SEES	Variable	Default: 3 min Tolerance: ±10%
HBT_PDU_TRANSMITTER	Variable	Default: 2 s Tolerance: ±10%
HBT_PDU_TSPI	Variable	Default: 30 s Tolerance: ±10%
HBT_PDU_UA	Variable	Default: 3 min Tolerance: ±10%
HBT_STATIONARY	Variable	Default: 1 min Tolerance: ±10%
HBT_TIMEOUT_MPLIER (see NOTE 2)	Variable	Default: 2.4
HQ_TOD_DIFF_THRSH	Variable	Default: 20 ms
IFF_AZ_THRSH	Variable	Default: 3°
IFF_CHG_LATENCY	Variable	Default: 2 s
IFF_EL_THRSH	Variable	Default: 3°
IFF_IP_REPLY_TIMER	Variable	Default: 30 s
IFF_PDU_FINAL	Variable	Default: 10 s
IFF_PDU_RESUME	Variable	Default: 10 s
IO_UNTIL_FURTHER_NOTICE	Fixed	65 535
MAX_PDU_SIZE_BITS	Fixed	65 536
MAX_PDU_SIZE_OCTETS	Fixed	8192

Table 25—Symbolic names *(continued)*

Symbolic name	Fixed or variable	Fixed and default parameter values
MINEFIELD_CHANGE	Variable	Default: 2.5 s
MINEFIELD_RESPONSE_TIMER	Variable	Default: 1 s
MULTIPLES_PRESENT	Fixed	0
NO_AGG	Fixed	0
NO_APPLIC	Fixed	0
NO_BEAM	Fixed	0
NO_CATEGORY	Fixed	0
NO_EMITTER	Fixed	0
NO_ENTITY	Fixed	0
NO_FIRE_MISSION	Fixed	0
NO_KIND	Fixed	0
NO_OBJECT	Fixed	0
NO_PATTERN	Fixed	0.0
NO_REF_NUMBER	Fixed	0
NO_SITE	Fixed	0
NO_SPECIFIC	Fixed	0
NO_SPECIFIC_ENTITY	Fixed	NO_SITE: NO_APPLIC: NO_ENTITY
NO_SUBCAT	Fixed	0
NO_VALUE	Fixed	0
NON_SYNC_THRSH	Variable	Default: 1 min
POWER_ENGINE_OFF	Fixed	–100.0
POWER_IDLE	Fixed	0.0
POWER_MAX_AFTERBURNER	Fixed	100.0
POWER_MILITARY	Fixed	50.0
POWER_MIN_AFTERBURNER	Fixed	51.0
REPAR_REC_T1	Variable	Default: 5 s
REPAR_SUP_T1	Variable	Default: 12 s
REPAR_SUP_T2	Variable	Default: 12 s
RESUP_REC_T1	Variable	Default: 5 s
RESUP_REC_T2	Variable	Default: 55 s
RESUP_SUP_T1	Variable	Default: 1 min

Table 25—Symbolic names *(continued)*

Symbolic name	Fixed or variable	Fixed and default parameter values
RQST_ASSIGN_ID	Fixed	FFFE (H)
SEES_NDA_THRSH	Variable	Default: ±2° in the axis of deflection
SEES_PS_THRSH	Variable	Default: ±10% of the maximum value of the Power Setting
SEES_RPM_THRSH	Variable	Default: ±5% of the maximum engine speed in RPM
SMALLEST_MTU_OCTETS [NOTE 1]	Variable	Default: 1400 octets for Internet Protocol Version 4 networks
SM_REL_RETRY_CNT	Variable	Default: 3
SM_REL_RETRY_DELAY	Variable	Default: 2 s
TARGET_ID_UNKNOWN	Fixed	NO_SITE: NO_APPLIC: NO_ENTITY
TIMESTAMP_AHEAD	Variable	Default: 5 s
TIMESTAMP_BEHIND	Variable	Default: 5 s
TI_TIMER1_DFLT	Variable	Default: 2 s
TI_TIMER2_DFLT	Variable	Default: 12 s
TO_AUTO_RESPONSE_TIMER	Variable	Default: 5 s
TO_MAN_RESPONSE_TIMER	Variable	Default: 120 s
TR_TIMER1_DFLT	Variable	Default: 5 s
TR_TIMER2_DFLT	Variable	Default: 60 s
TRANS_ORIENT_THRSH	Variable	Default: 180°
TRANS_POS_THRSH	Variable	Default: 500 m
UA_ORIENT_THRSH	Variable	Default: 2°
UA_POS_THRSH	Variable	Default: 10 m
UA_SRPM_ROC_THRSH	Variable	Default: ±10% of maximum rate of change
UA_SRPM_THRSH	Variable	Default: ±5% of maximum shaft rate in RPM
Legend and NOTEs: NOTE 1—See 6.3.4 for additional information concerning the Smallest MTU parameter.		

Table 25—Symbolic names *(continued)*

<table>
<tr><th>Symbolic name</th><th>Fixed or variable</th><th>Fixed and default parameter values</th></tr>
<tr><td colspan="3">
Symbolic Name. The symbolic name used in this standard for a fixed or variable parameter. There is no requirement that a simulation application has to use the exact rendition of the symbolic name shown in the standard, only that it be able to be identified and matched to a specific fixed or variable parameter shown in this table. The symbolic name HBT_<name> is used in the standard to signify heartbeat parameters in general.
Fixed/Variable. Indicates whether the symbolic name is a FIXED value not subject to change or a variable parameter. If a variable parameter, the word DEFAULT indicates that the numeric value is the default value in the absence of an Exercise agreement that has assigned another value.
Default. Indicates if this is a default value. Default values are only applicable to variable parameters.
Fixed and Default Parameter Values. Shows the fixed value for fixed parameters and the default value for variable parameters. When "User-defined" is shown, it indicates that the parameter has multiple state values and a single default value is not applicable. The exercise agreement should define the values to be used.
Tolerance. The tolerance of the actual occurrence of the use of this value. The tolerance value will be in degrees or a percentage that the variable parameter may vary from the required value. The required value will be either the value specified in an exercise agreement or, if unspecified by an exercise agreement, then the default value listed in this table. (See the example below.)
Notations:
s seconds
m meters
m/s meters/second
min minutes
± plus or minus
% percent
° degrees
NOTE 2—The entity timeout parameter is based on taking the specific entity heartbeat parameter and multiplying it by the HBT_TIMEOUT_MPLIER. Example: HBT_ESPDU_PLATFORM_AIR multiplied by HBT_TIMEOUT_MPLIER = the specific entity timeout parameter for air platform entities.
NOTE 3—The entity damage status timeout parameter is calculated by multiplying HBT_PDU_ENTITY_DAMAGE by HBT_DAMAGE_TIMEOUT_MPLIER.
</td></tr>
</table>

6.2 Basic data types and records

6.2.1 General

This subclause specifies requirements for basic data types and records. Enumeration values are given in SISO-REF-010 unless otherwise stated.

6.2.2 Acoustic Emitter System record

Information about a particular UA emitter shall be represented using an Acoustic Emitter System record. This record shall consist of three fields: Acoustic System Name, Function, and Acoustic ID Number. These fields are described below:

a) *Acoustic System Name*. This field shall specify the system for a particular UA emitter. This field shall be represented by a 16-bit enumeration (see [UID 144]).

b) *Function*. This field shall describe the function of the acoustic system. This field shall be represented by an 8-bit enumeration (see [UID 145]).

c) *Acoustic ID Number*. This field shall specify the UA emitter identification number relative to a specific system. This field shall be represented by an 8-bit unsigned integer. This field allows the differentiation of multiple systems on an entity, even if in some instances two or more of the systems may be identical UA emitter types. Numbering of systems shall begin with the value 1.

The format of the Acoustic Emitter System record shall be as shown in Table 26.

Table 26—Acoustic Emitter System record

Field size (bits)	Field name	Data type
16	Acoustic System Name	16-bit enumeration
8	Function	8-bit enumeration
8	Acoustic ID Number	8-bit unsigned integer
Total Acoustic Emitter System record size = 32 bits		

6.2.3 Aggregate Identifier record

The unique designation of each aggregate in an exercise shall be specified by an Aggregate Identifier record. The Aggregate Identifier is not an entity and shall not be treated as such. A subaggregate within an aggregation shall also be uniquely identified using the Aggregate Identifier record. The Aggregate State PDU may contain a list of entities for which Entity State PDUs are separately being issued. In this case, those entities are designated using Entity Identifier records.

The Aggregate Number shall not contain ALL_AGGS, NO_AGG or RQST_ASSIGN_ID.

The Aggregate Identifier is an object identifier. General object identifier requirements are specified in 4.2.5.2.

The format of the Aggregate Identifier record shall be as shown in Table 27.

Table 27—Aggregate Identifier record

<table>
<tr><th>Field size (bits)</th><th>Field name</th><th>Data type</th></tr>
<tr><td rowspan="2">32</td><td rowspan="2">Simulation Address</td><td>Site Number—16-bit unsigned integer</td></tr>
<tr><td>Application Number—16-bit unsigned integer</td></tr>
<tr><td>16</td><td>Aggregate Number</td><td>16-bit unsigned integer</td></tr>
<tr><td colspan="3">Total Aggregate Identifier record size = 48 bits</td></tr>
</table>

6.2.4 Aggregate Marking record

Aggregate marking shall be specified by the Aggregate Marking record. This record shall specify the character set used in the marking and the string of characters to be interpreted for display. The character set shall be specified by an 8-bit enumeration (see [UID 45]). The string of characters shall be represented by a 31-element character string. This string shall begin with the most significant octet located at the lowest address. The designation of unused characters shall be specified by the character set. The format of the Aggregate Marking record shall be as shown in Table 28.

Table 28—Aggregate Marking record

Field size (bits)	Field name	Data type
8	Character set	8-bit enumeration
8	1st Character	8-bit unsigned integer
8	2nd Character	8-bit unsigned integer
	• • •	
8	31st Character	8-bit unsigned integer
Total Aggregate Marking record size = 256 bits		

6.2.5 Aggregate Type record

The type of aggregate shall be specified by an Aggregate Type record. The Aggregate Type record indicates the type and organization of an aggregate. It shall contain the following fields:

a) *Aggregate Kind.* This field shall identify the grouping criterion used to group the aggregate. This field shall be represented by an 8-bit enumeration indicating the common characteristic of the aggregate (see [UID 206]).

b) *Domain.* This field shall specify the domain in which the aggregate operates. This field shall be represented by an 8-bit enumeration (see [UID 8]). If a single domain does not apply, then the domain field shall contain the number zero.

c) *Country.* This field shall specify the country with which the aggregate is associated. This field shall be represented by a 16-bit enumeration (see [UID 29]). If the aggregate is not associated with any country, then the country field shall contain the number zero.

d) *Category.* This field shall specify the category that describes the aggregate. This field shall be represented by an 8-bit enumeration (see [UID 207]). The enumerations for this field depend on the Aggregate Kind field.

e) *Subcategory.* This field shall specify the subcategory to which the aggregate belongs based on the Category field. This field shall be represented by an 8-bit enumeration (see [UID 208]).

f) *Specific.* This field shall identify specific information about the aggregate based on the Subcategory field. This field shall be represented by an 8-bit enumeration (see [UID 209]).

g) *Extra.* This field shall identify extra information needed to describe the aggregate. This field shall be represented by an 8-bit enumeration. The enumerations for this field depend on the type of aggregate represented.

The format of the Aggregate Type record shall be as shown in Table 29.

Table 29—Aggregate Type record

Field size (bits)	Field name	Data type
8	Aggregate Kind	8-bit enumeration
8	Domain	8-bit enumeration
16	Country	16-bit enumeration
8	Category	8-bit enumeration
8	Subcategory	8-bit enumeration
8	Specific	8-bit enumeration
8	Extra	8-bit enumeration
Total Aggregate Type record size = 64 bits		

6.2.6 Angle representation

Angles shall be specified as 32-bit floating point numbers expressed in radians.

6.2.7 Angular Velocity Vector record

The angular velocity of simulated entities shall be represented by the Angular Velocity Vector record. This record shall specify the rate at which an entity's orientation is changing. The angular velocity shall be measured in radians per second measured about each of the entity's own coordinate axes. The record shall consist of three fields. The first field shall represent velocity about the x-axis, the second about the y-axis, and the third about the z-axis [see item a) in 6.2.96]. The positive direction of the angular velocity is defined by the right-hand rule. The format of the Angular Velocity Vector record shall be shown as in Table 30.

Table 30—Angular Velocity Vector record

Field size (bits)	Field name	Data type
32	Rate about x-axis	32-bit floating point
32	Rate about y-axis	32-bit floating point
32	Rate about z-axis	32-bit floating point
Total Angular Velocity Vector record size = 96 bits		

6.2.8 Antenna Pattern record

6.2.8.1 General

The following subclause describes antenna pattern records. The total length of each record shall be a multiple of 64 bits.

6.2.8.2 Beam Antenna Pattern record

This record specifies the direction, pattern, and polarization of radiation from a radio transmitter's antenna. The radiation pattern produced by the antenna is defined with respect to a beam coordinate system. The beam coordinate system is a right-handed Cartesian coordinate system with the center of the beam along the *x*-axis of the system (see Figure 42). In the Transmitter PDU, the origin of the beam coordinate system is specified by the Antenna Location and may also be specified by the Relative Antenna Location fields. The Beam Antenna Pattern record contains the following fields:

a) *Beam Direction.* This field shall specify the rotation that transforms the reference coordinate system into the beam coordinate system. Either world coordinates or entity coordinates (see 1.6.3) may be used as the reference coordinate system. The reference coordinate system in use shall be specified by the Reference System field of the Beam Antenna Pattern record. The Beam Direction field shall be represented as an Euler Angles record (see 6.2.32).

b) *Azimuth Beamwidth.* This field shall specify the full width of the beam to the –3 dB power density points in the x–y plane of the beam coordinate system. Azimuth beamwidth shall be represented by a 32-bit floating point number in units of radians.

c) *Elevation Beamwidth.* This field shall specify the full width of the beam to the –3 dB power density points in the x–z plane of the beam coordinate system. Elevation beamwidth shall be represented by a 32-bit floating point number in units of radians.

d) *Reference System.* This field shall specify the reference coordinate system with respect to which beam direction is specified. The reference system shall be represented as an 8-bit enumeration (see [UID 168]). The value of this field shall not change over the duration of an exercise. Different radios in the same exercise may use different reference systems. The world coordinate system is the preferred reference system for training applications. The entity coordinate system should only be used where highly directional antennas have to be precisely modeled. Use of the entity coordinates for a reference system requires the receipt of Entity State PDUs by radio simulation applications and may adversely affect their performance. Support for an entity coordinate reference system is an optional part of this standard.

e) E_Z. This field shall specify the magnitude of the Z-component (in beam coordinates) of the Electrical field at some arbitrary single point in the main beam and in the far field of the antenna. E_Z shall be represented by a 32-bit floating point number.

f) E_X. This field shall specify the magnitude of the X-component (in beam coordinates) of the Electrical field at some arbitrary single point in the main beam and in the far field of the antenna. E_X shall be represented by a 32-bit floating point number.

g) *Phase.* This field shall specify the phase angle between E_Z and E_X in radians.

The format of the Beam Antenna Pattern record shall be as shown in Table 31.

Table 31—Beam Antenna Pattern record

Field size (bits)	Field name	Data type
96	Beam Direction	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
32	Azimuth Beamwidth	32-bit floating point

Table 31—Beam Antenna Pattern record *(continued)*

Field size (bits)	Field name	Data type
32	Elevation Beamwidth	32-bit floating point
8	Reference System	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	E_Z	32-bit floating point
32	E_X	32-bit floating point
32	Phase	32-bit floating point
32	Padding	32 bits unused
Total Beam Antenna Pattern record size = 320 bits		

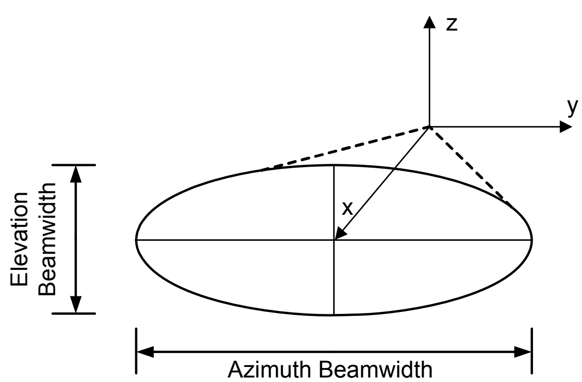


Figure 42—Beam coordinates

6.2.9 Association record

An entity's associations with other entities and/or locations shall be represented by one or more Association records. For each association, this record shall specify the type of the association, the associated entity's Entity ID, and/or the associated location's world coordinates.

This record may be used (optionally) in a transfer transaction to send internal state data from the divesting simulation to the acquiring simulation (see 5.9.4). This record may also be used for other purposes.

In a transfer transaction, this record shall associate either an entity or a location with the entity being transferred, but not both. If more than one entity or location is associated with the transferred entity, then multiple Association records may be used.

The Association record shall contain the following fields:

a) *Association Type.* This field shall indicate the type of association. It shall be represented by an 8-bit enumeration (see [UID 330]).

b) *Padding.* This is an 8-bit padding field.

c) *Associated Entity ID.* This field shall identify the associated entity. If there is no associated entity, this field shall be set to NO_SPECIFIC_ENTITY. This field shall be represented by an Entity Identifier record (see 6.2.28).

d) *Associated Location in World Coordinates.* This field shall specify the associated location in world coordinates. If there is no associated location, this field shall be set to 0.0, 0.0, 0.0 (all bits set to zero). This field shall be represented by a World Coordinates record (see 6.2.98).

The format of the Association record shall be as shown in Table 32.

Table 32—Association record

Field size (bits)	Field name	Data type
8	Association Type	8-bit enumeration
8	Padding	8 bits unused
48	Associated Entity ID	Entity Identifier record (see 6.2.28)
192	Associated Location in World Coordinates	World Coordinates record (see 6.2.98)
Total Association record size = 256 bits		

6.2.10 Attribute record

The Attribute record shall be used to convey information for one or more attributes. Attribute records shall conform to the standard variable record format of the Standard Variable Specification record (see 6.2.83). The fields of the Attribute record are as follows:

a) *Record Type*. This field shall specify the record type for this Attribute record and shall be represented by a 32-bit enumeration.

b) *Record Length*. This field shall specify the total length of the Attribute record in octets. The value of the Attribute Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 64-bit boundary. The Attribute Record Length value shall be a multiple of 8. This field shall be represented by a 16-bit unsigned integer.

c) *Record-Specific fields*. The attribute data format conforming to that specified by the Record Type.

d) *Padding*. Zero to seven octets of padding as necessary to align the end of the record on a 64-bit boundary. The record alignment requirement may be achieved by placing padding fields anywhere in the Attribute record as deemed appropriate, not necessarily at the end of the record.

The format of the Attribute record shall be as shown in Table 33.

Table 33—Attribute record

Field size (bits)	Field name	Data type
32	Record Type	32-bit enumeration
16	Record Length = $6 + K + P$	16-bit unsigned integer
$8K$	Record-Specific fields	K octets
$8P$	Padding	Padding to 64-bit boundary—P octets
Total Attribute record size = $48 + 8K + 8P$ bits where K is the length of the Record-Specific field in octets P is the number of padding octets, which is $\lceil (6 + K)/8 \rceil 8 - (6 + K)$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

6.2.11 Beam Data record

The specification of beam-specific data necessary to describe the scan volume of an emitter beam shall be communicated in a Beam Data record. (See Figure 43.)

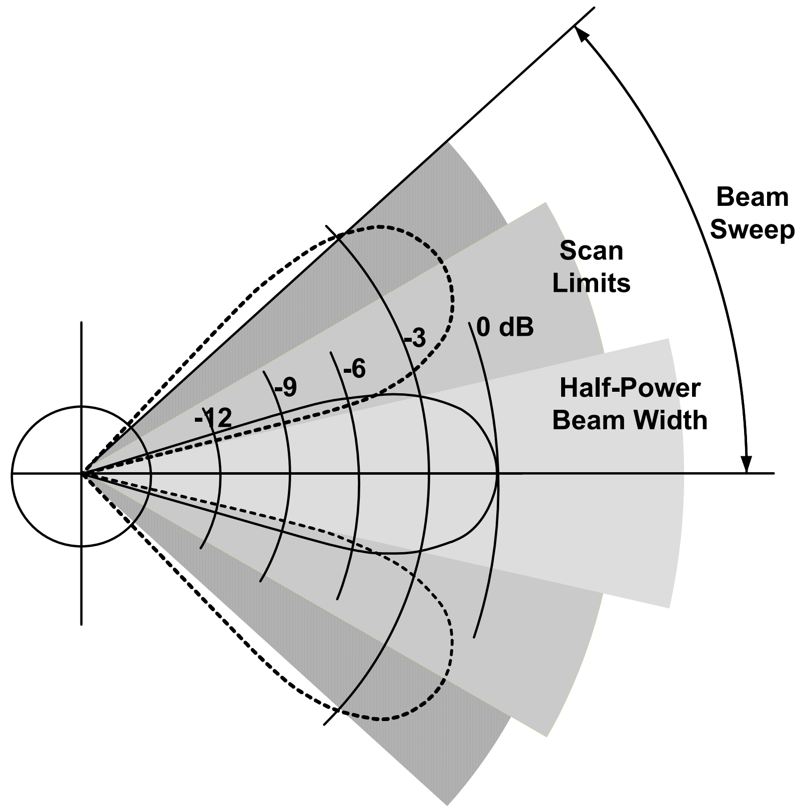


Figure 43—Beam sweep

The fields of this record are as follows:

a) *Beam Azimuth Center.* This field, in conjunction with the fields of item b) through item d) below, shall specify the beam azimuth and elevation centers and the corresponding half-angles of the beam azimuth and elevation sweep required to describe the scan volume covered by the emitter beam main lobe. Sweep half-angles describe half of the full sweep, that is, the angle from the center to either edge of the beam sweep, including 3 dB half-power beam width. This scan volume does not include energy emitted in antenna side lobes. (See Figure 44.)

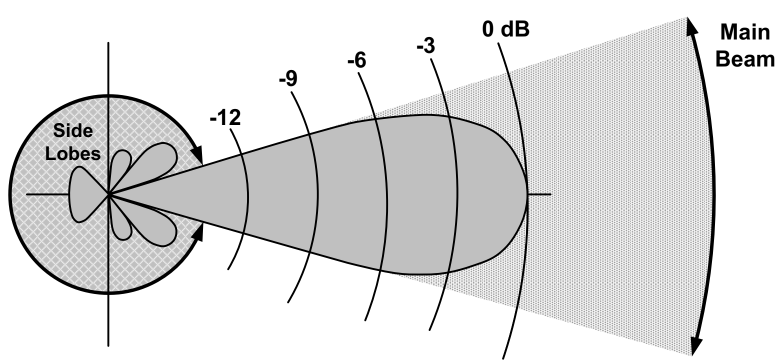


Figure 44—Side lobes

All angles shall be measured in relation to the emitter coordinate system. The azimuth center for 2π radians (360°) scan systems shall be 0, and the azimuth sweep shall be π radians (180°). It should be noted that the scan volume described does not take into account masking by the entity. Masking determinations are a part of the regeneration process and require appropriate database information be processed by the receiving entity. This field and the three fields of item b) through item d) below shall be represented by 32-bit floating point numbers representing radians.

b) *Beam Azimuth Sweep.* See Beam Azimuth Center [item a) above].

c) *Beam Elevation Center.* See Beam Azimuth Center [item a) above].

d) *Beam Elevation Sweep.* See Beam Azimuth Center [item a) above].

e) *Beam Sweep Sync.* This field is provided to allow a receiver to synchronize its regeneration scan pattern to that of the emitter. When employed, this field shall specify, in the semi-open range [0.0, 100.0), the percentage of the pattern scanned from its origin. Note that this semi-open range includes the start value but not the end value. This field shall be represented by a 32-bit floating point number.

The format of the Beam Data record shall be as shown in Table 34.

Table 34—Beam Data record

Field size (bits)	Field name	Data type
32	Beam Azimuth Center	32-bit floating point
32	Beam Azimuth Sweep	32-bit floating point
32	Beam Elevation Center	32-bit floating point
32	Beam Elevation Sweep	32-bit floating point
32	Beam Sweep Sync	32-bit floating point
Total Beam Data record size = 160 bits		

6.2.12 Beam Status record

Information related to the status of a beam shall be represented by the Beam Status record. This record is contained in the Beam Status field of the EE PDU. This record shall contain the following fields:

a) *Beam State*. This 1-bit enumeration field (see [UID 318]) shall indicate whether the beam is Active (0) or Deactivated (1). An active beam is emitting, and a deactivated beam is no longer emitting.

b) *Padding*. Padding to 8 bits.

The format of the Beam Status record shall be as shown in Table 35.

Table 35—Beam Status record

Field name	Bit	Data type
Beam State	0	1-bit enumeration
Padding	1 to 7	7 bits unused
Total Beam Status record size = 8 bits		

6.2.13 Change/Options record

The Change/Options record shall be used to communicate the reasons for the issuance of an IFF PDU and to provide the status of various options that may be associated with an interrogator or transponder. Some of the fields are common to all system types, and some only apply to certain system types. For this reason, the Change/Options record is described in B.2.4.

6.2.14 Clock Time record

Time measurements that surpass 1 h shall be represented by a Clock Time record. The fields of this record are as follows:

a) *Hours*. This field shall specify the hours since 0000 h 1 January 1970 UTC. This field shall be represented by a 32-bit integer.

b) *Time Past the Hour*. This field shall specify the time past the hour indicated in the Hour field. This field shall be represented by a Timestamp (see 6.2.88).

The reference time represented shall be either Absolute or Relative Time (see Annex G). The Clock Time record is used in the Real-World and Simulation Time fields of the Start/Resume and Stop/Freeze PDUs and other PDUs as required. The format of the Clock Time record shall be as shown in Table 36.

Table 36—Clock Time record

Field size (bits)	Field name	Data type
32	Hour	32-bit integer
32	Time Past the Hour	Timestamp—32-bit unsigned integer
Total Clock Time record size = 64 bits		

6.2.15 Damage Description records

6.2.15.1 General

Detailed damage sustained by an entity shall be communicated by specific Damage Description records in the Entity Damage Status PDU. Damage Description records shall use the Standard Variable record format of the Standard Variable Specification record (see 6.2.83). Each Damage Description record is defined in this subclause. New Damage Description records may be defined at some future date as needed.

6.2.15.2 Directed Energy Damage Description record

Detailed damage sustained by an entity due to directed energy shall be communicated by the Directed Energy Damage Description record. The location of the damage shall be based on a relative x, y, z location from the center of the entity. The Directed Energy Damage Description record shall contain the following fields:

a) *Record Type*. This field shall identify this record as a Directed Energy Damage Description record. It shall be represented by a 32-bit enumeration.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a

64-bit boundary. The Record Length shall be a multiple of 8. The Record Length shall be represented by a 16-bit unsigned integer.

c) *Damage Location*. This field shall identify the location of the centroid of the damage location and shall be represented by an Entity Coordinate Vector (see 6.2.96) expressed in units of meters.

d) *Damage Diameter*. This field shall identify the size of the damaged location in units of meters. It shall be represented as a 32-bit floating point number.

e) *Temperature*. This field shall identify the average temperature of the damaged region in units of degrees Celsius. If the firing entity does not model or otherwise cannot provide this information, a value of –273.15 °C shall be used. It shall be represented as a 32-bit floating point number.

f) *Component Identification*. This field shall be specified by an 8-bit enumeration (see [UID 314]).

g) *Component Damage Status*. This field shall be specified by an 8-bit enumeration (see [UID 315]).

h) *Component Visual Damage Status*. This field shall be specified by an 8-bit record (see [UID 317]).

i) *Component Visual Smoke Color*. This field shall be specified by an 8-bit enumeration (see [UID 316]).

j) *Fire Event ID*. For any component damage resulting from the receipt of a DIS PDU (e.g., a DE Fire PDU or Detonation PDU), this field shall be set to the Fire Event ID from that PDU. If no Fire Event Identifier is applicable, this field shall be set to all zeroes. This field shall be represented by an Event Identifier record (see 6.2.33).

The format of the Directed Energy Damage Description record shall be as shown in Table 37.

Table 37—Directed Energy Damage Description record

Field size (bits)	Field name	Data type
32	Record Type = 4500	32-bit enumeration
16	Record Length = 40	16-bit unsigned integer
16	Padding	16 bits unused
96	Damage Location	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
32	Damage Diameter	32-bit floating point
32	Temperature	32-bit floating point
8	Component Identification	8-bit enumeration
8	Component Damage Status	8-bit enumeration
8	Component Visual Damage Status	8-bit record
8	Component Visual Smoke Color	8-bit enumeration
48	Fire Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
16	Padding	16 bits unused
Total Directed Energy Damage Description record size = 320 bits		

6.2.16 Data Filter record

The Data Filter record shall be used to identify which of the optional data fields are contained in the Minefield Data PDU or requested in the Minefield Query PDU. This is a 32-bit record. For each field, true denotes that the data is requested or present and false denotes that the data is neither requested nor present. Usage of the Data Filter record in the Minefield Data PDU is described in 5.10.4.2 and 7.9.4. Usage of the Data Filter record in the Minefield Query PDU is described in 5.10.3.2 and 7.9.3. The fields are described below and the format of the Data Filter record shall be as shown in Table 38:

a) *Ground Burial Depth Offset*. Identifies whether Ground Burial Depth Offset data is requested or present. If true, then the Mine Orientation data shall also be sent.
b) *Water Burial Depth Offset*. Identifies whether Water Burial Depth Offset data is requested or present. If true, then the Ground Burial Depth Offset and Mine Orientation data shall also be sent.
c) *Snow Burial Depth Offset*. Identifies whether Snow Burial Depth Offset data is requested or present. If true, then the Ground Burial Depth Offset and Mine Orientation data shall also be sent.
d) *Mine Orientation*. Identifies whether Mine Orientation data is requested or present. If any of the burial offset data is present, then the Mine Orientation shall also be sent.
e) *Thermal Contrast*. Identifies whether Thermal Contrast data is requested or present.
f) *Reflectance*. Identifies whether Reflectance data is requested or present.
g) *Mine Emplacement Time*. Identifies whether Mine Emplacement Time data is requested or present.
h) *Trip/Detonation Wire*. Identifies whether Trip/Detonation Wire data is requested or present. Within the Minefield Data PDU, these are the Number of Trip/Detonation Wires, Number of Vertices, and Vertex fields.
i) *Fusing*. Identifies whether Fusing data is requested or present.
j) *Scalar Detection Coefficient*. Identifies whether Scalar Detection Coefficient data is requested or present.
k) *Paint Scheme*. Identifies whether Paint Scheme data is requested or present.

Table 38—Data Filter record

Field name	Bit	Data type
Ground Burial Depth Offset	0	Boolean
Water Burial Depth Offset	1	Boolean
Snow Burial Depth Offset	2	Boolean
Mine Orientation	3	Boolean
Thermal Contrast	4	Boolean
Reflectance	5	Boolean
Mine Emplacement Time	6	Boolean
Trip/Detonation Wire	7	Boolean
Fusing	8	Boolean
Scalar Detection Coefficient	9	Boolean
Paint Scheme	10	Boolean
Padding	11–31	Unused
Total Data Filter record size = 32 bits		

6.2.17 Data Query Datum Specification record

The identification of a list of Fixed and Variable Datum records being queried shall be specified using the Data Query Datum Specification record. This record shall specify the number and identification of records requested. The fields of this record are as follows:

a) *Number of Fixed Datums.* This field shall specify the number of Fixed Datum fields and shall be represented by a 32-bit unsigned integer.

b) *Number of Variable Datums.* This field shall specify the number of Variable Datum fields and shall be represented by a 32-bit unsigned integer.

c) *Fixed Datum ID.* This field shall be represented by a 32-bit enumeration (see [UID 66]).

d) *Variable Datum ID.* This field shall be represented by a 32-bit enumeration (see [UID 66]).

The format of the Data Query Datum Specification record shall be shown as in Table 39.

Table 39—Data Query Datum Specification record

Field size (bits)	Field name	Data type
32	Number of Fixed Datums (*N*)	32-bit unsigned integer
32	Number of Variable Datums (*M*)	32-bit unsigned integer
32	Fixed Datum #1	Datum ID—32-bit enumeration
	• • •	
32	Fixed Datum #*N*	Datum ID—32-bit enumeration
32	Variable Datum #1	Variable Datum ID—32-bit enumeration
	• • •	
32	Variable Datum #*M*	Variable Datum ID—32-bit enumeration
Total Data Query Datum Specification record size = 64 + 32*N* + 32*M* bits where *N* is the number of fixed datums *M* is the number of variable datums		

6.2.18 Datum Specification record

The Datum Specification record is used to convey fixed and variable datum information. Variable datums are useful to supply database names, character strings, bit streams, and so on. Fixed datums are more efficient for conveying information requiring a single number.

Any datum in a Datum Specification record containing a single numeric value that is 32 bits or less in length shall be conveyed by a Fixed Datum record. Only one fixed datum is allowed per Fixed Datum Value field.

A Variable Datum record shall be used for all data values in a Datum Specification record that are over 32 bits in length or require more than one numeric value. All data transmitted by character strings in a Datum Specification record shall be variable datums. Only one variable datum is allowed per Variable Datum field.

The fields of this record are as follows:

a) *Number of Fixed Datum Records*. This field shall specify the number of Fixed Datum records and shall be represented by a 32-bit unsigned integer.
b) *Number of Variable Datum Records*. This field shall specify the number of Variable Datum records and shall be represented by a 32-bit unsigned integer.
c) *Fixed Datum records*. These fields shall contain zero or more Fixed Datum records (see 6.2.37).
d) *Variable Datum records*. These fields shall contain zero or more Variable Datum records (see 6.2.93).

The format of the Datum Specification record shall be as shown in Table 40.

Table 40—Datum Specification record

Field size (bits)	Field name	Data type
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
64	Fixed Datum record #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum record #*N*	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum record #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value —K_1 bits
		Padding to 64-bit boundary—P_1 bits
	• • •	

Table 40—Datum Specification record *(continued)*

Field size (bits)	Field name	Data type
$64 + K_M + P_M$	Variable Datum record #*M*	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Datum Specification record size = $64 + 64N + \sum_{i=1}^{M}(64 + K_i + P_i)$ bits

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is length of variable datum value i in bits
P_i is is the number of padding bits for Variable Datum Record #i, which is $\lceil K_i/64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

6.2.19 Descriptor records

6.2.19.1 General

The Descriptor field is found in the Fire and Detonation PDU. It provides additional characteristics related to something that is fired or launched, or that detonates, bursts, or ignites. There are three types of descriptor records. Each descriptor record type has the same format whether it is included in the Fire or Detonation PDU:

a) *Fire PDU*. Two descriptor records are applicable to the Fire PDU. The appropriate descriptor record shall be included in each Fire PDU. The Munition Descriptor record describes characteristics of a munition to support visual and sensor models. The Expendable Descriptor record describes characteristics of an expendable, such as chaff and flares, including both countermeasure and signal flares, to support visual and sensor models.

b) *Detonation PDU*. Three descriptor records are applicable to the Detonation PDU. The appropriate descriptor record shall be included in each Detonation PDU. The Munition Descriptor record describes characteristics of the detonation of a munition. The Explosion Descriptor record describes an explosion, either a non-munition explosion or a munition that has exploded for a reason other than as a result of its intended use. The Expendable Descriptor record is intended to be used to provide information about the bursting of chaff or the ignition of a flare. If additional information is desired to be conveyed about a detonation or explosion, one or more VP records may also be included in the VP record section of the Detonation PDU.

6.2.19.2 Munition Descriptor record

The firing or detonation of a munition shall be represented by a Munition Descriptor record. A munition, for the purpose of this record, shall be defined as any munition type for which the Kind field of the Entity Type record is set to Munition (2). This includes both detonations as a result of its intended use (e.g., firing a missile at a target) as well as unintended-use detonations such as the following: an accidental explosion; an explosion due to a demolition event or an explosive ordnance disposal (EOD) activity; or a secondary explosion caused by some other munition or non-munition explosion.

The Munition Descriptor record is applicable to the Fire PDU and Detonation PDU. The fields of this record are as follows:

a) *Munition Type*. This field shall be specified by an Entity Type record (see 6.2.30).
b) *Warhead*. This field shall specify the type of warhead. It shall be represented by a 16-bit enumeration (see [UID 60]).
c) *Fuse*. This field shall indicate the fusing. It shall be represented by a 16-bit enumeration (see [UID 61]).
d) *Quantity*. This field shall represent the number of rounds fired in the burst or the number of munitions simultaneously launched. It shall be represented by a 16-bit unsigned integer.
e) *Rate*. This field shall indicate the rate of fire such as the rounds per minute for a munition. If the Quantity field is equal to one, the Rate field shall contain zero. It shall be represented by a 16-bit unsigned integer.

The format of the Munition Descriptor record shall be as shown in Table 41.

Table 41—Munition Descriptor record

Field size (bits)	Field name	Data type
64	Munition Type	Entity type—64-bit Entity Type record
16	Warhead	16-bit enumeration
16	Fuse	16-bit enumeration
16	Quantity	16-bit unsigned integer
16	Rate	16-bit unsigned integer
Total Munition Descriptor record size = 128 bits		

6.2.19.3 Explosion Descriptor record

The explosion of a non-munition shall be represented by an Explosion Descriptor record. The exploding object may be an entity for which an Entity State PDU has been issued or may be an articulated or attached part of an entity. If it is necessary to provide more details of the characteristics of the explosion, one or more VP records may be developed and included in the VP record section of the Detonation PDU to convey such information.

NOTE—Any munition that explodes, whether for its intended purpose or not, is represented by the Munition Descriptor record and not by an Explosion Description record.

The Explosion Descriptor record is applicable to the Detonation PDU. The fields of this record shall be as follows:

a) *Exploding Object Type*. This field shall indicate the type of object that exploded. It shall be represented by an Entity Type record (see 6.2.30).
b) *Explosive Material*. This field shall indicate the material that exploded. This may be a material that may explode under certain conditions (e.g., gasoline or grain dust) as well as a material that is

intended to cause an explosion (e.g., TNT). It shall be represented by a 16-bit enumeration (see [UID 310]).

c) *Explosive Force*. This field shall indicate the explosive force expressed as the equivalent kilograms of TNT (4.184×10^6 Joules per kilogram). It shall be represented as a 32-bit floating point number.

The format of the Explosion Descriptor record shall be as shown in Table 42.

Table 42—Explosion Descriptor record

Field size (bits)	Field name	Data type
64	Exploding Object Type	64-bit Entity Type record
16	Explosive Material	16-bit enumeration
16	Padding	16 bits unused
32	Explosive Force	32-bit floating point
Total Explosion Descriptor record size = 128 bits		

6.2.19.4 Expendable Descriptor record

The burst of a chaff expendable or ignition of a flare shall be represented by an Expendable Descriptor record. The Expendable Descriptor record is applicable to the Fire and Detonation PDU. The fields of this record are as follows:

a) *Expendable Type*. This field shall indicate the type of expendable. It shall be represented by an Entity Type record (6.2.30) and shall be identical to the Expendable Type in the associated Fire PDU.

b) *Padding*. This field shall consist of 64 bits of padding.

The format of the Expendable Descriptor record shall be as shown in Table 43.

Table 43—Expendable Descriptor record

Field size (bits)	Field name	Data type
64	Expendable Type	64-bit Entity Type record
64	Padding	64 bits unused
Total Expendable Descriptor record size = 128 bits		

6.2.20 Directed Energy (DE) records

6.2.20.1 General

Directed Energy (DE) records shall be used to communicate additional information about a directed energy weapon system and to describe precise directed energy beam targeting information to other applications. DE records use the variable record format specified in the Standard Variable Specification record (see 6.2.83).

Each DE record is defined in this subclause. New DE records may be defined at some future date as the characteristics of directed energy weapons evolve.

6.2.20.2 DE Area Aimpoint record

The DE Area Aimpoint record shall be used to communicate targeting information when the target of the directed energy weapon is an area. The area may or may not be associated with one or more target entities.

The fields of the DE Area Aimpoint record are as follows:

a) *Record Type*. This field shall identify this record as a DE Area Aimpoint record. It shall be represented by a 32-bit enumeration.

b) *Record Length*. This field shall indicate the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 64-bit boundary. The Record Length shall be a multiple of 8. The Record Length shall be represented by a 16-bit unsigned integer.

c) *Beam Antenna Pattern Record Count*. This field shall contain the count of Beam Antenna Pattern records that are included as part of this record. This value shall be represented by a 16-bit unsigned integer.

d) *DE Target Energy Deposition Record Count*. This field shall contain the count of DE Target Energy Deposition records that are included as part of this record. This value shall be represented by a 16-bit unsigned integer.

e) *Beam Antenna Pattern records*. This field shall contain one or more non-self-identifying Beam Antenna Pattern records (see 6.2.8.2).

f) *DE Target Energy Deposition records*. This field shall contain zero or more non-self-identifying DE Target Energy Deposition records (see 6.2.20.4).

g) *Padding*. Padding shall be inserted, as required, to reach a 64-bit boundary.

The format of the DE Area Aimpoint record shall be as shown in Table 44.

Table 44—DE Area Aimpoint record

Field size (bits)	Field name	Data type
32	Record Type = 4001	32-bit enumeration
16	Record Length	16-bit unsigned integer
16	Padding	16 bits unused
16	Beam Antenna Pattern Record Count (*N*)	16-bit unsigned integer
16	DE Target Energy Deposition Record Count (*M*)	16-bit unsigned integer
320	Beam Antenna Pattern record #1	320 bits (see 6.2.8.2)
	⋮	
320	Beam Antenna Pattern record #*N*	320 bits (see 6.2.8.2)
96	DE Target Energy Deposition record #1	96 bits (see 6.2.20.4)

Table 44—DE Area Aimpoint record *(continued)*

Field size (bits)	Field name	Data type
⋮		
96	DE Target Energy Deposition record #M	96 bits (see 6.2.20.4)
$8P$	Padding	Padding to 64-bit boundary—P octets

Total DE Area Aimpoint record size = $96 + 320N + 96M + 8P$ bits

where

N is the number of Beam Antenna Pattern records
M is the number of DE Target Energy Deposition records
P is the number of padding octets, which is $4(\lceil M/2 \rceil 2 - M)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

6.2.20.3 DE Precision Aimpoint record

The DE Precision Aimpoint record shall be used to communicate targeting information when the target of the directed energy weapon is not an area but a specific target entity. Use of this record assumes that the DE weapon would not fire unless a target is known and is currently tracked. Therefore, the Target Entity ID field shall be filled with the entity ID of the known target. This record provides information to allow a firing entity to describe precise directed energy beam targeting information to other applications. Refer to Annex A for a detailed description of the reference coordinate frames employed in this record.

The fields of the DE Precision Aimpoint record are as follows:

a) *Record Type*. This field shall identify this record as a DE Precision Aimpoint record. It shall be represented by a 32-bit enumeration.

b) *Record Length*. This field shall indicate the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 64-bit boundary. The Record Length shall be a multiple of 8. The Record Length shall be represented by a 16-bit unsigned integer.

c) *Target Spot World Location*. This field shall contain the instantaneous location of the target spot in world coordinates. Units are meters. This field shall be represented by a World Coordinates record (see 6.2.98).

d) *Target Spot Entity Location*. This field shall contain the instantaneous location of the target spot in the target entity's coordinate frame; units shall be expressed in meters. This value shall be represented by an Entity Coordinate vector (see 6.2.96).

e) *Target Spot Velocity*. This field shall contain the instantaneous target spot velocity. The reference frame used for this field shall be the entity coordinate frame. Units are meters per second. This field shall be represented by a Linear Velocity Vector record (see 6.2.96).

f) *Target Spot Acceleration*. This field shall contain the instantaneous target spot acceleration. The reference frame used for this field shall be the entity coordinate frame. Units are meters per second per second. This field shall be represented by a Linear Acceleration Vector record (see 6.2.96).

g) *Target Entity ID*. This field shall identify the entity ID for the target affected by the weapon. This field shall be represented by an Entity Identifier record (see 6.2.28).

h) *Target Component Identifier*. This field shall identify the entity component that is being targeted by the DE weapon. This field shall be specified by an 8-bit enumeration. This field shares the same enumeration values as the Component Identification field of a Directed Energy Damage Description record.

i) *Beam Spot Type*. This field shall be specified by an 8-bit enumeration (see [UID 311]). The beam spot type specifies how the DE precision weapon energy is distributed on the target. The beam spot type can be either Gaussian, Top Hat, or other, depending on the model chosen.

NOTE—The shape of the beam spot also affects energy distribution and is described by the Beam Spot Cross-Section fields. [See item j) through item l) below.]

j) *Beam Spot Cross-Section Semi-Major Axis*. This field shall identify the length of the semi-major axis of an ellipse approximating the irradiance region arriving at the target (see Annex A). Ellipse dimensions are measured in the plane perpendicular to the direction of the beam (units in meters). This field shall be represented by a 32-bit floating point value.

k) *Beam Spot Cross-Section Semi-Minor Axis*. This field shall identify the length of the semi-minor axis of an ellipse approximating the irradiance region arriving at the target (see Annex A). Ellipse dimensions are measured in the plane perpendicular to the direction of the beam (units in meters). This field shall be represented by a 32-bit floating point value.

l) *Beam Spot Cross-Section Orientation Angle*. This field shall identify the orientation angle from the local "down" vector to the semi-major axis of an ellipse approximating the irradiance region arriving at the target (see Annex A). Ellipse orientation is measured in the plane perpendicular to the direction of the beam (units in radians). This field shall be represented by a 32-bit floating point value.

m) *Peak Irradiance*. This field shall identify the current peak irradiance of emissions in units of Watts per square meter and shall be represented by a 32-bit floating point number.

The format of the DE Precision Aimpoint record shall be as shown in Table 45.

Table 45—DE Precision Aimpoint record

Field size (bits)	Field name	Data type
32	Record Type =4000	32-bit enumeration
16	Record Length = 96	16-bit unsigned integer
16	Padding	16 bits unused
192	Target Spot World Location	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
96	Target Spot Entity Location	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
96	Target Spot Velocity	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point

Table 45—DE Precision Aimpoint record *(continued)*

Field size (bits)	Field name	Data type
96	Target Spot Acceleration	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
48	Target Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Target Component Identifier	8-bit enumeration
8	Beam Spot Type	8-bit enumeration
32	Beam Spot Cross-Section Semi-Major Axis	32-bit floating point
32	Beam Spot Cross-Section Semi-Minor Axis	32-bit floating point
32	Beam Spot Cross-Section Orientation Angle	32-bit floating point
32	Peak Irradiance	32-bit floating point
32	Padding	32 bits unused
Total DE Precision Aimpoint record size = 768 bits		

6.2.20.4 DE Target Energy Deposition record

Directed energy deposition properties for a target entity shall be communicated using the DE Target Energy Deposition record. This record is required to be included inside another DE record as it does not have a record type.

The DE Target Energy Deposition record shall contain the following fields:

a) *Target Entity ID*. This field shall identify the target entity. This field shall be represented by an Entity Identifier record (see 6.2.28).

b) *Padding*. This field shall contain 16 bits of padding.

c) *Peak Irradiance*. This field shall identify the current peak irradiance deposited at this target in units of Watts per square meter and shall be represented by a 32-bit floating point number.

The format of the DE Target Energy Deposition record shall be as shown in Table 46.

Table 46—DE Target Energy Deposition record

Field size (bits)	Field name	Data type
48	Target Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer

Table 46—DE Target Energy Deposition record *(continued)*

Field size (bits)	Field name	Data type
16	Padding	16 bits unused
32	Peak Irradiance	32-bit floating point
Total DE Target Energy Deposition record size = 96 bits		

6.2.21 EE Attribute records

6.2.21.1 Blanking Sector attribute record

6.2.21.1.1 General

The Blanking Sector attribute record may be used to convey persistent areas within a scan volume where emitter power for a specific active emitter beam is reduced to an insignificant value, e.g., –60 dBm. In the example illustrated in Figure 45, a ground-based circular-scanning radar utilizes a single blanking sector to limit the radiation hazard in a nearby bivouac area. An additional Blanking Sector attribute record may be added to account for another no-radiate area. The Blanking Sector attribute record shall not be used to communicate no-radiate areas that can be referenced to an emitter mode. For example, the Blanking Sector attribute record is not appropriate for use in communicating the change from a 0° to 90° elevation scan volume to a 70° to 90° elevation scan volume when a radar switches from normal search to anti-ballistic missile search mode. The Blanking Sector attribute record shall be included in an Attribute PDU as specified herein. See also 5.7.3.

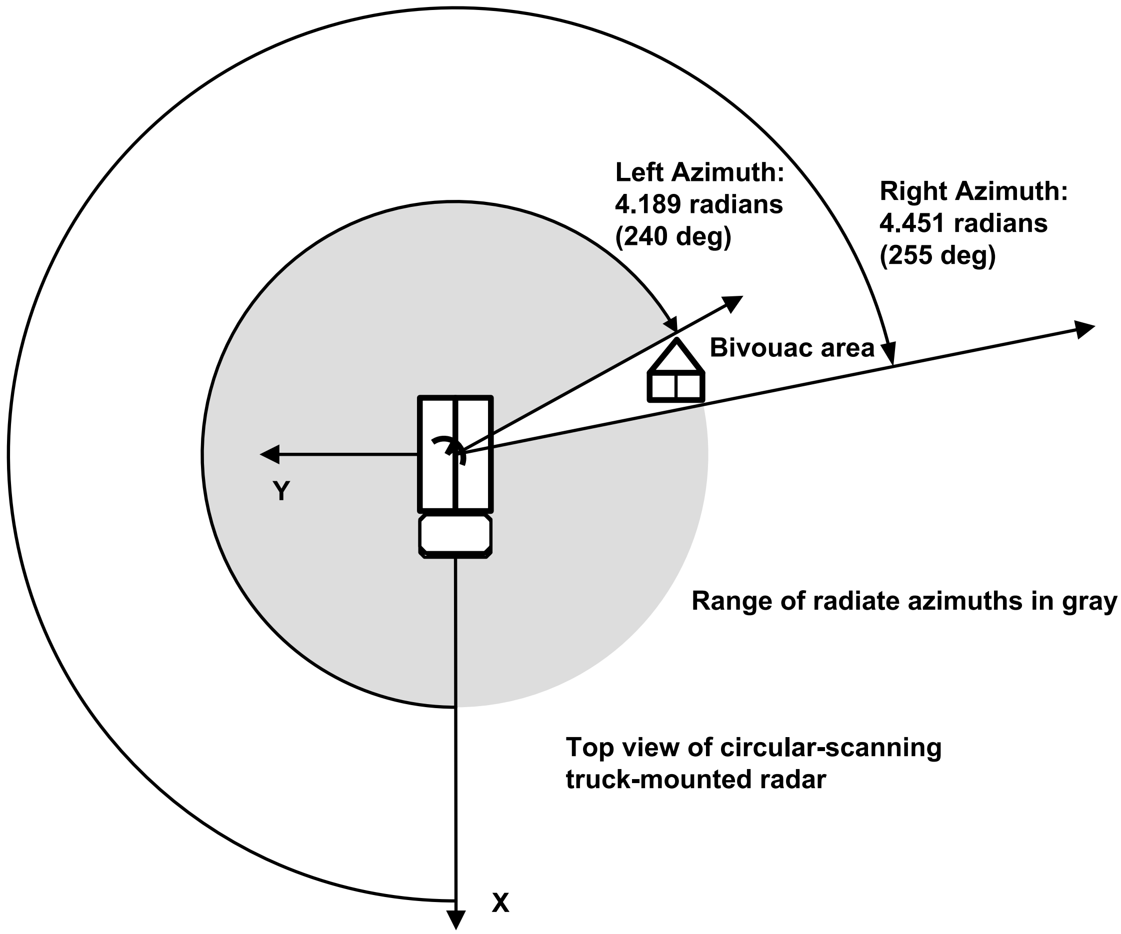


Figure 45—Blanking sector example

6.2.21.1.2 Record definition

The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 3500.

b) *Record Length*. This field shall specify the length of the Blanking Sector attribute record in octets. This field shall be specified by a 16-bit unsigned integer.

c) *Emitter Number*. This field indicates the emitter system for which the blanking sector values are applicable. This field shall be represented by an 8-bit unsigned integer (see 7.6.2).

d) *Beam Number*. This field indicates the beam for which the blanking sector values are applicable. This field shall be represented by an 8-bit unsigned integer (see 7.6.2).

e) *State Indicator*. This field shall be used to indicate whether blanking sector data have changed since issuance of the last Blanking Sector attribute record for this beam, if the Blanking Sector attribute record is part of a heartbeat update to meet periodic update requirements or if blanking sector data for the beam has ceased. This field shall be represented by an 8-bit enumeration (see [UID 300]). (See also 7.6.2.)

f) *Left Azimuth*. This field is provided to indicate the left-most azimuth (clockwise in radians) for which emitted power is reduced. This angle is measured in the *x–y* plane of the radar's entity coordinate system (see 1.6.3). The range of permissible values is 0 to 2π, with zero pointing in the *x*-direction. This field shall be represented by a 32-bit floating point number.

g) *Right Azimuth*. This field is provided to indicate the right-most azimuth (clockwise in radians) for which emitted power is reduced. This angle is measured in the *x–y* plane of the radar's entity coordinate system (see 1.6.3). The range of permissible values is 0 to 2π, with zero pointing in the *x*-direction. This field shall be represented by a 32-bit floating point number.

h) *Lower Elevation*. This field is provided to indicate the lowest elevation (in radians) for which emitted power is reduced. This angle is measured positive upward with respect to the *x–y* plane of the radar's entity coordinate system (see 1.6.3). The range of permissible values is $-\pi/2$ to $\pi/2$. This field shall be represented by a 32-bit floating point number.

i) *Upper Elevation*. This field is provided to indicate the highest elevation (in radians) for which emitted power is reduced. This angle is measured positive upward with respect to the *x–y* plane of the radar's entity coordinate system (see 1.6.3). The range of permissible values is $-\pi/2$ to $\pi/2$. This field shall be represented by a 32-bit floating point number.

j) *Residual Power*. This field shall specify the residual effective radiated power in the blanking sector in dBm. This field shall be represented by a 32-bit floating point number (see 6.2.22).

The format of the Blanking Sector attribute record shall be as shown in Table 47.

Table 47—Blanking Sector attribute record

Field size (bits)	Field name	Data type
32	Record Type = 3500	32-bit enumeration
16	Record Length = 40	16-bit unsigned integer
16	Padding	16 bits unused
8	Emitter Number	8-bit unsigned integer
8	Beam Number	8-bit unsigned integer
8	State Indicator	8-bit enumeration
8	Padding	8 bits unused
32	Left Azimuth	32-bit floating point
32	Right Azimuth	32-bit floating point
32	Lower Elevation	32-bit floating point
32	Upper Elevation	32-bit floating point
32	Residual Power	32-bit floating point
64	Padding	64 bits unused
Total Blanking Sector attribute record size = 320 bits		

6.2.21.1.3 Issuance rules

The issuance rules for this record are as follows:

a) The State Indicator value for the initial Attribute PDU with Blanking Sector attribute records shall be set to Changed Data (1).

b) Multiple Blanking Sector attribute records may be used to indicate more than one blanking sector. Multiple overlapping Blanking Sector attribute records may be used to build a single blanking sector.

c) An Attribute PDU shall be issued when a blanking sector is added or removed for the emitter. It shall include Blanking Sector attribute record(s) for each blanking sector added or removed. For an added or removed blanking sector, the State Indicator value shall be set to Changed Data (1) or Ceased (2), respectively.

d) An Attribute PDU shall be issued when changes in blanking sector azimuth or elevation values exceed specified thresholds. The azimuth and elevation thresholds shall be identified by the symbolic names EE_AZ_THRSH and EE_EL_THRSH, respectively. (See Table 25 for parameter details and default values.) The Attribute PDU shall include Blanking Sector attribute records for each sector with changed values. The State Indicator value shall be set to Changed Data (1).

e) Attribute PDUs containing Blanking Sector attribute records for active blanking sectors shall be issued at the heartbeat interval specified for the associated EE PDU (see 5.3.6.3). The State Indicator value shall be set to Heartbeat (0).

f) Simulations shall not issue Blanking Sector attribute records with Beam Number or Emitter Number values indicating NO_BEAM or NO_EMITTER, respectively (see 6.1.8).

g) If all beams on the emitter system are applying the blanking sector, the Beam Number may indicate ALL_BEAMS (see 6.1.8).

h) If all emitters on the entity are applying the blanking sector, the Emitter Number may indicate ALL_EMITTERS (see 6.1.8).

i) To remove a blanking sector, an Attribute PDU shall be issued with State Indicator set to Ceased (2). Until the beam is again supplemented with blanking sector data, no subsequent Blanking Sector attribute records shall be issued for the beam.

6.2.21.1.4 Receipt rules

The receipt rules for this record are as follows:

a) Upon receipt of an Attribute PDU containing one or more Blanking Sector attribute records, the receiving simulation application may consider the receiver's location with respect to the union of blanking sectors to determine whether the emissions affect the receiver.

b) If the countdown timer for a beam blanking sector expires, then receiving simulations shall consider that blanking sector to be inactive. The timeout parameter shall be established as the value of HBT_PDU_EE multiplied by HBT_TIMEOUT_MPLIER (see 6.1.8 for details of these two parameters and their default values).

c) Upon receipt of EE PDUs deactivating beams (see 5.7.3.6), receiving simulations shall consider associated blanking sectors to be inactive.

6.2.21.2 Angle Deception attribute record

6.2.21.2.1 General

The Angle Deception attribute record may be used to communicate discrete values that are associated with angle deception jamming that cannot be referenced to an emitter mode. The values provided in the record shall be considered valid only for the victim radar beams listed in the jamming beam's Track/Jam Data records (provided in the associated EE PDU). (The victim radar beams are those that are targeted by the jammer.)

6.2.21.2.2 Record definition

The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 3501.

b) *Record Length*. This field shall specify the length of the Angle Deception attribute record in octets. This field shall be specified by a 16-bit unsigned integer.

c) *Emitter Number*. This field indicates the emitter system for which the angle deception values are applicable. This field shall be represented by an 8-bit unsigned integer (see 7.6.2).

d) *Beam Number*. This field indicates the jamming beam for which the angle deception values are applicable. This field shall be represented by an 8-bit unsigned integer (see 7.6.2).

e) *State Indicator*. This field shall be used to indicate whether angle deception data have changed since issuance of the last Angle Deception attribute record for this beam, whether the Angle Deception attribute record is part of a heartbeat update to meet periodic update requirements, or whether the angle deception data for the beam has ceased. This field shall be represented by an 8-bit enumeration (see [UID 300]). (See also 7.6.2.)

f) *Azimuth Offset*. This field indicates the relative azimuth angle at which the deceptive radar returns are centered. This angle is measured in the *x–y* plane of the victim radar's entity coordinate system (see 1.6.3). This angle is measured in radians from the victim radar entity's azimuth for the true jammer position to the center of the range of azimuths in which deceptive radar returns are perceived as shown in Figure 46. Positive and negative values indicate that the perceived positions of the jammer are right and left of the true position of the jammer, respectively. The range of permissible values is $-\pi$ to π. This field shall be represented by a 32-bit floating point number.

g) *Azimuth Width*. This field indicates the range of azimuths (in radians) through which the deceptive radar returns are perceived, centered on the azimuth offset as shown in Figure 46. The range of permissible values is 0 to 2π radians. This field shall be represented by a 32-bit floating point number.

h) *Azimuth Pull Rate*. This field indicates the rate (in radians per second) at which the Azimuth Offset value is changing. Positive and negative values indicate that the Azimuth Offset is moving to the right or left, respectively. This field shall be represented by a 32-bit floating point number.

i) *Azimuth Pull Acceleration*. This field indicates the rate (in radians per second squared) at which the Azimuth Pull Rate value is changing. Azimuth Pull Acceleration is defined as positive to the right and negative to the left. This field shall be represented by a 32-bit floating point number.

j) *Elevation Offset*. This field indicates the relative elevation angle at which the deceptive radar returns begin. This angle is measured as an angle with respect to the *x–y* plane of the victim radar's entity coordinate system (see 1.6.3). This angle is measured in radians from the victim radar entity's elevation for the true jammer position to the center of the range of elevations in which deceptive radar returns are perceived as shown in Figure 47. Positive and negative values indicate that the perceived positions of the jammer are above and below the true position of the jammer, respectively. The range of permissible values is $-\pi/2$ to $\pi/2$. This field shall be represented by a 32-bit floating point number.

k) *Elevation Width*. This field indicates the range of elevations (in radians) through which the deceptive radar returns are perceived, centered on the elevation offset as shown in Figure 47. The range of permissible values is 0 to π radians. This field shall be represented by a 32-bit floating point number.

l) *Elevation Pull Rate*. This field indicates the rate (in radians per second) at which the Elevation Offset value is changing. Positive and negative values indicate that the Elevation Offset is moving up or down, respectively. This field shall be represented by a 32-bit floating point number.

m) *Elevation Pull Acceleration*. This field indicates the rate (in radians per second squared) at which the Elevation Pull Rate value is changing. Elevation Pull Acceleration is defined as positive to upward and negative downward. This field shall be represented by a 32-bit floating point number.

The format of the Angle Deception attribute record shall be as shown in Table 48.

Table 48—Angle Deception attribute record

Field size (bits)	Field name	Data type
32	Record Type = 3501	32-bit enumeration
16	Record Length = 48	16-bit unsigned integer
16	Padding	16 bits unused
8	Emitter Number	8-bit unsigned integer
8	Beam Number	8-bit unsigned integer
8	State Indicator	8-bit enumeration
8	Padding	8 bits unused
32	Azimuth Offset	32-bit floating point
32	Azimuth Width	32-bit floating point
32	Azimuth Pull Rate	32-bit floating point
32	Azimuth Pull Acceleration	32-bit floating point
32	Elevation Offset	32-bit floating point
32	Elevation Width	32-bit floating point
32	Elevation Pull Rate	32-bit floating point
32	Elevation Pull Acceleration	32-bit floating point
32	Padding	32 bits unused
Total Angle Deception attribute record size = 384 bits		

Figures 46 and 47 illustrate the concept of angle deception jamming.

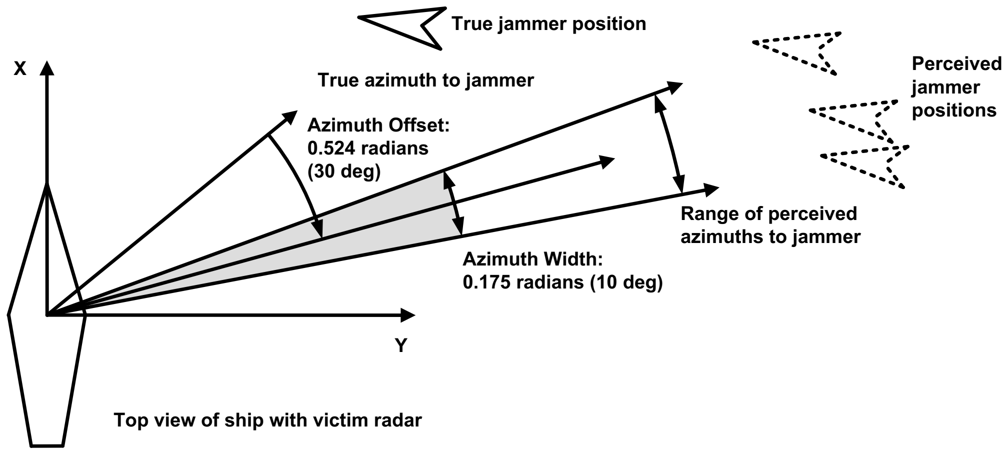


Figure 46—Angle deception azimuth

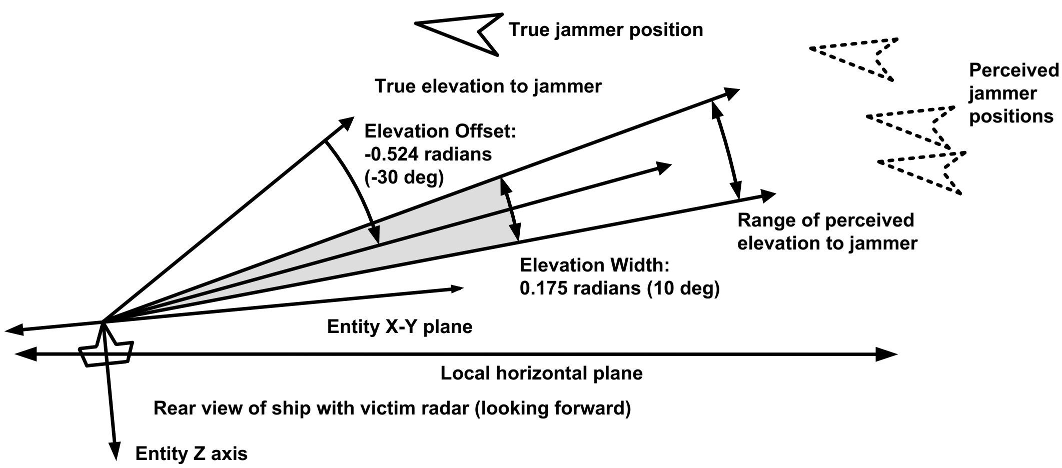


Figure 47—Angle deception elevation

6.2.21.2.3 Issuance rules

The issuance rules for this record are as follows:

a) The initial Attribute PDU with an Angle Deception attribute record may be issued for any beam with angle deception jamming indicated in the Jamming Technique record (see 6.2.49). The State Indicator value shall be set to Changed Data (1).

b) An Attribute PDU shall be issued when the offset or width values exceed specified thresholds for angle deception. The azimuth and elevation thresholds shall be identified by the symbolic names EE_AZ_THRSH and EE_EL_THRSH, respectively. (See Table 25 for parameter details and default values.) The State Indicator value shall be set to Changed Data (1).

c) An Attribute PDU shall be issued when the pull rate or pull acceleration values exceed specified thresholds. The rate and acceleration thresholds shall be identified by the symbolic names EE_AD_PULRAT_THRSH and EE_AD_PULACC_THRSH, respectively. (See Table 25 for parameter details and default values.) The State Indicator value shall be set to Changed Data (1).

d) After issuance of the initial Attribute PDU with an Angle Deception attribute record, subsequent Attribute PDUs shall be issued at the heartbeat interval specified for the EE PDU (see 5.3.6.3 and 5.7.3.8). The State Indicator value shall be set to Heartbeat (0).

e) Simulations shall not issue Angle Deception attribute records with Beam Number or Emitter Number values indicating NO_BEAM or NO_EMITTER, respectively (see 6.1.8).

f) If all beams on the jamming emitter system are applying the angle deception technique, the Beam Number may indicate ALL_BEAMS (see 6.1.8).

g) If all emitters on the jamming entity are applying the angle deception technique, the Emitter Number may indicate ALL_EMITTERS (see 6.1.8).

h) To stop supplementing a jamming beam with angle deception data, an Attribute PDU with an Angle Deception attribute record shall be issued with State Indicator set to Ceased (2). Until the beam is again supplemented with angle deception data, no subsequent Angle Deception attribute records shall be issued for the beam.

6.2.21.2.4 Receipt rules

The receipt rules for this record are as follows:

a) Upon receipt of an Attribute PDU containing one or more Angle Deception attribute records corresponding to an angle deception jamming beam, the receiving simulation application may use the Angle Deception attribute record to determine whether the emissions affect the receiver.

b) If the countdown timer for a beam angle deception expires, then receiving simulations shall disregard previously received angle deception data for that beam. The timeout parameter shall be established as the value of HBT_PDU_EE multiplied by HBT_TIMEOUT_MPLIER (see 6.1.8 for details of these two parameters and their default values).

c) If the Angle Deception attribute record does not correspond to an angle deception jamming beam, receiving simulations shall disregard the angle deception data.

d) Upon receipt of EE PDUs deactivating angle deception jamming beams (see 5.7.3.6), receiving simulations shall disregard angle deception data for those beams.

6.2.21.3 False Targets attribute record

6.2.21.3.1 General

The False Targets attribute record shall be used to communicate discrete values that are associated with false targets jamming that cannot be referenced to an emitter mode. The values provided in the False Targets attribute record shall be considered valid only for the victim radar beams listed in the jamming beam's Track/Jam Data records (provided in the associated EE PDU).

6.2.21.3.2 Record definition

The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 3502.

b) *Record Length*. This field shall specify the length of the False Targets attribute record in octets. This field shall be specified by a 16-bit unsigned integer.

c) *Emitter Number*. This field indicates the emitter system generating the false targets. This field shall be represented by an 8-bit unsigned integer (see 7.6.2).

d) *Beam Number*. This field indicates the jamming beam generating the false targets. This field shall be represented by an 8-bit unsigned integer (see 7.6.2).

e) *State Indicator*. This field shall be used to indicate whether false target data have changed since issuance of the last False Targets attribute record for this beam, whether the False Targets attribute record is part of a heartbeat update to meet periodic update requirements, or whether false target data for the beam has ceased. This field shall be represented by an 8-bit enumeration (see [UID 300]). (See also 7.6.2.)

f) *False Target Count*. This field shall specify the number of false targets presented to the victim radar. This field shall be represented by a 16-bit unsigned integer.

g) *Walk Speed*. This field shall specify the speed (in meters per second) at which false targets move toward the victim radar. Negative values shall indicate a velocity away from the victim radar. This field shall be represented by a 32-bit floating point number.

h) *Walk Acceleration*. This field shall specify the rate (in meters per second squared) at which false targets accelerate toward the victim radar. Negative values shall indicate an acceleration direction away from the victim radar. This field shall be represented by a 32-bit floating point number.

i) *Maximum Walk Distance*. This field shall specify the distance (in meters) that a false target is to walk before it pauses in range. This field shall be represented by a 32-bit floating point number.

j) *Keep Time*. This field shall specify the simulation time (in seconds) that a false target is to be held at the Maximum Walk Distance before it resets to its initial position. This field shall be represented by a 32-bit floating point number.

k) *Echo Spacing*. This field shall specify the distance between false targets in meters. A positive value for this field shall indicate that the second and subsequent false targets appear at the given spacing relative to the first false target away from the victim radar (i.e., at increasing ranges from the victim). A negative value shall indicate that the second and subsequent false targets appear toward the victim radar relative to the first false target. This field shall be represented by a 32-bit floating point number.

l) *First Target Offset*. Sets the position of the first false target relative to the jamming entity in meters. A positive value for this field shall indicate that the first target appears at the given distance relative to the jamming entity away from the victim radar (i.e., at an increasing range from the victim). A negative value shall indicate the first false target appears toward the victim radar relative to the jamming entity. A value of zero shall place the first false target at the same position as the jamming entity.

The format of the False Targets attribute record shall be as shown in Table 49.

Table 49—False Targets attribute record

Field size (bits)	Field name	Data type
32	Record Type = 3502	32-bit enumeration
16	Record Length = 40	16-bit unsigned integer
16	Padding	16 bits unused
8	Emitter Number	8-bit unsigned integer
8	Beam Number	8-bit unsigned integer
8	State Indicator	8-bit enumeration
8	Padding	8 bits unused

Table 49—False Targets attribute record *(continued)*

Field size (bits)	Field name	Data type
16	Padding	16 bits unused
16	False Target Count	16-bit unisgned integer
32	Walk Speed	32-bit floating point
32	Walk Acceleration	32-bit floating point
32	Maximum Walk Distance	32-bit floating point
32	Keep Time	32-bit floating point
32	Echo Spacing	32-bit floating point
32	First Target Offset	32-bit floating point
Total False Targets attribute record size = 320 bits		

Figure 48 illustrates the effect modeled through use of the False Targets attribute record. The jammer is initially positioned 40 000 m from the target radar (the jammer position is not shown in Figure 48). The False Target Count value is 3, Walk Speed is 2000 m/s, Walk Acceleration is 450 m/s^2, Maximum Walk Distance is 20000 m, Keep Time is 10 s, Echo Spacing is –15 000 m, and First Target Offset is 10 000 m. Note that the positive First Target Offset value places the first false target away from the victim radar (40 000 + 10 000 = 50 000 m) at Time = 0, while the negative Echo Spacing places the second and subsequent false targets incrementally toward the victim relative to the first false target, at 35 000 m and 20 000 m, respectively, at Time = 0.

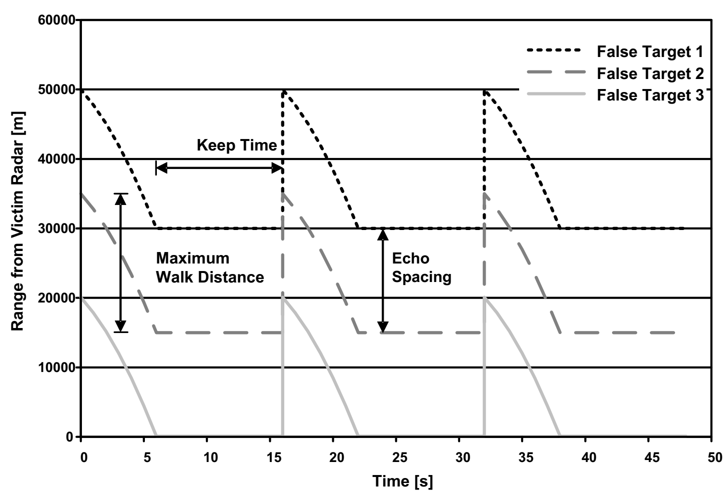


Figure 48—False targets jamming

6.2.21.3.3 Issuance rules

The issuance rules for this record are as follows:

a) The initial Attribute PDU with one or more False Targets attribute records may be issued for any beam with false targets jamming indicated in the Jamming Technique record (see 6.2.49). The State Indicator value shall be set to Changed Data (1).

b) Once an initial Attribute PDU with a False Targets attribute record is issued, an Attribute PDU containing one or more False Targets attribute records shall be issued with the State Indicator value set to Changed Data (1):

    1) When the False Target Count value changes.

    2) When changes in Walk Velocity, Walk Acceleration, Maximum Walk Distance, Keep Time, or Echo Spacing values exceed specified thresholds. The thresholds shall be identified by the symbolic names EE_FT_VEL_THRSH, EE_FT_ACC_THRSH, EE_FT_MWD_THRSH, EE_FT_KT_THRSH, and EE_FT_ESP_THRSH, respectively. (See Table 25 for parameter details and default values.)

c) After issuance of the initial Attribute PDU with a False Targets attribute record, subsequent Attribute PDUs shall be issued at the heartbeat interval specified for the EE PDU (see 5.3.6.3 and 5.7.3.8). The State Indicator value shall be set to Heartbeat (0).

d) Simulations shall not issue False Targets attribute records with Beam Number or Emitter Number values indicating NO_BEAM or NO_EMITTER, respectively (see 6.1.8).

e) If all beams on the jamming emitter system are applying the false targets technique, the Beam Number may indicate ALL_BEAMS (see 6.1.8).

f) If all emitters on the jamming entity are applying the false targets technique, the Emitter Number may indicate ALL_EMITTERS (see 6.1.8).

g) To stop supplementing a jamming beam with false targets data, an Attribute PDU with a False Targets attribute record shall be issued with State Indicator set to Ceased (2). Until the beam is again supplemented with false targets data, no subsequent False Targets attribute records shall be issued for the beam.

6.2.21.3.4 Receipt rules

The receipt rules for this record are as follows:

a) Upon receipt of an Attribute PDU containing False Targets attribute records for an emitter previously received in an EE PDU with a jamming beam function and false targets jamming technique, the receiving simulation application may consider the False Targets attribute records to determine whether the emissions are detectable and whether the emissions affect the capabilities of the radar.

b) If the countdown timer for a beam false target expires, then receiving simulations may consider the false target technique to be ineffective for that beam. The timeout parameter shall be established as the value of HBT_PDU_EE multiplied by HBT_TIMEOUT_MPLIER (see 6.1.8 for details of these two parameters and their default values).

c) If the False Targets attribute record does not correspond to a false targets jamming beam, receiving simulations shall disregard the false targets data.

d) Upon receipt of EE PDUs deactivating false targets jamming beams (see 5.7.3.6), receiving simulations shall disregard false targets data for those beams.

6.2.22 EE Fundamental Parameter Data record

The EE Fundamental Parameter Data record contains electromagnetic emission regeneration parameters that are variable throughout a scenario dependent on the actions of the participants in the simulation. This record also provides basic parametric data that may be used to support low-fidelity simulations that do not have the processing capability to model a high-fidelity regeneration of emission beams. This record shall consist of five fields. The fields of this record are as follows:

a) *Frequency*. For a non-frequency-agile emission, this field shall specify the average center frequency in Hertz. For a frequency-agile emission, this field shall specify the middle of the range of center frequencies:

$$F = \frac{(f_u + f_l)}{2}$$

where

F is the Frequency field value
f_u is the upper limit of center frequencies
f_l is the lower limit of center frequencies

Frequency modulation for a particular emitter beam and mode shall be derived from database parameters stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

b) *Frequency Range*. This field shall specify the difference between the upper and lower limit of center frequencies corresponding to the Frequency field. Thus, for non-frequency-agile emissions, the frequency range shall be zero. For frequency-agile emissions, it shall indicate the range of agile radio frequencies:

$$R = f_u - f_l$$

where

R is the Frequency Range field value
f_u is the upper limit of center frequencies
f_l is the lower limit of center frequencies

The actual list of center frequencies for a particular emitter beam and mode shall be derived from database parameters stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

c) *Effective Radiated Power*. This field shall specify the average effective radiated power for the emission in dBm. For a radar or jammer, this field shall indicate the peak of the transmitted power. Thus, it includes peak transmitter power, transmission line losses, and peak of the antenna gain. Effective Radiated Power modulation for a particular emitter beam and mode shall be derived from database parameters stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

d) *Pulse Repetition Frequency*. This field shall specify the average pulse repetition frequency of the emission in Hertz. Pulse Repetition Frequency modulation for a particular emitter beam and mode shall be derived from database parameters stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

e) *Pulse Width*. This field shall specify the average pulse width of the emission in microseconds. Pulse modulation for a particular emitter beam and mode shall be derived from database parameters stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

The format of the EE Fundamental Parameter Data record shall be as shown in Table 50.

Table 50—EE Fundamental Parameter Data record

Field size (bits)	Field name	Data type
32	Frequency	32-bit floating point
32	Frequency Range	32-bit floating point
32	Effective Radiated Power	32-bit floating point
32	Pulse Repetition Frequency	32-bit floating point
32	Pulse Width	32-bit floating point
Total EE Fundamental Parameter Data record size = 160 bits		

6.2.23 Emitter System record

Information about a particular emitter shall be represented using an emitter system record. Use of the Emitter System record in the EE PDU is described in 5.7.3.3. This record shall consist of three fields: Emitter Name, Emitter Function, and Emitter Number. The fields of this record are as follows:

a) *Emitter Name.* This field shall specify the nomenclature for a particular emitter and shall be represented by a 16-bit enumeration (see [UID 75]).
b) *Emitter Function.* This field shall specify the general function associated with a particular emitter. Typical functions include airborne fire control, ground surveillance radar, and so on. This field is intended to help receiving entities determine whether the EE PDU is of interest to the systems simulated by that entity. This field shall be represented by an 8-bit enumeration (see [UID 76]).
c) *Emitter Number.* This field shall specify the specific emitter system associated with an entity. This field allows the differentiation of multiple emitter systems on an entity even if in some instances two or more of the emitters are of identical emitter types. This field shall be represented by an 8-bit unsigned integer.

The format of the Emitter System record shall be as shown in Table 51.

Table 51—Emitter System record

Field size (bits)	Field name	Data type
16	Emitter Name	16-bit enumeration
8	Emitter Function	8-bit enumeration
8	Emitter Number	8-bit unsigned integer
Total Emitter System record size = 32 bits		

6.2.24 Engine Fuel record

An entitys engine fuel information shall be represented by one or more Engine Fuel records. For each type or location of engine fuel, this record shall specify the type, location, and quantity of engine fuel that an entity contains.

This record may be used (optionally) in a transfer transaction to send internal state data from the divesting simulation to the acquiring simulation (see 5.9.4). This record may also be used for other purposes.

The Engine Fuel record can be used to represent either non-location-specific or location-specific engine fuel information, as follows:

— *Nonspecific Fuel Location*. One Engine Fuel record is used for all engine fuel associated with the entity if the *Fuel Location* field is zero.

— *Specific Fuel Location*. One Engine Fuel record may be used to indicate the quantity of fuel remaining for each fuel location specified in the *Fuel Location* field associated with the entity.

The Engine Fuel records associated with an entity shall be either Nonspecific Fuel Location (one record) or Specific Fuel Location (one or more records). There shall be no mixing of specific and nonspecific fuel locations in a given transfer.

The Engine Fuel record shall contain the following fields:

a) *Fuel Quantity*. This field shall indicate the quantity of fuel remaining, in increments specified by the Fuel Measurement Units field. When the Fuel Location field is set to zero, the fuel quantity applies to all engines associated with the entity. This field shall be represented by a 32-bit unsigned integer.

b) *Fuel Measurement Units*. This field shall identify the fuel measurement units. It shall be represented by an 8-bit enumeration (see [UID 328]).

c) *Fuel Type*. This field shall identify the type of fuel. It shall be represented by an 8-bit enumeration (see [UID 413]).

d) *Fuel Location.* This field shall indicate the location of the fuel as related to the entity. A zero indicates no specific location. It shall be represented by an 8-bit enumeration (see [UID 329]).

The format of the Engine Fuel record shall be as shown in Table 52.

Table 52—Engine Fuel record

Field size (bits)	Field name	Data type
32	Fuel Quantity	32-bit unsigned integer
8	Fuel Measurement Units	8-bit enumeration
8	Fuel Type	8-bit enumeration
8	Fuel Location	8-bit enumeration
8	Padding	8 bits unused
Total Engine Fuel record size = 64 bits		

6.2.25 Engine Fuel Reload record

An entity's engine fuel reload information shall be represented by one or more Engine Fuel Reload records. For each type or location of engine fuel, this record shall specify the type, location, fuel measurement units, reload quantity, and maximum quantity for engine fuel either as applicable to the whole entity or as a specific fuel location (tank).

This record shall be required to be sent if one or more Engine Fuel records are included in a transfer transaction. One Engine Fuel Reload record shall be sent for each Engine Fuel record that is sent and shall mirror the type of station representation (nonspecific or specific) found in that record. This record may also be used for other purposes. This record is optional for receipt. If processed, the data may be used or ignored as deemed appropriate by the simulation (e.g., an instant reload feature would override reload times, and an infinite quantity feature would override quantity values).

The Engine Fuel Reload record can be used to represent either non-location-specific or location-specific engine fuel information, as follows:

— *Nonspecific Fuel Location*. One Engine Fuel Reload record is used for all engine fuel associated with the entity if the Fuel Location field is zero.

— *Specific Fuel Location*. One Engine Fuel Reload record shall indicate the standard and maximum quantities and reload times for a quantity of fuel remaining for each fuel location (tank) specified in the Fuel Location field associated with the entity.

The Engine Fuel Reload records associated with an entity shall be either Nonspecific Fuel Location (one record) or Specific Fuel Location (one or more records). There shall be no mixing of specific and nonspecific fuel locations in a given transfer.

The reload times may vary for different fuel locations that can accept a specific engine fuel type so only an overall reload time is given for loading the standard or maximum quantity at either a fuel location or for the entire entity, whichever is specified. How quantities and reload times are calculated is beyond the scope of this standard.

The Engine Fuel Reload record shall contain the following fields:

a) *Standard Quantity*. This field shall identify the standard quantity of this fuel type normally loaded at this fuel location if a fuel location is specified. If the Fuel Location field is set to zero, then this is the total quantity of this fuel type that would be present in a standard reload of all applicable fuel locations associated with this entity. This field shall be represented by a 32-bit unsigned integer.

b) *Maximum Quantity*. This field shall specify the maximum quantity of this fuel type that this fuel location is capable of holding when a fuel location is specified. This would be the value used when a maximum reload was desired to be set for this fuel location. If the Fuel Location field is set to zero, then this is the maximum quantity of this fuel type that would be present on this entity at all fuel locations that can accept this fuel type. This field shall be represented by a 32-bit unsigned integer.

c) *Standard Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the standard quantity of this fuel type at this specific fuel location. When the Fuel Location field is set to zero, this shall be the number of seconds of simulation time required to perform a standard quantity reload of this fuel type at all applicable fuel locations for this entity. This field shall be represented by a 32-bit unsigned integer.

d) *Maximum Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the maximum possible quantity of this fuel type at this fuel location. When the Fuel Location field is set to zero, this shall be the number of seconds of simulation time required to perform a maximum quantity load/reload of this fuel type at all applicable fuel locations for this entity. This field shall be represented by a 32-bit unsigned integer.

e) *Fuel Measurement Units*. This field shall identify the fuel measurement units. It shall be represented by an 8-bit enumeration (see [UID 328]).

f) *Fuel Type*. This field shall identify the type of fuel. It shall be represented by an 8-bit enumeration (see [UID 413]).

g) *Fuel Location*. This field shall indicate the location of the fuel as related to the entity. A zero indicates no specific location. It shall be represented by an 8-bit enumeration (see [UID 329]).

The format of the Engine Fuel Reload record shall be as shown in Table 53.

Table 53—Engine Fuel Reload record

Field size (bits)	Field name	Data type
32	Standard Quantity	32-bit unsigned integer
32	Maximum Quantity	32-bit unsigned integer
32	Standard Quantity Reload Time	32-bit unsigned integer
32	Maximum Quantity Reload Time	32-bit unsigned integer
8	Fuel Measurement Units	8-bit enumeration
8	Fuel Type	8-bit enumeration
8	Fuel Location	8-bit enumeration
8	Padding	8 bits unused
Total Engine Fuel Reload record size = 160 bits		

6.2.26 Entity Appearance record

The appearance of an entity shall be specified by an Entity Appearance record. This record shall be defined as a 32-bit record (see [UID 31-43]).

6.2.27 Entity Capabilities record

The capabilities of an entity shall be specified by an Entity Capabilities record. This record shall be defined as a 32-bit record (see [UID 55]).

6.2.28 Entity Identifier record

The unique designation of each entity in an event or exercise that is not contained in a Live Entity PDU shall be specified by an Entity Identifier record. The Entity Identifier (Entity ID) record shall consist of a Simulation Address (see 6.2.80) and an Entity Number.

The Entity Identifier is an object identifier. General requirements for all object identifiers are specified in 4.2.5.2.

The format of the Entity Identifier record shall be as shown in Table 54.

Table 54—Entity Identifier record

Field size (bits)	Field name	Data type
32	Simulation Address	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
16	Entity Number	16-bit unsigned integer
Total Entity Identifier record size = 48 bits		

6.2.29 Entity Marking record

Entity markings shall be specified by the Entity Marking record. This record shall specify the character set used in the marking and the string of characters to be interpreted for display. The character set shall be specified by an 8-bit enumeration (see [UID 45]). The string of characters shall be represented by an 11-element character string. This string shall begin with the most significant octet located at the lowest address. The designation of unused characters shall be specified by the character set. The format of the Entity Marking record shall be as shown in Table 55.

Table 55—Entity Marking record

Field size (bits)	Field name	Data type
8	Character set	8-bit enumeration
8	1st Character	8-bit unsigned integer
8	2nd Character	8-bit unsigned integer
	• • •	
8	11th Character	8-bit unsigned integer
Total Entity Marking record size = 96 bits		

6.2.30 Entity Type record

The type of entity in a DIS exercise shall be specified by an Entity Type record. This record shall specify the kind of entity, the country of design, the domain, the specific identification of the entity, and any extra information necessary for describing the entity. Fields not used shall contain the value zero. A table of comprehensive entity type enumerations can be found in SISO-REF-010 (see [UID 30]). The fields of this record are as follows:

a) *Entity Kind.* This field shall identify the kind of entity described by the Entity Type record. This field shall be represented by an 8-bit enumeration (see [UID 7]).

b) *Domain.* This field shall specify the domain in which the entity operates (e.g., subsurface, surface, and land) except for munition entities. For munition entities this field shall specify the domain of the target (for example, the munition might be a surface-to-air, so the domain would be anti-air). This field shall be represented by an 8-bit enumeration (see [UID 8 and UID 14]).

c) *Country.* This field shall specify the country to which the design of the entity or its design specification is attributed. This field shall be represented by a 16-bit enumeration (see [UID 29]).

d) *Category.* This field shall specify the main category that describes the entity. This field shall be represented by an 8-bit enumeration.

e) *Subcategory.* This field shall specify a particular subcategory to which an entity belongs based on the Category field. This field shall be represented by an 8-bit enumeration.

f) *Specific.* This field shall specify specific information about an entity based on the Subcategory field. This field shall be represented by an 8-bit enumeration.

g) *Extra.*This field shall specify extra information required to describe a particular entity. The contents of this field shall depend on the type of entity represented. This field shall be represented by an 8-bit enumeration.

The format of the Entity Type record shall be as shown in Table 56.

Table 56—Entity Type record

Field size (bits)	Field name	Data type
8	Entity Kind	8-bit enumeration
8	Domain	8-bit enumeration
16	Country	16-bit enumeration
8	Category	8-bit enumeration
8	Subcategory	8-bit enumeration
8	Specific	8-bit enumeration
8	Extra	8-bit enumeration
Total Entity Type record size = 64 bits		

6.2.31 Environment record

6.2.31.1 General

Specific information about a geometry, a state associated with a geometry, a bounding sphere containing all listed geometries, or an associated entity ID shall be specified by an Environment record. This record shall specify the type, length, index, and specific geometry or state record. The Environment record shall contain the following fields:

a) *Type.* This field shall specify the record type and shall be a 32-bit enumeration (see [UID 250]).

b) *Length.* This field shall specify the length in bits of the Environment record and shall be represented by a 16-bit unsigned integer.

c) *Index.* This field shall identify the sequentially numbered record index and shall be represented by an 8-bit unsigned integer.

d) *Geometry or State record.* This field shall contain one geometry record or one state record (see 6.2.31.2 and 6.2.31.3).

e) *Padding.* This space is reserved to bring the record length to a 64-bit boundary.

The format of the Environment record shall be as shown in Table 57.

Table 57—Environment record

Field size (bits)	Field name	Data type
32	Type	32-bit enumeration
16	Length ($64 + K + P$)	16-bit unsigned integer
8	Index	8-bit unsigned integer
8	Padding	8 bits unused
K	Geometry or State record	K bits
P	Padding	Padding to 64-bit boundary—P bits
Total Environment record size = $64 + K + P$ bits where K is the length of the geometry or state record in bits P is the number of padding bits, which is $\lceil K/64 \rceil 64 - K$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

6.2.31.2 Geometry record

These records describe a geometry within the engagement space. They may also describe the method by which that geometry shall be dead reckoned and the parameters used in that dead reckoning. If there is only one geometry record in an Environmental Process PDU, it shall be the first environmental record attached to the PDU. If there are multiple geometry records in an Environmental Process PDU, then the first environmental record attached to the PDU shall be a bounding sphere geometry record. The bounding sphere is a sphere that encloses all the geometries described by this environmental process. The bounding sphere shall not be dead reckoned. The bounding sphere record is intended to be used for filtering by receiving applications. See [UID 251-267] for Geometry record definitions.

6.2.31.3 State record

These records specify the value of environmental parameters within the specified volume or volumes. Each state record shall follow the geometry record that describes the region of influence for the parameters contained in that state record. They may also describe the method by which that parameter shall be dead reckoned and the parameters used in that dead reckoning. If an environmental process is associated with a particular entity within the simulation, then the associated Entity Identifier may be included within a state record. The state record may also contain the update rate for the environmental model it represents. See [UID 268-269] for State record definitions.

6.2.32 Euler Angles record

Orientation of a simulated entity shall be specified by the Euler Angles record. This record shall specify three angles as described in Figure 3 and 3.1. These angles shall be specified with respect to the entity's coordinate system. The three angles shall each be specified by a 32-bit floating point number representing radians. The format of the Euler Angles record shall be as shown in Table 58.

Table 58—Euler Angles record

Field size (bits)	Field name	Data type
32	Psi (ψ)	32-bit floating point
32	Theta (θ)	32-bit floating point
32	Phi (ϕ)	32-bit floating point
Total Euler Angles record size = 96 bits		

6.2.33 Event Identifier record

6.2.33.1 General

An event is defined as some condition involving one or more PDUs that a simulation application desires to document. This is accomplished by assigning a specific Event ID to the PDU or PDUs associated with the event. Event IDs are solely for data analysis and shall not be used to determine the validity of a PDU as to whether or not it should be processed. No PDU shall be considered invalid and not processed based on an erroneous or unexpected Event ID value. A simulation application may log such an occurrence for later analysis. There is no continuity requirement to continue an Event ID number sequence in the event a simulation application is no longer an active participant in an exercise. However, nothing in this standard precludes a simulation application from having the capability to maintain continuity of Event ID numbers should it re-enter an exercise.

6.2.33.2 Event identification

Events shall be specified by the Event Identifier (Event ID) record. Event identification is the association of an event with one or more PDUs that are transmitted by a specific simulation application. The only PDUs that are required to have a nonzero Event ID are shown in Table 61. All other uses of the Event ID are optional and at the discretion of the simulation application or as required by exercise agreements. When the Event ID is not required, all fields of the Event ID record for a PDU shall be set to NO_VALUE. A specific event is initiated by the transmission of the first PDU that contains a new Event ID and ends when the last PDU for the same Event ID is transmitted. The same Event ID representing a specific event may be included in various PDU types and for various Entity IDs if they are all related to the same event. The only restriction is that duplicate, concurrent Event IDs are not allowed. For example, if Event ID 3 is being used to associate all the types of PDUs for local entities involved in an engagement, then the same Event ID 3 cannot be used by the same simulation application for another event that is happening simultaneously. An Event ID can be reused by a simulation once all Event IDs have been cycled through.

6.2.33.3 Record format

The Event Identifier record shall consist of a Simulation Address record and a 16-bit unsigned integer specifying the event number:

a) The Event Identifier record shall have two formats. Format A is shown in Table 59 and applies to all PDUs except the Live Entity (LE) PDU. Format B applies to the Live Entity PDU and is shown in Table 60.

Table 59—Event Identifier record—format A

Field size (bits)	Field name	Data type
32	Simulation Address	Site number—16-bit unsigned integer
		Application number—16-bit unsigned integer
16	Event Number	16-bit unsigned integer
Total Event Identifier record – format A size = 48 bits		

Table 60—Event Identifier record—format B

Field size (bits)	Field name	Data type
16	Simulation Address	Site number—8-bit unsigned integer
		Application number—8-bit unsigned integer
16	Event Number	16-bit unsigned integer
Total Event Identifier record – format B size = 32 bits		

b) The event number shall be assigned by the simulation application that initiates an event as follows:

   1) The first event number used by a simulation application when the application is initially started or re-initialized for an exercise shall be any arbitrary number defined by the application. It shall be incremented by one to the next greatest number for each new event that is needed.

   2) Event numbers shall be obtained from a single master Event Number list within the simulation application. Event numbers shall be able to be assigned incrementally in 1 unit increments in the range from 1 to 65 535 decimal.

   3) In the case where all possible values are exhausted, the numbers shall be reused beginning at the first currently unused lowest number.

c) A sufficient range of numbers shall be allocated to minimize the reuse of an event number during an exercise. The maximum event number used may be less than the maximum number that can be contained in the Event Number field.

d) If a munitions entity is created by a simulation application other than the simulation application that issued the Fire PDU, the simulation application for the munitions entity shall set the Event ID in the Detonation PDU to the same value as the Event ID from the associated Fire PDU.

e) If a munitions entity is transferred after initiation to another simulation application, the simulation application that assumes ownership shall set the Event ID in the Detonation PDU to the same value as the Event ID from the associated Fire PDU. (In this case, the simulation address portion of the Event ID will reflect the original simulation application and not the originator of the Detonation PDU.) See 5.9.4 Transfer Ownership for how this is accomplished.

f) Table 61 lists all the PDUs that have an Event Identifier record and those that require the presence of a nonzero Event ID.

Table 61—PDUs with the Event ID field

PDU	Required Event IDs
Collision PDU	None
Collision Elastic PDU	None
DE Fire PDU	Same as Fire PDU
Detonation PDU	Same as Fire PDU
EE PDU	None
Fire PDU	Same as Detonation PDU and DE Fire PDU
IFF PDU	None
LE Fire PDU	Same as LE Detonation PDU
LE Detonation PDU	Same as LE Fire PDU
UA PDU	None

6.2.34 Exercise identifier

Exercise identification shall be specified by an 8-bit unsigned integer value. This value shall be unique to each exercise occurring simultaneously on the same communications medium [see item b) in 6.2.66].

6.2.35 Expendable record

An entity's expendable (e.g., chaff and flares) information shall be represented by one or more Expendable records. For each type or location of expendable, this record shall specify the type, location, and quantity of expendables that an entity contains.

This record may be used (optionally) in a transfer transaction to send internal state data from the divesting simulation to the acquiring simulation (see 5.9.4). This record may also be used for other purposes.

The Expendable record can be used to represent either non-location-specific or location-specific expendable information, as follows:

— *Nonspecific Stations*. One Expendable record is used for each type of expendable when stations/launchers are not specified (i.e., the *Station/Launcher* field is zero).

— *Specific Stations*. One Expendable record is used to indicate the quantity of an expendable type located at one specific station/launcher when the *Station/Launcher* field value is nonzero.

Expendable records associated with an entity shall be indicated with either Nonspecific Stations or Specific Stations. There shall be no mixing of the two types in a given transfer.

The Expendable record shall contain the following fields:

a) *Expendable Type*. This field shall identify the entity type of the expendable. This field shall be represented by an Entity Type record (see 6.2.30).

b) *Station/Launcher*. This field shall indicate the station or launcher to which the expendable is assigned and shall be represented by a 32-bit enumeration. A zero value shall indicate that this Expendable record is not associated with any particular station/launcher. To specify a particular station/launcher, the enumerations represented by the Parameter Type field of either an attached or articulated part shall be used. See 6.2.94.2, 6.2.94.3, and Annex I.

c) *Quantity*. This field shall identify the quantity remaining of this expendable and shall be represented by a 16-bit unsigned integer. When the Station/Launcher field is set to zero, the quantity represents the total quantity of this expendable type for this entity. Otherwise, this field indicates the quantity of the expendable type at the specified station/launcher.

d) *Expendable Status*. This field shall specify the status of the expendable. It shall be represented by an 8-bit enumeration (see [UID 327]).

The format of the Expendable record shall be as shown in Table 62.

Table 62—Expendable record

Field size (bits)	Field name	Data type
64	Expendable Type	Entity Type record (see 6.2.30)
32	Station/Launcher	32-bit enumeration
16	Quantity	16-bit unsigned integer
8	Expendable Status	8-bit enumeration
8	Padding	8 bits unused
Total Expendable record size = 128 bits		

6.2.36 Expendable Reload record

An entity's expendable (e.g., chaff and flares) information shall be represented by one or more Expendable Reload records. For each type or location of expendable, this record shall specify the type, location, and quantity of expendables that an entity contains.

This record shall be required to be sent if one or more Expendable records are included in a transfer transaction. One Expendable Reload record shall be sent for each Expendable record that is sent and shall mirror the type of station representation (nonspecific or specific) found in that record. This record may also be used for other purposes. This record is optional for receipt. If processed, the data may be used or ignored as deemed appropriate by the simulation (e.g., an instant reload feature would override reload times, and an infinite quantity feature would override quantity values).

The Expendable Reload record can be used to represent either non-location-specific or location-specific expendable information, as follows:

— *Nonspecific Stations*. One Expendable Reload record is used for each type of expendable when stations/launchers are not specified (i.e., the Station/Launcher field is zero).

— *Specific Stations*. One Expendable Reload record is used to indicate the standard and maximum quantities and reload times of an expendable type located at one specific station/launcher when the Station/Launcher field value is nonzero.

Expendable Reload records associated with an entity shall be indicated with either Nonspecific Stations or Specific Stations. There shall be no mixing of the two types in a given transfer.

The reload times may vary for different stations/launchers that can accept a specific expendable type, so only an overall reload time is given for loading the standard or maximum quantity at either a station/launcher or for the entire entity, whichever is specified. How quantities and reload times are calculated is beyond the scope of this standard.

The Expendable Reload record shall contain the following fields:

a) *Expendable Type*. This field shall identify the entity type of the expendable. This field shall be represented by an Entity Type record (see 6.2.30).

b) *Station/Launcher*. This field shall indicate the station or launcher to which the expendable is assigned and shall be represented by a 32-bit enumeration. A zero value shall indicate that this Expendable Reload record is not associated with any particular station/launcher. To specify a particular station/launcher, the enumerations represented by the Parameter Type field of either an attached or articulated part shall be used. See 6.2.94.2, 6.2.94.3, and Annex I.

c) *Standard Quantity*. This field shall identify the standard quantity of this expendable type normally loaded at this station/launcher if a station/launcher is specified. If the Station/Launcher field is set to zero, then this is the total quantity of this expendable type that would be present in a standard reload of all applicable stations/launchers associated with this entity. This field shall be represented by a 16-bit unsigned integer.

d) *Maximum Quantity*. This field shall specify the maximum quantity of this expendable type that this station/launcher is capable of holding when a station/launcher is specified. This would be the value used when a maximum reload was desired to be set for this station/launcher. If the Station/Launcher field is set to zero, then this is the maximum quantity of this expendable type that would be present on this entity at all stations/launchers that can accept this expendable type. This field shall be represented by a 16-bit unsigned integer.

e) *Standard Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the standard quantity of this expendable type at this specific station/launcher. When the Station/Launcher field is set to zero, this shall be the number of seconds of simulation time required to perform a standard quantity reload of this expendable type at all applicable stations/launchers for this entity. This field shall be represented by a 32-bit unsigned integer.

f) *Maximum Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the maximum possible quantity of this expendable type at this station/launcher. When the Station/Launcher field is set to zero, this shall be the the number of seconds of simulation time required to perform a maximum quantity load/reload of this expendable type at all applicable stations/launchers for this entity. This field shall be represented by a 32-bit unsigned integer.

The format of the Expendable Reload record shall be as shown in Table 63.

Table 63—Expendable Reload record

Field size (bits)	Field name	Data type
64	Expendable Type	Entity Type record (see 6.2.30)
32	Station/Launcher	32-bit enumeration
16	Standard Quantity	16-bit unsigned integer

Table 63—Expendable Reload record *(continued)*

Field size (bits)	Field name	Data type
16	Maximum Quantity	16-bit unsigned integer
32	Standard Quantity Reload Time	32-bit unsigned integer
32	Maximum Quantity Reload Time	32-bit unsigned integer
Total Expendable Reload record size = 192 bits		

6.2.37 Fixed Datum record

Fixed datum information shall be represented using the Fixed Datum record. This record shall specify the fixed datum type and the value for that fixed datum type. The fields of this record are as follows:

a) *Fixed Datum ID*. This field shall specify the type of fixed datum to be communicated. This field shall consist of a 32-bit enumeration (see [UID 66]).

b) *Fixed Datum Value.* This field shall specify the value for a particular Fixed Datum ID and shall be fixed at 32 bits in length. The field may be used to convey one 8-, 16-, or 32-bit data value as specified by the Fixed Datum ID. Padding bits shall follow an 8- or 16-bit value to fill the 32-bit field.

The format of the Fixed Datum record shall be as shown in Table 64.

Table 64—Fixed Datum record

Field size (bits)	Field name	Data type
32	Fixed Datum ID	32-bit enumeration
32	Fixed Datum Value	32 bits (varies)
Total Fixed Datum record size = 64 bits		

6.2.38 Force identification

This field shall distinguish the different teams or sides in a DIS exercise and shall be specified by an 8-bit enumeration (see [UID 6]).

6.2.39 Fundamental Operational Data record

The identification of certain basic operational data for an IFF system shall be specified by a Fundamental Operational Data record. This record consists of a fixed field format. The System Status and Information Layers fields are required to be implemented by all system types. Data Field 1, Data Field 2, and Parameter 1 through 6 fields are system-specific fields that apply to one or more, but not all, system types. The records for these fields are contained in either 6.2 or B.2 depending on the type of record. Clause B.5 lists the IFF systems, and each system description contains an Information Content paragraph that indicates which fields of the Fundamental Operational Data record are required to be implemented and the associated record, if any, for each field.

The fields of this record shall be as follows:

a) *System Status*. This field shall provide system status information. It shall contain the System Status record (see B.2.52).

b) *Data Field 1*. This field is a system-specific field whose content shall depend on the system type. The data type for this field will depend on the use of this field by a system type and may contain a single field value or a record.

c) *Information Layers*. This field shall specify the information layers that are present. It shall contain the Information Layers record (see 6.2.45).

d) *Data Field 2*. This field is a system-specific field whose content shall depend on the system type (see Annex B). The data type for this field will depend on the use of this field by a system type.

e) *Parameters 1 to 6*. These fields are system-specific fields whose content shall depend on the system type (see Annex B). The data type for these fields will depend on the use of this field by a system type.

The format of the Fundamental Operational Data record shall be as shown in Table 65.

Table 65—Fundamental Operational Data record

Field size (bits)	Field name	Data type
8	System Status	System Status record (see B.2.52)
8	Data Field 1	8 bits defined by system type
8	Information Layers	8-bit record
8	Data Field 2	8 bits defined by system type
16	Parameter 1	16 bits defined by system type
16	Parameter 2	16 bits defined by system type
16	Parameter 3	16 bits defined by system type
16	Parameter 4	16 bits defined by system type
16	Parameter 5	16 bits defined by system type
16	Parameter 6	16 bits defined by system type
Total Fundamental Operational Data record size = 128 bits		

6.2.40 Grid Axis Descriptor record

The Grid Axis Descriptor record provides detailed information about each grid axes and grid location coordinates for environmental state variables for use in the Gridded Data PDU. This record provides for the representation of grids with constant spacing or variable spacing. This record shall contain information about the coordinate of the origin and endpoint of the axis, number of grid points along the axis, interleaf factor, axis type, number of axis grid points and coordinate scale factor, coordinate offset value, and grid location coordinates.

In the following descriptions, the domain shall refer to the entire volume for which gridded data are to be transmitted. This region shall be defined by specifying the origin and endpoint for each grid axis. Use of x, y, and z to denote the three coordinates of a grid point is for convenience and does not imply the sole use of

Cartesian coordinates. A more general coordinate representation (x_1, x_2, x_3) or x_i is used in the tables to provide generality. As a region may be sufficiently large to require distribution via multiple Gridded Data PDUs, any reference to PDU refers to data contained only within the current Gridded Data PDU. Volumetric gridded data shall be transmitted over multiple Gridded Data PDUs with each PDU containing data for subsets of the volume selected by horizontal, vertical, or volume sampling.

The Grid Axis Descriptor record shall contain the following fields:

a) *Domain Initial* x_i. This field shall specify the coordinate of the origin (or initial value) for the x_i axis for the environmental state variable data contained within the bounding domain. This field shall be represented by a 64-bit floating point number.

b) *Domain Final* x_i. This field shall specify the coordinate of the endpoint (or final value) for the x_i axis for the environmental state variable data contained within the bounding domain. This field shall be represented by a 64-bit floating point number.

c) *Domain Points* x_i. This field shall specify the number of grid points along the x_i domain axis for the environmental state variable data. This number may exceed the number of points along an axis for a single Gridded Data PDU if there are multiple PDUs being issued to contain all the data. This field shall be represented by a 16-bit unsigned integer.

d) *Interleaf Factor*. This field shall specify the integer valued interleaf factor along a domain (grid) axis. A value of one shall indicate no subsampling (interleaving), while integer values greater than one shall indicate the sampling frequency along an axis. This field shall be represented by an 8-bit unsigned integer.

e) *Axis Type*. This field shall specify the type of grid axis represented with the Grid Axis Descriptor record. Setting this field to Regular Axis (0) shall indicate a grid axis with constant grid spacing, while setting it to Irregular Axis (1) shall indicate a grid axis with variable grid spacing. This field shall be represented by an 8-bit enumeration (see [UID 377]).

f) *Regular (Fixed Spacing) Axis Data*

    1) Number of Points on x_i Axis. This field shall specify the number of grid locations along the x_i axis for the environmental state variable data contained within the current PDU. This field shall be represented by a 16-bit unsigned integer.

    2) Initial Index x_i. This field shall specify the index of the initial grid point for the current PDU along the x_i domain axis. A value of zero shall indicate that the PDU grid and the domain grid have the same initial point. This field shall be represented by a 16-bit unsigned integer.

g) *Irregular (Variable Spacing) Axis Data*

    1) Number of Points on x_i Axis. This field shall specify the number of grid locations along the x_i axis for the environmental state variable data contained within the current PDU. This field shall be represented by a 16-bit unsigned integer.

    2) Initial Index x_i. This field shall specify the index of the initial grid point for the current PDU along the x_i domain axis. A value of zero shall indicate that the PDU grid and the domain grid have the same initial point. This field shall be represented by a 16-bit unsigned integer.

    3) Coordinate Scale. This field shall specify the value that linearly scales the coordinates of the grid locations for the x_i axis. This field shall be represented by a 64-bit floating point number.

    4) Coordinate Offset. This field shall specify the constant offset value that shall be applied to the grid locations for the x_i axis (for example, the Earth's radius is the *z*-axis offset). The default value shall be zero. This field shall be represented by a 64-bit floating point number.

    5) $x_i(N_i)$ Values. This field shall specify the coordinate values for the N_i grid locations along the irregular (variable spacing) x_i axis for environmental data values contained within the PDU. These fields shall be represented by a 16-bit unsigned integer.

    6) Padding. This space is reserved to bring the record length to a 64-bit boundary.

The format of the Grid Axis Descriptor record shall be as shown in Table 66 and Table 67.

Table 66—Grid Axis Descriptor record for regular (fixed spacing) axis data

Field size (bits)	Field name	Data type
64	Domain Initial x_i	64-bit floating point
64	Domain Final x_i	64-bit floating point
16	Domain Points x_i	16-bit unsigned integer
8	Interleaf Factor	8-bit unsigned integer
8	Axis Type	8-bit enumeration
16	Number of Points on x_i Axis (N_i)	16-bit unsigned integer
16	Initial Index x_i	16-bit unsigned integer
Total Grid Axis Descriptor record for regular (fixed spacing) axis data size = 192 bits		

Table 67—Grid Axis Descriptor record for irregular (variable spacing) axis data

Field size (bits)	Field name	Data type
64	Domain Initial x_i	64-bit floating point
64	Domain Final x_i	64-bit floating point
16	Domain Points x_i	16-bit unsigned integer
8	Interleaf Factor	8-bit unsigned integer
8	Axis Type	8-bit enumeration
16	Number of Points on x_i Axis (N_i)	16-bit unsigned integer
16	Initial Index x_i	16-bit unsigned integer
64	Coordinate Scale x_i	64-bit floating point
64	Coordinate Offset x_i	64-bit floating point
16	$x_i(1)$ Values	16-bit unsigned integer
⋮		

Table 67—Grid Axis Descriptor record for irregular (variable spacing) axis data *(continued)*

Field size (bits)	Field name	Data type
16	$x_i(N_i)$ Value	16-bit unsigned integer
$8P_i$	Padding	Padding to 64-bit boundary—P_i octets
Total Grid Axis Descriptor record for irregular (variable spacing) axis data size = 320 + $16N_i$ + $8P_i$ bits where N_i is the number of points on the x_i axis P_i is the number of padding octets on the x_i axis, which is $2(\lceil N_i/4 \rceil 4 - N_i)$ $\lceil x \rceil$ is the largest integer < x + 1.		

6.2.41 Grid Data record

The Grid Data record contains the actual environmental state variable data for each grid location. The record specifies the data sample type, the format by which the data are represented, field scale factor and offset values, and the actual data values in 8-bit, 16-bit, or 32-bit values as a function of the representation field. The Grid Data record shall contain the following fields:

a) *Sample Type*. This field shall specify the environmental data sample contained in the PDU (e.g., *u*-component of wind). This field shall be represented by a 16-bit enumeration (see [UID 246]).

b) *Data Representation*. This field shall specify the value that describes the data representation method for the environmental state variable data contained within the Data Values fields of this record. This field shall be represented by a 16-bit enumeration (see [UID 247]).

c) *Data Representation Type = Type 0 (0)*

   1) Number of Octets. This field shall specify the number of octets of environmental state variable data values contained in this record. This field shall be represented by a 16-bit unsigned integer.

   2) Data Values. This field shall specify the environmental state variable data values. The data shall be represented as a stream of octets, the interpretation of which shall be agreed to prior to the start of the exercise.

   3) Padding. This space is reserved to bring the record length to a 16-bit boundary.

   NOTE—Data Representation Type 0 is specifically designed to be flexible to accommodate anything from character data to data expressed as floating point numbers.

d) *Data Representation Type = Type 1 (1)*

   1) Field Scale. This field shall specify the constant scale factor used to scale the environmental state variable data values contained in this record. This field shall be represented by a 32-bit floating point number.

   2) Field Offset. This field shall specify the constant offset used to scale the environmental state variable data values contained in this record. This field shall be represented by a 32-bit floating point number.

   3) Number of Values. This field shall specify the number of environmental state variable data values contained in this record. This field shall be represented by a 16-bit unsigned integer.

   4) Data Values. This field shall specify the environmental state variable data values. The data in this field shall be represented by a 16-bit unsigned integer.

   5) Padding. This space is reserved to bring the record length to a 32-bit boundary.

e) *Data Representation Type = Type 2 (2)*
   1) Number of Values. This field shall specify the number of environmental state variable data values contained in this record. This field shall be represented by a 16-bit unsigned integer.
   2) Data Values. This field shall specify the environmental state variable data values. The data in this field shall be represented by a 32-bit floating point number.

The merged format of the Grid Data record shall be as shown in Table 68, Table 69, and Table 70. In the actual implementation, Sample Type and Data Representation are united with the data representation array.

Table 68—Grid Data record for representation type 0

Field size (bits)	Field name	Data type
16	Sample Type	16-bit enumeration
16	Data Representation	16-bit enumeration
16	Number of Octets (*N*)	16-bit unsigned integer
8	Data Value #1	8-bit unsigned integer
	• • •	
8	Data Value #*N*	8-bit unsigned integer
8*P*	Padding	Padding to 16-bit boundary—*P* octets
Total Grid Data record (representation 0) size = $48 + 8N + 8P$ bits where N is the number of values *P* is the number of padding octets, which is $\lceil N/2 \rceil 2 - N$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

Table 69—Grid Data record for representation type 1

Field size (bits)	Field name	Data type
16	Sample Type	16-bit enumeration
16	Data Representation	16-bit enumeration
32	Field Scale	32-bit floating point
32	Field Offset	32-bit floating point
16	Number of Values (*N*)	16-bit unsigned integer
16	Data Value #1	16-bit unsigned integer

Table 69—Grid Data record for representation type 1 *(continued)*

Field size (bits)	Field name	Data type
	• • •	
16	Data Value #N	16-bit unsigned integer
$8P$	Padding	Padding to 32-bit boundary—P octets
Total Grid Data record (representation 1) size = 112+ $16N + 8P$ bits where N is the number of values P is the number of padding octets, which is $2(\lceil (N+1)/2 \rceil 2 - (N+1))$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

Table 70—Grid Data record for representation type 2

Field size (bits)	Field name	Data type
16	Sample Type	16-bit enumeration
16	Data Representation	16-bit enumeration
16	Number of Values (N)	16-bit unsigned integer
16	Padding	16 bits unused
32	Data Value #1	32-bit floating point
	• • •	
32	Data Value #N	32-bit floating point
Total Grid Data record (representation 2) size = 64+ $32N$ bits where N is the number of values		

6.2.42 Group identifier

The unique designation of a group of entities contained in the IsGroupOf PDU shall be specified by the Group Identifier record. The Group Identifier (Group ID) does not represent an entity but rather a group of entities for the purpose of conserving bandwidth. This group is not an aggregation, and no interactions related to the group represented by a Group Identifier shall be allowed. The Group Identifier shall not be included in any PDU field that specifies an Entity Identifier. If an IsGroupOf PDU is processed by a simulation, it shall not create an active entity based on the Group ID. A simulation shall only process information in the PDU and interact with the entities listed in the IsGroupOf PDU if they exist as separate entities derived from the receipt of the Entity State or Entity State Update PDU. An entity that is contained in an Aggregation State PDU shall not be allowed to be an entity in the IsGroupOf PDU and vice versa.

The Group Identifier is an object identifier. General object identifier requirements are specified in 4.2.5.2.

The format of the Group Identifier record shall be as shown in Table 71.

Table 71—Group Identifier record

Field size (bits)	Field name	Data type
32	Simulation Address	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
16	Group Number	16-bit unsigned integer
Total Group Identifier record size = 48 bits		

6.2.43 IFF Data Specification record

The IFF Data Specification record shall be used when variable records are required to be included in a layer format for Layers 3 through 7. It is identical to a Standard Variable Specification record format except that alignment is to a 32-bit boundary for each IFF Data record instead of to a 64-bit boundary. Due to lack of 64-bit alignment in the IFF PDU, new records designed to conform to the IFF Data Specification record shall not contain 64-bit float or 64-bit integer values.

The IFF Data Specification record shall contain the following fields:

a) *Number of IFF Data Records*. This field shall specify the number of IFF Data records and shall be represented by a 16-bit unsigned integer. If no records are present for an issuance of the PDU, this field shall be set to zero and no records shall be included.

b) The IFF data record shall contain the following fields:

    1) Record Type. This field shall indicate the unique record number assigned to this IFF Data record. It shall be represented by a 32-bit enumeration (see [UID 66]).

    2) Record Length. This field shall indicate the record length expressed as the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 32-bit boundary. All Record Length values shall be a multiple of 4. The Record Length shall be represented by a 16-bit unsigned integer.

    3) Record-Specific fields. These are the data fields of the record. Any number and types of data fields may be included.

    4) Padding. Padding shall be explicitly included in each record as necessary to make the record length a multiple of 4 octets (32 bits) so that any record that follows is automatically aligned.

NOTE 1—The record length requirement can be achieved by placing padding fields anywhere in the IFF Data record and not necessarily at the end of the record.

NOTE 2—The IFF Data Specification record format does not include padding between IFF Data records because those records are always a multiple of 4 octets in length.

The format of the IFF Data Specification record shall be as shown in Table 72.

Table 72—IFF Data Specification record

Field size (bits)	Field name	Data type
16	Number of IFF Data Records (*N*)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IFF Data record #*1*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_1 + P_1)$
		Record-Specific fields—K_1 octets
		Padding to 32-bit boundary—P_1 octets
• • •		
$48 + 8K_N + 8P_N$	IFF Data record #*N*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_N + P_N)$
		Record-Specific fields—K_N octets
		Padding to 32-bit boundary—P_N octets

IFF Data Specification record size = $16 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

N is the number of IFF Data records
K_i is the length of the Record-Specific field in IFF Data record #*i* in octets
P_i is the number of padding octets in IFF Data record #*i*, which is $\lceil (6 + K_i)/8 \rceil 8 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

6.2.44 IFF Fundamental Parameter Data record

The specification of the fundamental energy radiation characteristics of a transponder or interrogator system emission shall be communicated by an IFF Fundamental Parameter Data record. The fields of this record shall be common to all systems except for the System-Specific Data field. The System-Specific Data field shall be used to convey unique data applicable to a specific system type. The fields of this record are as follows:

a) *ERP*. This field shall specify the average peak radiated power for the emission in dBm. This field shall be represented by a 32-bit floating point number.

b) *Frequency*. This field shall specify the center frequency of the emission in Hertz. Frequency modulation for a particular emitter and mode shall be derived from database parameters stored in the receiving entity. This field shall be represented by a 32-bit floating point number.

c) *PgRF*. When applied to originators, this field shall specify the number of interrogations per second emitted. This field shall contain zero when applied to responder (i.e., transponder) systems. This field shall be represented by a 32-bit floating point number.

d) *Pulse Width.* This field shall specify the duration in microseconds of the fundamental pulse of which the interrogation or reply is composed. This field shall be specified by a 32-bit floating point number.

e) *Burst Length.* This field shall specify the number of emissions generated in a single burst. This field shall contain zero for continuously emitting systems and shall contain the value one for responders. This field shall be represented as a 32-bit unsigned integer.

f) *Applicable Modes.* This field shall specify the modes to which the fundamental parameter data apply. This field shall be represented by an 8-bit enumeration (see [UID 339]).

g) *System Specific Data*. This field shall be used for system specific data associated with the system type. This field shall be represented by 24 bits as defined by for a specific system type. The System-Specific Data formats for applicable system types are contained in Annex B.

The format of the IFF Fundamental Parameter Data record shall be as shown in Table 73.

Table 73—IFF Fundamental Parameter Data record

Field size (bits)	Field name	Data type
32	ERP	32-bit floating point
32	Frequency	32-bit floating point
32	PgRF	32-bit floating point
32	Pulse Width	32-bit floating point
32	Burst Length	32-bit unsigned integer
8	Applicable Modes	8-bit enumeration
24	System-Specific Data	24 bits defined by system type
Total IFF Fundamental Parameter Data record size = 192 bits		

6.2.45 Information Layers record

The Information Layers record is applicable to the IFF PDU and is used for two purposes in that PDU as follows:

a) *Layer 1—Fundamental Operational Data Record*. The Information Layers record contained in the Fundamental Operational Data record indicates which information layers (layers) are present in an IFF PDU. In this case, each Layer 1 through 7 field shall be set to either Not Present (0) or Present (1).

b) *Layer 5—Applicable Layers Field*. The Information Layers record contained in the Applicable Layers field of Layer 5 indicates to which layer(s) the IFF data records present in Layer 5 are applicable. In this case, each Layer 1 through 7 field shall be set to either Not Applicable (0) or Applicable (1).

The format of the Information Layers record shall be as shown in Table 74.

Table 74—Information Layers record

Field name	Bit	Value
Not used	0	1 bit unused
Layer 1	1	Enumeration
Layer 2	2	Enumeration
Layer 3	3	Enumeration
Layer 4	4	Enumeration
Layer 5	5	Enumeration
Layer 6	6	Enumeration
Layer 7	7	Enumeration
Total Information Layers record size = 8 bits		

6.2.46 Intercom Communications Parameters record

Additional parameters to the Intercom Control PDU are represented in Intercom Communications Parameters records. An Intercom Communications Parameters record shall be padded to a multiple of 32 bits. The fields of the Intercom Communications Parameters record shall consist of the following: Record Type, Record Length, and multiple Record-Specific fields representing the Intercom Parameters. These fields are described as follows:

a) *Record Type.* This field shall specify the type of intercom parameters contained in the Record-Specific fields. This field shall be represented by a 16-bit enumeration (see [UID 185]).

b) *Record Length.* This field shall specify the total length of the Record-Specific fields in octets. This field shall be represented by a 16-bit unsigned integer.

c) *Record-Specific fields.* These variable-number, variable-length fields shall specify additional intercom communications parameters. The length and content of each record is a function of the Record Type field.

d) *Padding.* This space is reserved to bring the record length into alignment as specified in 6.3.2.

The format of the Intercom Communications Parameter record shall be as shown in Table 75.

Table 75—Intercom Communication Parameters record

Field size (bits)	Field name	Data type
16	Record Type	16-bit enumeration
16	Record Length	16-bit unsigned integer (*R*)

Table 75—Intercom Communication Parameters record *(continued)*

Field size (bits)	Field name	Data type
$8R$	Record-Specific fields	R octets
$8P$	Padding	Padding to 32-bit boundary—P octets
Total Intercom Communication Parameters record size = 32+ $8R$ + $8P$ bits where R is the length of the Record-Specific fields in octets P is the number of padding octets, which is $\lceil R/4 \rceil 4 - R$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

6.2.47 Intercom Identifier record

The unique designation of an attached or unattached intercom in an event or exercise when using the Intercom Method shall be specified by the Intercom Identifier record. The Intercom Identifier record shall consist of the Intercom Reference ID field and the Intercom Number field. The format of the Intercom Identifier record shall be as shown in Table 76.

Table 76—Intercom Identifier record

Field size (bits)	Field name	Data type
48	Intercom Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Intercom Number	16-bit unsigned integer
Total Intercom Identifier record size = 64 bits		

6.2.48 IO records

6.2.48.1 General

IO records shall be used to communicate additional information about information operations. IO records use the variable record format specified in the Standard Variable Specification record (see 6.2.83). Each IO record is defined in this subclause. New IO records may be defined at some future date as the characteristics of information operations evolve.

6.2.48.2 IO Communications Node record

A communications node entity that is part of a simulated communications network shall be identified by an IO Communications Node record.

The fields of this record are as follows:

a) *Record Type*. This field shall identify the type as the IO Communications Node record and shall be specified by a 32-bit enumeration.

b) *Record Length*. This field shall indicate the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 64-bit boundary. The Record Length shall be a multiple of 8. The Record Length shall be represented by a 16-bit unsigned integer.

c) *Communications Node Type*. This field shall indicate the present type of communications node (sender, receiver, etc.). It shall be represented by an 8-bit enumeration (see [UID 294]).

d) *Communications Node ID*. This field shall indicate the communications node identifier for the entity that is part of a simulated communications network. It shall be represented by a Communications Node ID record (see 6.2.48.4).

The format of the IO Communications Node record shall be as shown in Table 77.

Table 77—IO Communications Node record

Field size (bits)	Field name	Data type
32	Record Type = 5501	32-bit enumeration
16	Record Length = 16	16-bit unsigned integer
8	Communications Node Type	8-bit enumeration
8	Padding	8 bits unused
64	Communications Node ID	Entity ID—48-bit record (see 6.2.28)
		Element ID—16-bit unsigned integer
Total IO Communication Node record size = 128 bits		

6.2.48.3 IO Effect record

The identification of IO effects on an entity when calculated by an IO simulation shall be communicated using the IO Effect record.

The fields of this record are as follows:

a) *Record Type*. This field shall identify the type as the IO Effect record and shall be specified by a 32-bit enumeration.

b) *Record Length*. This field shall indicate the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 64-bit boundary. The Record Length shall be a multiple of 8. The Record Length shall be represented by a 16-bit unsigned integer.

c) *IO Status*. This field shall indicate whether the IO effect has an effect on the sender, receiver, message(s), or some combination of them. This field shall be represented by an 8-bit enumeration (see [UID 290]).

d) *IO Link Type*. This field shall indicate the IO link type as a logical or physical link or node. This field shall be represented by an 8-bit enumeration (see [UID 291]).

e) *IO Effect*. This field shall indicate the IO effect associated with this IO attack. This field shall be represented by an 8-bit enumeration (see [UID 292]).

f) *IO Effect Duty Cycle*. This field shall indicate the IO effect duty cycle represented as a percentage in the range of 0% to 100% in 1% increments. This field shall be represented by an 8-bit unsigned integer.

g) *IO Effect Duration*. This field shall indicate the duration of the IO effect in seconds, from 1 s to 65 534 s. It shall be set to indicate IO_UNTIL_FURTHER_NOTICE (65 535) if the duration has no fixed simulation time interval. (See 6.1.8 for parameter details and default values.) It shall be set to zero only if the IO Effect field is set to Terminate Effect (255). This field shall be represented by a 16-bit unsigned integer.

h) *IO Process*. This field shall indicate the IO process being performed. This field shall be represented by a 16-bit enumeration (see [UID 293]).

The format of the IO Effect record shall be as shown in Table 78.

Table 78—IO Effect record

Field size (bits)	Field name	Data type
32	Record Type = 5500	32-bit enumeration
16	Record Length = 16	16-bit unsigned integer
8	IO Status	8-bit enumeration
8	IO Link Type	8-bit enumeration
8	IO Effect	8-bit enumeration
8	IO Effect Duty Cycle	8-bit unsigned integer
16	IO Effect Duration	16-bit unsigned integer
16	IO Process	16-bit enumeration
16	Padding	16 bits unused
Total IO Effect record size = 128 bits		

6.2.48.4 Communications Node ID record

The identification of a communications node shall be conveyed using the Communications Node ID record. The combination of Entity ID and Element ID uniquely identify an IO communications node within a simulation exercise.

The fields of this record are as follows:

a) *Entity ID*. This field shall identify the entity containing the communications node. This field shall be represented by an Entity Identifier record (see 6.2.28).

b) *Element ID*. This field shall identify a particular communications element (e.g., radio, network component, or modem) within a given entity. If this element is a radio for which a Transmitter PDU is being issued, this value shall correspond to the Radio Number in that PDU. The Element ID field shall be represented by a 16-bit unsigned integer.

The format of the Communications Node ID record shall be as shown in Table 79.

Table 79—Communications Node ID record

Field size (bits)	Field name	Data type
48	Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Element ID	16-bit unsigned integer
Total Communication Node ID record size = 64 bits		

6.2.49 Jamming Technique record

The methods used to conduct electronic warfare shall be specified by a Jamming Technique record (see 5.7.3.8). This record shall consist of a Kind, Category, Subcategory, and Specific indication of the technique (see [UID 284]). The fields of this record are as follows:

a) *Kind*. This field shall be represented by an 8-bit enumeration and indicates the broadest category of jamming.

b) *Category*. This field shall be represented by an 8-bit enumeration and permits more detailed definition of techniques defined by the Kind.

c) *Subcategory*. This field shall be represented by an 8-bit enumeration and permits more detailed definition of techniques defined by the Kind and Category.

d) *Specific*. This field shall be represented by an 8-bit enumeration and provides a means for high-fidelity electronic warfare emitters to indicate the most detailed definition of techniques.

The format of the Jamming Technique record shall be as shown in Table 80.

Table 80—Jamming Technique record

Field size (bits)	Field name	Data type
8	Kind	8-bit enumeration
8	Category	8-bit enumeration
8	Subcategory	8-bit enumeration
8	Specific	8-bit enumeration
Total Jamming Technique record size = 32 bits		

6.2.50 Launched Munition record

A launched munition entity's firing and targeting information shall be represented by a Launched Munition record. This record shall specify the munition's fire event ID, firing entity ID, target entity ID, and/or target location.

This record may be used in a transfer transaction to send internal state data from the divesting simulation to the acquiring simulation (see 5.9.4). The use of this record is mandatory if the entity being transferred is a munition that was previously specified in the Munition Entity ID field of a Fire PDU. If there is no corresponding Fire PDU for a munition being transferred (e.g., a submunition), then use of the Launched Munition record is optional, but if it is used in this manner, the Fire Event ID field shall be set to all zeroes, and the Firing Entity ID field shall be set to NO_SPECIFIC_ENTITY. This record may also be used for other purposes.

When transferring a launched munition entity, this record shall provide the firing and targeting information for the entity being transferred. If there was a Fire PDU previously issued for the munition, the divesting simulation shall set the firing fields of this record in accordance with the (original) Fire PDU's Event ID field and Firing Entity ID field. The targeting information shall be based on the munition model's current targeting solution. The firing information provided by this record will allow the acquiring simulation to properly issue a subsequent Detonation PDU (with respect to the Detonation PDU's Source Entity ID field and Event ID field), or if the munition is transferred again, the (original) firing information shall be passed along to the next acquiring simulation.

Either the *Target Entity ID* or the *Target Location in World Coordinates* field may be specified, but not both.

Additional target entities and/or target locations may be communicated using the Association record (see 6.2.9). If one or more Association records are to be used in this fashion, they shall immediately follow the Launched Munition record.

The Launched Munition record shall contain the following fields:

a) *Fire Event ID.* This field shall contain the same data as in the Event ID field of the Fire PDU that communicated the launch of the munition. This field shall be represented by an Event Identifier record (see 6.2.33).

b) *Firing Entity ID.* This field shall contain the same data as in the Firing Entity ID field of the Fire PDU that communicated the launch of the munition. This field shall be represented by an Entity Identifier record (see 6.2.28).

c) *Target Entity ID.* This field shall identify the munition's target entity. The value of this field may be different than that contained in the Fire PDU that communicated the launch of the munition. If there is no identified target entity, this field shall contain the value TARGET_ID_UNKNOWN. This field shall be represented by an Entity Identifier record (see 6.2.28).

d) *Target Location in World Coordinates*. This field shall specify the munition's geographic target location in world coordinates. This field shall be set to 0.0, 0.0, 0.0 (all bits set to zero) if there is no specific geographic target. This field shall be represented by a World Coordinates record (see 6.2.98).

e) *Padding.* Three 16-bit padding fields are required as shown in order to maintain proper octet alignment when a World Coordinates record is included.

The format of the Launched Munition record shall be as shown in Table 81.

Table 81—Launched Munition record

Field size (bits)	Field name	Data type
48	Fire Event ID	Event Identifier record (see 6.2.33)
16	Padding	16 bits unused
48	Firing Entity ID	Entity Identifier record (see 6.2.28)
16	Padding	16 bits unused
48	Target Entity ID	Entity Identifier record (see 6.2.28)
16	Padding	16 bits unused
192	Target Location in World Coordinates	World Coordinates record (see 6.2.98)
Total Launched Munition record size = 384 bits		

6.2.51 Layer Header record

The identification of the additional information layer number, layer-specific information, and length of the layer shall be specified by a Layer Header record. The fields of this record are as follows:

a) *Layer Number*. This field shall identify the layer number. The field shall be represented as an 8-bit unsigned integer.

b) *Layer-Specific Information*. This field shall specify the layer-specific information that varies by System Type (see 6.2.87) and Layer Number. This field shall be represented by an 8-bit enumeration.

c) *Length*. This field shall specify the length in octets of the layer, including the Layer Header record and shall be represented by a 16-bit unsigned integer.

The format of the Layer Header record shall be as shown in Table 82.

Table 82—Layer Header record

Field size (bits)	Field name	Data type
8	Layer Number	8-bit unsigned integer
8	Layer-Specific Information	8-bit enumeration
16	Length	16-bit unsigned integer
Total Layer Header record size = 32 bits		

6.2.52 Linear Segment Parameter record

The specification of an individual segment of a linear segment synthetic environment object in a Linear Object State PDU shall be represented by a Linear Segment Parameter record. This record shall specify the number of the segment; any modification in the segment's location or orientation; the dynamic appearance attributes of the segment; the location and orientation of the segment; and the length, height, depth, and width of the segment. The Linear Segment Parameter record shall contain the following information:

a) *Segment Number.* This field shall identify the individual segment of the linear segment and shall be specified by an 8-bit unsigned integer.

b) *Segment Modification.* This field shall identify whether a modification has been made to the point object's location or orientation and shall be specified by an 8-bit enumeration (see [UID 241]).

c) *General Segment Appearance*. This field shall specify general dynamic appearance attributes of the segment. This record shall be defined as a 16-bit record (see [UID 229]).

d) *Specific Segment Appearance*. This field shall specify specific dynamic appearance attributes of the segment. This record shall be defined as a 32-bit record.

e) *Segment Location.* This field shall specify the location of the linear segment in the simulated world and shall be represented by a World Coordinates record (see 6.2.98).

f) *Segment Orientation.* This field shall specify the orientation of the linear segment about the segment location and shall be represented by a Euler Angles record (see 6.2.32).

g) *Segment Length.* The length of the linear segment, in meters, extending in the positive *x*-direction shall be specified by a 32-bit floating point.

h) *Segment Width.* The total width of the linear segment, in meters, shall be specified by a 32-bit floating point. One half of the width shall extend in the positive *y*-direction, and one half of the width shall extend in the negative *y*-direction.

i) *Segment Height.* The height of the linear segment, in meters, above ground shall be specified by a 32-bit floating point.

j) *Segment Depth.* The depth of the linear segment, in meters, below ground level shall be specified by a 32-bit floating point.

The format of the Linear Segment record shall be as shown in Table 83.

Table 83—Linear Segment Parameter record

Field size (bits)	Field name	Data type
8	Segment Number	8-bit unsigned integer
8	Segment Modification	8-bit enumeration
16	General Segment Appearance	16-bit record
32	Specific Segment Appearance	32-bit record
192	Segment Location	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
96	Segment Orientation	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
32	Segment Length	32-bit floating point
32	Segment Width	32-bit floating point

Table 83—Linear Segment Parameter record *(continued)*

Field size (bits)	Field name	Data type
32	Segment Height	32-bit floating point
32	Segment Depth	32-bit floating point
32	Padding	32 bits unused
Total Linear Segment Parameter record size = 512 bits		

6.2.53 Live Entity Identifier record

The unique designation of each entity in an event or exercise that is contained in a Live Entity PDU shall be specified by a Live Entity Identifier record. The Live Entity Identifier (Live Entity ID) record shall consist of a Live Simulation Address record (see 6.2.54) and an Entity Number.

No live entity shall be assigned an identifier equal to NO_LIVE_SITE: NO_LIVE_APPLIC: NO_ENTITY or ALL_LIVE_SITES: ALL_LIVE_APPLIC: ALL_ENTITIES. (See 6.1.8.)

The Live Entity Identifier is an object identifier. General object identifier requirements are specified in 4.2.5.2.

The format of the Live Entity Identifier record shall be as shown in Table 84.

Table 84—Live Entity Identifier record

Field size (bits)	Field name	Data type
16	Live Simulation Address	Live Site Number—8-bit unsigned integer
		Live Application Number—8-bit unsigned integer
16	Entity Number	16-bit unsigned integer
Total Live Entity Identifier record size = 32 bits		

6.2.54 Live Simulation Address record

6.2.54.1 General

A simulation's designation associated with all Live Entity IDs contained in Live Entity PDUs shall be specified by a Live Simulation Address record. A Live Simulation Address record shall consist of a Live Site Number and a Live Application Number. The format of the Live Simulation Address record shall be as shown in Table 85.

Table 85—Live Simulation Address record

Field size (bits)	Field name	Data type
8	Live Site Number	8-bit unsigned integer
8	Live Application Number	8-bit unsigned integer
Total Live Simulation Address record size = 16 bits		

6.2.54.2 Live site number

A live site is defined as a facility, an installation, an organizational unit, or a geographic location that has one or more applications capable of participating in a distributed event using Live Entity PDUs for live objects. A facility, an installation, an organizational unit, or a geographic location may have multiple sites associated with it. The Site Number is the first component of the Live Simulation Address, which defines a live simulation.

Note that if the site chooses to produce live objects as entities using the Entity State and other related PDUs, then the Simulation Address record specified in 6.2.80 shall be used.

The following requirements shall apply:

a) The range of valid Live Site Numbers that may be assigned to a specific live site is from 1 to 255.

b) A simulation or live range site shall be able to use any Live Site Number assignment (i.e., configuration) in the range specified in item a) above. In addition, it shall be able to receive and process any Live Site Number in the range of valid Live Site Numbers. There are no special Live Site Numbers used for simulation management.

The manner in which Live Site Numbers are assigned is outside the scope of this standard. There is no requirement that a live site's Live Site Number be changed for each event in which it is participating.

6.2.54.3 Live application number

An application associated with a live site is termed a live application. A Live Application Number is the second component of a Live Simulation Address. Each live application participating in an event (e.g., training exercise, experiment, and test) shall be assigned a unique Live Application Number that is different from any other simulation or Live Application Number that is part of the same event from the same site.

The following requirements shall apply:

a) The range of valid Live Application Numbers that may be assigned to a specific live application is from 1 to 255.

b) A live range application shall be able to use any Live Application Number assignment (i.e., configuration) in the range specified in item a) above. In addition, it shall be able to receive and process any Live Application Number in the range of valid Live Application Numbers. There are no special Live Application Numbers used for simulation management.

The manner in which Live Application Numbers are assigned is outside the scope of this standard. There is no requirement that a live application's Live Application Number be changed for each event in which it is participating.

6.2.55 Mine Entity Identifier record

The unique designation of a mine contained in the Minefield Data PDU shall be specified by the Mine Entity Identifier record. The Mine Entity Identifier (Mine Entity ID) represents a special kind of entity as no Entity State PDU is issued for it. Its location and characteristics are only defined in the Minefield Data PDU. The Mine Entity ID may be included in any PDU field that specifies a referenced Entity Identifier such as a Target Entity Identifier field. If a Minefield Data PDU is processed by a simulation for the purpose of determining the location and characteristics of mines in a minefield, it may create an active remote entity based on the mine entity data. In this case, the Mine Entity Identifier shall be the Entity Identifier. A mine entity that is contained in a Minefield Data PDU shall not be allowed to be an entity in the IsGroupOf PDU and vice versa.

The Mine Entity Identifier is an object identifier. General object identifier requirements are specified in 4.2.5.2.

Note that a mine may also be designated as a munition entity for which Entity State PDUs are being used. The same mine, as indicated by the Mine Entity ID, shall not be both a mine in a Minefield Data PDU and a mine for which an Entity State PDU is issued. A simulation may remove an entity from a mine field and issue it as a separate Entity State PDU. If this occurs, the mine entity shall be removed from the Minefield Data PDU.

The format of the Mine Entity Identifier record shall be as shown in Table 86.

Table 86—Mine Entity Identifier record

Field size (bits)	Field name	Data type
32	Simulation Address	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
16	Mine Entity Number	16-bit unsigned integer
Total Mine Entity Identifier record size = 48 bits		

6.2.56 Minefield Identifier record

The unique designation of a minefield shall be specified by the Minefield Identifier record. A minefield consists of individual mine entities. The Minefield Identifier (Minefield ID) does not represent an entity but rather an aggregation of mine entities for the purpose of conserving bandwidth. The Minefield Identifier shall not be included in any PDU field that specifies an Entity Identifier. If any of the Minefield PDUs are processed by a simulation, it shall not create an active entity based on the Minefield ID. A simulation shall only interact with the mine entities listed in the Minefield Data PDU. A mine entity that is contained in a Minefield Data PDU shall not be allowed to be an entity in the IsGroupOf PDU and vice versa.

The Minefield Identifier is an object identifier. General object identifier requirements are specified in 4.2.5.2.

The format of the Minefield Identifier record shall be as shown in Table 87.

Table 87—Minefield Identifier record

Field size (bits)	Field name	Data type
32	Simulation Address	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
16	Minefield Number	16-bit unsigned integer
Total Minefield Identifier record size = 48 bits		

6.2.57 Minefield Sensor Type record

Information about a particular mine sensor shall be represented using a Minefield Sensor Type record. The fields in this record are described as follows:

a) *Sensor Type Category*. This field shall specify the category of the sensor. This field shall be represented by a 4-bit enumeration (see [UID 193]).

b) *Sensor Type Subcategory*. This field shall specify the subcategory of the sensor. This field shall be represented by a 12-bit enumeration (see [UID 194-201]).

Table 88—Minefield Sensor Type record

Field name	Bits	Data type
Sensor Type Category	0–3	4-bit enumeration
Sensor Type Subcategory	4–15	12-bit enumeration
Total Minefield Sensor Type record size = 16 bits		

6.2.58 Modulation Parameters record

Modulation parameters associated with a specific radio system shall be identified by a Modulation Parameters (MP) record. MP records are defined in Annex C.

The fields of the MP record shall be as follows:

a) *Record-Specific fields*. These fields shall specify the modulation parameters associated with a specific radio system. All Record-Specific fields shall adhere to the alignment rules specified in 6.3.2. These fields will vary depending on the radio system.

The format of the Modulation Parameters record shall be as shown in Table 89.

Table 89—Modulation Parameters record

Field size (bits)	Field name	Data type
$8R_1$	Record-Specific field 1	R_1 octets
• • •		
$8R_N$	Record-Specific field N	R_N octets
$8P$	Padding	Padding to 64-bit boundary—P octets

Total Modulation Parameters record size = $\left(\sum_{i=1}^{N} 8R_i\right) + 8P$ bits

where

N is the number of Record-Specific fields
R_i is the length of Record-Specific field #i in octets
P is the number of padding octets, which is $\left\lceil\left(\sum_{i=1}^{N} R_i\right)/8\right\rceil 8 - \sum_{i=1}^{N} R_i$
$\lceil x \rceil$ is the largest integer $< x + 1$.

6.2.59 Modulation Type record

Information about the type of modulation used for radio transmission shall be represented by a Modulation Type record. This record uniquely identifies the various sets of signal parameters (i.e., the modulation type) that are used to determine whether two radios may interoperate. The modulation is characterized in a generic fashion by the Spread Spectrum, Major Modulation, and Detail fields. The classes of interoperable modulation types are enumerated by the Radio System field. This record shall specify the spread-spectrum usage, major modulation type, detailed information, and system compatibility. The fields of this record are as follows:

a) *Spread Spectrum.* This field shall indicate the spread spectrum technique or combination of spread spectrum techniques in use. The Spread Spectrum field shall consist of a 16-bit record. Each independent type of spread spectrum technique shall be represented by a single element of this array. If a particular spread spectrum technique is in use, the corresponding array element shall be set to one; otherwise it shall be set to zero. All unused array elements shall be set to zero. The supported spread spectrum techniques and their assignment to elements of the 16-element array are illustrated in Table 90.

Table 90—Spread spectrum field definition

Field Name	Bit	Value
Frequency Hopping	0	Enumeration
Pseudo Noise	1	Enumeration
Time Hopping	2	Enumeration
Padding	3–15	13 bits unused
Total Spread Spectrum field size = 16 bits		

b) *Major Modulation.* This field shall specify the major classification of the modulation type. This field shall be represented by a 16-bit enumeration (see [UID 155]).

c) *Detail.* This field shall provide certain detailed information depending on the major modulation type. This field shall be represented by a 16-bit enumeration (see [UID 156-162]).

d) *Radio System*. This field shall specify the radio system associated with this Transmitter PDU and shall be used as the basis to interpret other fields whose values depend on a specific radio system. This field shall be represented by a 16-bit enumeration (see [UID 163]).

The format of the Modulation Type record shall be as shown in Table 91.

Table 91—Modulation Type record

Field size (bits)	Field name	Data type
16	Spread Spectrum	16-bit record
16	Major Modulation	16-bit enumeration
16	Detail	16-bit enumeration
16	Radio System	16-bit enumeration
Total Modulation Type record size = 64 bits		

6.2.60 Munition record

An entity's munition (e.g., bomb and missile) information shall be represented by one or more Munition records. For each type or location of munition, this record shall specify the type, location, quantity, and status of munitions that an entity contains.

This record may be used (optionally) in a transfer transaction to send internal state data from the divesting simulation to the acquiring simulation (see 5.9.4). This record may also be used for other purposes.

The Munition record can be used to represent either non-location-specific or location-specific munition information, as follows:

— *Nonspecific Stations*. One Munition record is used for each type of munition when stations/launchers are not specified (i.e., the *Station/Launcher* field is zero).
— *Specific Stations*. One Munition record is used to indicate the quantity of a munition type located at one specific station/launcher when the *Station/Launcher* field value is nonzero.

All Munition records associated with an entity shall be either Nonspecific Stations or Specific Stations. Only one of the two types of stations shall be specified in a given transfer.

The Munition record shall contain the following fields:

a) *Munition Type*. This field shall identify the entity type of the munition. This field shall be represented by an Entity Type record (see 6.2.30).

b) *Station/Launcher*. This field shall indicate the station or launcher to which the munition is assigned and shall be represented by a 32-bit enumeration. A zero value shall indicate that this Munition record is not associated with any particular station/launcher. To specify a particular station/launcher, the enumerations represented by the Parameter Type field of either an attached or articulated part shall be used. See 6.2.94.2, 6.2.94.3, and Annex I.

c) *Quantity*. This field shall identify the quantity remaining of this munition and shall be represented by a 16-bit unsigned integer. When the Station/Launcher field is set to zero, the quantity represents the total quantity of this munition type for this entity. Otherwise, this field indicates the quantity of the munition type at the specified station/launcher.

d) *Munition Status.* This field shall specify the status of the munition. It shall be represented by an 8-bit enumeration (see [UID 327]).

The format of the Munition record shall be as shown in Table 92.

Table 92—Munition record

Field size (bits)	Field name	Data type
64	Munition Type	Entity Type record (see 6.2.30)
32	Station/Launcher	32-bit enumeration
16	Quantity	16-bit unsigned integer
8	Munition Status	8-bit unsigned integer
8	Padding	8 bits unused
Total Munition record size = 128 bits		

6.2.61 Munition Reload record

Munition (weapon) reload information for an entity shall be communicated using a Munition Reload record. The Munition Reload record shall indicate weapons (munitions) previously communicated via the Munition record.

This record shall be required to be sent if one or more Munition records are included in a transfer transaction. One Munition Reload record shall be sent for each Munition record that is sent and shall mirror the type of station representation (nonspecific or specific) found in that record. This record may also be used for other purposes. This record is optional for receipt. If processed, the data may be used or ignored as deemed appropriate by the simulation (e.g., an instant reload feature would override reload times, and an infinite quantity feature would override quantity values).

— *Nonspecific Stations*. One Munition Reload record is used for each type of munition when stations/launchers are not specified (i.e., the *Station/Launcher* field is zero).

— *Specific Stations*. One Munition Reload record is used to indicate the standard and maximum quantities and reload times of a munition type located at one specific station/launcher when the *Station/Launcher* field value is nonzero.

All Munition Reload records associated with an entity shall be either Nonspecific Stations or Specific Stations. Only one of the two types of stations shall be specified in a given transfer.

The reload times may vary for different stations/launchers that can accept a specific munition type, so only an overall reload time is given for loading the standard or maximum quantity at either a station/launcher or for the entire entity, whichever is specified. How quantities and reload times are calculated is beyond the scope of this standard.

The Munition Reload record shall contain the following fields:

a) *Munition Type*. This field shall identify the entity type of the munition. This field shall be represented by an Entity Type record (see 6.2.30).

b) *Station/Launcher*. This field shall indicate the station or launcher to which the munition is assigned and shall be represented by a 32-bit enumeration. A zero value shall indicate that this Munition Reload record is not associated with any particular station/launcher. To specify a particular station/launcher, the enumerations represented by the Parameter Type field of either an attached or an articulated part shall be used. See 6.2.94.2, 6.2.94.3, and Annex I.

c) *Standard Quantity*. This field shall identify the standard quantity of this munition type normally loaded at this station/launcher if a station/launcher is specified. If the Station/Launcher field is set to zero, then this is the total quantity of this munition type that would be present in a standard reload of all applicable stations/launchers associated with this entity. This field shall be represented by a 16-bit unsigned integer.

d) *Maximum Quantity*. This field shall specify the maximum quantity of this munition type that this station/launcher is capable of holding when a station/launcher is specified. This would be the value used when a maximum reload was desired to be set for this station/launcher. If the Station/launcher field is set to zero, then this is the maximum quantity of this munition type that would be present on this entity at all stations/launchers that can accept this munition type. This field shall be represented by a 16-bit unsigned integer.

e) *Standard Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the standard quantity of this munition type at this specific station/launcher. When the Station/Launcher field is set to zero, this shall be the number of seconds of simulation time required to perform a standard quantity reload of this munition type at all applicable stations/launchers for this entity. This field shall be represented by a 32-bit unsigned integer.

f) *Maximum Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the maximum possible quantity of this munition type at this station/launcher. When the Station/Launcher field is set to zero, this shall be the number of seconds of simulation time required to perform a maximum quantity load/reload of this munition type at all applicable stations/launchers for this entity. This field shall be represented by a 32-bit unsigned integer.

The format of the Munition Reload record shall be as shown in Table 93.

Table 93—Munition Reload record

Field size (bits)	Field name	Data type
64	Munition Type	Entity Type record (see 6.2.30)
32	Station/Launcher	32-bit enumeration
16	Standard Quantity	16-bit unsigned integer
16	Maximum Quantity	16-bit unsigned integer
32	Standard Quantity Reload Time	32-bit unsigned integer
32	Maximum Quantity Reload Time	32-bit unsigned integer
Total Munition Reload record size = 192 bits		

6.2.62 Named Location Identification record

Information about the discrete positional relationship of the part entity with respect to the its host entity shall be specified by a Named Location Identification record. This record shall contain the following fields:

a) *Station Name.* This field shall specify the station name within the host at which the part entity is located. This field shall be specified by a 16-bit enumeration (see [UID 212]).

b) *Station Number.* This field shall specify the number of the particular wing station, cargo hold, and so on, at which the part is attached. This field shall be represented by a 16-bit unsigned integer.

The format of the Named Location Identification record shall be as shown in Table 94.

Table 94—Named Location Identification record

Field size (bits)	Field name	Data type
16	Station Name	16-bit enumeration
16	Station Number	16-bit unsigned integer
Total Named Location Identification record size = 32 bits		

6.2.63 Object Identifier record

The unique designation of an environmental object shall be specified by the Object Identifier record. An object is not an entity although it has some of the characteristics of an entity. An entity may interact with an environmental object to a limited extent. An Environment Manager application manages objects and the interaction of objects with entities, including interactions with munitions and environmental processes. It then transmits updates of object states to simulations that can process objects. An Object Identifier is contained in the Areal, Linear, and Point Object State PDUs, and the Environmental Process PDU. It is also used in other PDUs (see 4.2.5.2).

NOTE—There is also an Environmental entity that can be used for lower fidelity representation of certain environmental conditions. The Environmental kind refers to entities in the usual physical-object sense, e.g., clouds and ice bergs, as well as inherent characteristics of a particular environment, such as sea state or transmissivity profile.

The format of the Object Identifier record shall be as shown in Table 95.

Table 95—Object Identifier record

Field size (bits)	Field name	Data type
32	Simulation Address	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
16	Object Number	16-bit unsigned integer
Total Object Identifier record size = 48 bits		

6.2.64 Object Type record

The type of synthetic environment point, linear object, and areal object in a DIS exercise shall be specified by an Object Type record. This record shall specify the domain of the object, the kind of object, and the specific identification of the entity. Fields not used shall contain the value zero. Tables of comprehensive object type enumerations can be found in SISO-REF-010 (see [UID 226] for Point Object State, [UID 227] for Linear Object State, and [UID 228] for Areal Object state). The Object Type record shall contain the following fields:

a) *Domain.* This field shall specify the domain in which the object exists. This field shall be represented by an 8-bit enumeration (see [UID 8]).

b) *Object Kind.* This field shall identify the kind of object described by the Object Type record. This field shall be represented by an 8-bit enumeration (see [UID 225]).

c) *Category.* This field shall specify the main category that describes the object. This field shall be represented by an 8-bit enumeration.

d) *Subcategory.* This field shall specify a particular subcategory to which an object belongs based on the Category field. This field shall be represented by an 8-bit enumeration.

The format of the Object Type record shall be as shown in Table 96.

Table 96—Object Type record

Field size (bits)	Field name	Data type
8	Domain	8-bit enumeration
8	Object Kind	8-bit enumeration
8	Category	8-bit enumeration
8	Subcategory	8-bit enumeration
Total Object Type record size = 32 bits		

6.2.65 Ownership Status record

The Ownership Status record is used to convey entity and conflict status information associated with transferring ownership of an entity.

The Ownership Status record shall contain the following fields:

a) *Entity ID*. This field shall identify the entity that is the subject of the Ownership Status record. This field shall be represented by an Entity Identifier record (see 6.2.28).

b) *Ownership Status*. The ownership and/or ownership conflict status of the entity represented by the Entity ID field. This field shall be represented by an 8-bit enumeration (see [UID 332]).

The format of the Ownership Status record shall be as shown in Table 97.

Table 97—Ownership Status record

Field size (bits)	Field name	Data type
48	Entity ID	Entity Identifier record (see 6.2.28)
8	Ownership Status	8-bit enumeration
8	Padding	8 bits unused
Total Ownership Status record size = 64 bits		

6.2.66 PDU Header record

A PDU Header record shall be the first part of each PDU excluding a Live Entity (LE) PDU, which has a separate LE PDU Header record (see 9.3.2). The fields of this record are as follows:

a) *Protocol Version.* This field shall specify the version of protocol used in this PDU and shall be specified by an 8-bit enumeration (see [UID 3]). See 4.2.10 for additional information and associated issuance and receipt rules.

b) *Exercise Identifier.* This field shall specify the exercise to which the PDU pertains and shall be represented by an Exercise Identifier (see 6.2.34).

c) *PDU Type.* This field shall indicate the type of PDU that follows. This field shall be represented by an 8-bit enumeration (see [UID 4]).

d) *Protocol Family.* This field shall indicate the family of protocols to which the PDU belongs. This field shall be represented by an 8-bit enumeration (see [UID 5]).

e) *Timestamp.* This field shall specify the time that the data in the PDU was generated and shall be represented by a timestamp (see 6.2.88).

f) *Length.* This field shall specify the length of the PDU, including the PDU Header, in octets and shall be represented by a 16-bit unsigned integer. The Length shall be calculated as shown in the Total PDU size formulas at the bottom of each PDU table in Clause 7 and Clause 9, with the resulting bit size divided by 8 to attain the number of octets in the PDU.

g) *PDU Status*. This field shall specify the PDU status related to one or more PDU Types (see 6.2.67).

The format of the PDU Header record shall be as shown in Table 98.

Table 98—PDU Header record

Field size (bits)	Field name	Data type
8	Protocol Version	8-bit enumeration
8	Exercise Identifier	8-bit unsigned integer
8	PDU Type	8-bit enumeration
8	Protocol Family	8-bit enumeration
32	Timestamp	32-bit unsigned integer
16	Length	16-bit unsigned integer
8	PDU Status	8-bit record
8	Padding	8 bits unused
Total PDU Header record size = 96 bits		

6.2.67 PDU Status record

The PDU Status record shall be used to indicate status information that either (1) affects the processing of a specific PDU as a whole without regard to any specific data field in the PDU, (2) provides information related to the interpretation of one or more data fields or their content, or (3) provides information that affects the processing of an entity, other object, or environmental process associated with a PDU. This record shall be defined as an 8-bit record. Fields not applicable for a specific PDU type shall be set to zero. The PDU Status record consists of the following fields:

a) *Transferred Entity Indicator (TEI)*. The Transferred Entity Indicator field shall indicate whether the simulation identified in the primary Entity ID field (Site, Application) is the same one that owns the entity—No Difference (0) or whether a different simulation owns the entity—Difference (1). See Transfer Ownership (5.9.4). The TEI field shall occupy bit 0 of the PDU Status record for any PDU type that implements this field. This field shall be represented by a 1-bit enumeration (see [UID 301]).

b) *LVC Indicator (LVC)*. The LVC Indicator field shall indicate whether the data contained in the PDU is related to a Live (1), Virtual (2), or Constructive (3) entity. If the LVC designation is not able to be determined, this field shall be set to No Statement (0). See D.2.2.3, D.2.2.4, and D.2.2.5. The LVC Indicator field shall occupy bits 1 and 2 of the PDU Status record for any PDU type that implements this field. This field shall be represented by a 2-bit enumeration (see [UID 302]).

c) *Coupled Extension Indicator (CEI)*. The Coupled Extension Indicator field shall be used to indicate whether the PDU is Coupled (1) or Not Coupled (0) with an Attribute PDU (see 5.3.6.3). The CEI field shall occupy bit 3 of the PDU Status record for any PDU type that implements this field. This field shall be represented by a 1-bit enumeration (see [UID 303]).

d) *Fire Type Indicator (FTI)*. The Fire Type Indicator field shall indicate whether the type of object fired was a Munition (0) or an Expendable (1). The FTI field shall occupy bit 4 of the PDU Status record for any PDU type that implements this field. This field shall be represented by a 1-bit enumeration (see [UID 304]).

e) *Detonation Type Indicator (DTI)*. The Detonation Type Indicator field shall indicate whether the type of object that detonated, exploded, or burst was a Munition (0), Expendable (1), or Non-

Munition Explosion (2). The DTI field shall occupy bits 4 and 5 of the PDU Status record for any PDU type that implements this field. This field shall be represented by a 2-bit enumeration (see [UID 305]).

f) *Radio Attached Indicator (RAI)*. The Radio Attached Indicator field shall be used to indicate whether the radio transmitter or receiver is attached to an entity or object as follows: No Statement (0), Unattached (1), and Attached (2). The RAI field shall occupy bit 4 and bit 5 of the PDU Status record for the Transmitter, Signal, and Receiver PDUs. This field shall be represented by a 2-bit enumeration (see [UID 306]).

g) *Intercom Attached Indicator (IAI)*. The Intercom Attached Indicator field shall be used to indicate whether the intercom is attached to an entity or object as follows: No Statement (0), Unattached (1), and Attached (2). The IAI field shall occupy bit 4 and bit 5 of the PDU Status record for the Intercom Signal and Intercom Control PDUs. This field shall be represented by a 2-bit enumeration (see [UID 307]).

h) *IFF Simulation Mode (ISM)*. The IFF Simulation Mode field shall be set to Regeneration (0) when the Simulation Mode field of the Change/Options record for this IFF PDU is set to Regeneration (0). It shall be set to Interactive (1) when the Simulation Mode field of the Change/Options record for this IFF PDU is set to Interactive (1). This field shall occupy bit 4 of the PDU Status record. This field shall be represented by a 1-bit enumeration (see [UID 308]).

i) *Active Interrogation Indicator (AII)*. The Active Interrogation Indicator field shall indicate whether this IFF PDU represents an active interrogation when an interrogator is operating in the Regeneration Mode (see B.5.1.2.10). It shall be set to Not Active (0) or Active (1). This field shall occupy bit 5 of the PDU Status record. This field shall be represented by a 1-bit enumeration (see [UID 389]).

Table 99 indicates the PDU statuses associated with each PDU Type and their designated field position. Specific PDU Types are listed separately when there is more than one applicable PDU Status field for that PDU.

Table 99—PDU Status record

PDU Type	PDU Name	Field Bit Position							
		7	**6**	**5**	**4**	**3**	**2**	**1**	**0**
1	Entity State	—	—	—	—	CEI	LVC		TEI
2	Fire	—	—	—	FTI	CEI	LVC		—
3	Detonation	—	—	DTI		CEI	LVC		—
4 to 22	*All PDUs*	—	—	—	—	CEI	LVC		—
23	Electromagnetic Emission	—	—	—	—	CEI	LVC		TEI
24	Designator	—	—	—	—	CEI	LVC		TEI
25	Transmitter	—	—	RAI		CEI	LVC		TEI
26	Signal	—	—	RAI		CEI	LVC		TEI
27	Receiver	—	—	RAI		CEI	LVC		TEI
28	IFF	—	—	AII	ISM	CEI	LVC		TEI
29 to 30	*All PDUs*	—	—	—	—	CEI	LVC		—
31	Intercom Signal	—	—	IAI		CEI	LVC		TEI

Table 99—PDU Status record *(continued)*

PDU Type	PDU Name	Field Bit Position							
		7	6	5	4	3	2	1	0
32	Intercom Control	—	—	IAI		CEI	LVC		TEI
33 to 40	*All PDUs*	—	—	—	—	CEI	LVC		—
41	Environmental Process	—	—	—	—	CEI	LVC		TEI
42 to 66	*All PDUs*	—	—	—	—	CEI	LVC		—
67	Entity State Update	—	—	—	—	CEI	LVC		TEI
68	Directed Energy Fire	—	—	—	—	CEI	LVC		—
69	Entity Damage Status	—	—	—	—	CEI	LVC		—
70	IO Action	—	—	—	—	CEI	LVC		—
71	IO Report	—	—	—	—	CEI	LVC		—
72	Attribute	—	—	—	—	—	LVC		—
73 to 255	None assigned	—							

Legend

AII Active Interrogation Indicator
CEI Coupled Extension Indicator
TEI Transferred Entity Indicator
LVC LVC Indicator
FTI Fire Type Indicator
DTI Detonation Type Indicator
RAI Radio Attached Indicator
IAI Intercom Attached Indicator
ISM IFF Simulation Mode

6.2.68 Propulsion System Data record

The SEES Propulsion System Data record contains information describing the propulsion systems of the entity. This information shall be provided for each active propulsion system defined. The fields for this record are defined as follows:

a) *Power Setting.* This field shall specify the power setting of the current propulsion system being described. This value shall represent the power setting after any response lags have been incorporated. This field shall be represented by a 32-bit floating point number. The Power Setting is a normalized value indicating the operator's power controller setting (e.g., the throttle position).

   In normal operation, the value shall be set in the range of POWER_IDLE to POWER_MILITARY (maximum without afterburner) inclusive, proportional to the controller position.

   If an afterburner is engaged, the setting shall be in the range of POWER_MIN_AFTERBURNER to POWER_MAX_AFTERBURNER inclusive.

   If the propulsion system is off, the setting shall be set to POWER_ENGINE_OFF.

   See Table 25 for values assigned to these symbolic names.

b) *Engine RPM.* This field shall specify the current engine speed in RPM. This field shall be represented by a 32-bit floating point number.

The format of the Propulsion System Data record shall be as shown in Table 100.

Table 100—Propulsion System Data record

Field size (bits)	Field name	Data type
32	Power Setting	32-bit floating point
32	Engine RPM	32-bit floating point
Total Propulsion System Data record size = 64 bits		

6.2.69 Protocol Mode

The Protocol Mode record shall be used to identify which protocol mode is being used to communicate minefield data. Usage of the Protocol Mode record in the Minefield State PDU is described in 5.10 and 7.9.2. The fields are described below, and the format of the Protocol Mode record shall be as shown in Table 101:

a) *Protocol Mode*. This 2-bit enumeration shall identify which protocol mode is being used to communicate minefield data (see [UID 336]).

Table 101—Protocol Mode record

Field name	Bits	Data type
Padding	0–13	14 bits unused
Protocol Mode	14–15	2-bit enumeration
Total Protocol Mode record size = 16 bits		

6.2.70 Radio Identifier record

The unique designation of an attached or unattached radio in an event or exercise shall be specified by the Radio Identifier record. The Radio Identifier record shall consist of the Radio Reference ID field and the Radio Number field. For an intercom simulation using the Transmitter and Signal PDUs to model a Simple Intercom (see C.3), the Radio Identifier refers to the identifier of the intercom communications device. The format of the Radio Identifier record shall be as shown in Table 102.

Table 102—Radio Identifier record

Field size (bits)	Field name	Data type
48	Radio Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Radio Number	16-bit unsigned integer
Total Radio Identifier record size = 64 bits		

6.2.71 Radio Type record

The type of radio in a DIS exercise shall be specified by a Radio Type record. The Radio Type record is identical in field format to the Entity Type record. In addition, the Radio Type record has definitions for the Category, Subcategory, Specific, and Extra fields that are unique to radios. The Radio Type record is included in the Transmitter PDU to identify the specific radio transmitter equipment. A radio is typically associated with a platform entity. However, it may be associated with an object or be a separate radio entity. When the radio is a separate entity, the same set of enumerated values shall be used in the Radio Type record of its Transmitter PDU and the Entity Type record of its Entity State PDU. The enumerations for the Radio Type record are defined in [UID 22] for Category and [UID 23] for Subcategory/Radio Nomenclature System. The comprehensive list of specific radio models is contained in the subset of [UID 30] with Entity Kind set to Radio (7). The fields of this record shall be as follows:

a) *Entity Kind.* This field shall be represented by an 8-bit enumeration and shall be set to the Radio kind (7).

b) *Domain.* The Domain field shall be set to the domain that the radio normally operates in. If it normally operates in multiple domains, either one of the domains it operates in may be specified or a domain of Other may be used. The same domain values as apply to platforms shall be used. The Domain field shall be represented by an 8-bit enumeration.

c) *Country.* This field shall specify the country to which the design of the radio is attributed and shall be represented by a 16-bit enumeration.

d) *Category*. This field shall specify the primary type of information that is transmitted or received by the radio when in operation (e.g., voice transmission/reception and data link transmission/reception). It shall be represented by an 8-bit enumeration.

e) *Subcategory*. This field shall specify the Radio Nomenclature System associated with the radio identified in the Specific field (e.g., a JETDS PRC-series radio). This field shall be represented by an 8-bit enumeration.

f) *Specific.* This field shall identify a specific radio associated with the Radio Nomenclature System contained in the Subcategory field. (e.g., PRC-117). This field shall be represented by an 8-bit enumeration.

g) *Extra*. This field shall specify the version or variation of the radio. This field shall be represented by an 8-bit enumeration.

The format of the Radio Type record shall be as shown in Table 103.

Table 103—Radio Type record

Field size (bits)	Field name	Data type
8	Entity Kind	8-bit enumeration
8	Domain	8-bit enumeration
16	Country	16-bit enumeration
8	Category	8-bit enumeration
8	Subcategory	8-bit enumeration
8	Specific	8-bit enumeration
8	Extra	8-bit enumeration
Total Radio Type record size = 64 bits		

Example:

The enumeration for AN/PRC-117 is 7.0.225.1.43.117.0, which is identified as follows:

Kind = Radio (7)
Domain = Other (0)
Country = United States (225)
Category = Voice Transmission/Reception (1)
Subcategory = PRC Set 1 (43)
Specific = PRC-117 (117)
Extra = 0 (No version or variation)

6.2.72 Record Query Specification record

The identification of the records being queried shall be specified using the Record Query Specification record. This record shall specify the number and identification of records requested. The fields of this record are as follows:

a) *Number of Records*. This field shall specify the number of Record ID fields. This field shall be represented by a 32-bit unsigned integer.
b) *Record ID*. This field shall specify the identification of the records being queried and shall be represented by a 32-bit enumeration (see [UID 66]).

The format of the Record Query Specification record shall be as shown in Table 104.

Table 104—Record Query Specification record

Field size (bits)	Field name	Data type
32	Number of Records (N)	32-bit unsigned integer
32	Record ID #1	32-bit enumeration
• • •		
32	Record ID #N	32-bit enumeration
Total Record Query Specification record size = 32 + 32*N* bits where *N* is the number of records		

6.2.73 Record Specification record

Record information shall be represented using the Record Specification record. This record shall specify the number of record sets contained in the Record Specification record and the record details. The fields of this record are as follows:

a) *Number of Record Sets.* This field shall specify the number of record sets and shall be represented by a 32-bit unsigned integer.

b) *Record ID.* This field shall specify the data structure used to convey the parameter values of the record for each record. The field shall be represented by a 32-bit enumeration (see [UID 66]).

c) *Record Set Serial Number.* This field shall specify the serial number of the first record in the block of records and shall be represented by a 32-bit unsigned integer.

d) *Record Length.* This field shall specify the length, in bits, of the record and shall be represented by a 16-bit unsigned integer.

e) *Record Count.* This field shall specify the number of records included in the record set and shall be represented by a 16-bit unsigned integer.

f) *Record Values.* This field contains the concatenated records of the format specified by the Record ID field. The length of this field is the Record Length multiplied by the Record Count, in units of bits.

g) *Padding.* Padding of 0 to 63 unused bits as required for 64-bit alignment of the Record Set field.

The format of the Record Specification record shall be as shown in Table 105.

Table 105—Record Specification record

Field size (bits)	Field name	Data type
32	Number of Record Sets (R)	32-bit unsigned integer
$128 + L_1Q_1 + P_1$	Record Set #1	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_1)
		Record Count—16-bit unsigned integer (Q_1)
		Record Values—(L_1Q_1) bits
		Padding to 64-bit boundary—P_1 bits
	⋮	
$128 + L_RQ_R + P_R$	Record Set #R	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_R)
		Record Count—16-bit unsigned integer (Q_R)
		Record Values—(L_RQ_R) bits
		Padding to 64-bit boundary—P_R bits

Total Record Specification record size = $32 + \sum_{i=1}^{R} (128 + L_iQ_i + P_i)$ bits

where

R is the number of Record Sets
L_i is the record length in Record Set i
Q_i is the record count in Record Set i
P_i is the number of padding bits for Record Set i, which is $\lceil (L_iQ_i)/64 \rceil 64 - L_iQ_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

6.2.74 Relationship record

The relationship of the part entity to its host entity shall be specified by a Relationship record. The fields of this record shall be as follows:

a) *Nature.* This field shall specify the nature or purpose for joining of the part entity to the host entity and shall be represented by a 16-bit enumeration (see [UID 210]).

b) *Position.* This field shall specify the position of the part entity with respect to the host entity and shall be represented by a 16-bit enumeration (see [UID 211]).

The format of the Relationship record shall be as shown in Table 106.

Table 106—Relationship record

Field size (bits)	Field name	Data type
16	Nature	16-bit enumeration
16	Position	16-bit enumeration
Total Relationship record size = 32 bits		

6.2.75 Request ID

The Request ID is a 32-bit monotonically increasing integer identifier inserted by the SM into all Simulation Management PDUs. This is a unique identifier that shall be used to identify the latest in a series of competing requests and identifying acknowledgments.

6.2.76 Secondary Operational Data record

Additional operational data for an IFF emitting system and the number of IFF Fundamental Parameter Data records shall be communicated by a Secondary Operational Data record. The fields of this record are as follows:

a) *Operational Parameters 1 and 2.* These fields shall specify additional operational characteristics of the IFF emitting system. Each 8-bit field will vary depending on the system type.

b) *Number of IFF Fundamental Parameter Data Records*. This field shall specify the number of IFF Fundamental Parameter Data records that follow. This field shall be represented by a 16-bit unsigned integer.

The format of the Secondary Operational Data record shall be as shown in Table 107.

Table 107—Secondary Operational Data record

Field size (bits)	Field name	Data type
8	Operational Parameter 1	8 bits defined by system type
8	Operational Parameter 2	8 bits defined by system type
16	Number of IFF Fundamental Parameter Data Records	16-bit unsigned integer
Total Secondary Operational Data record size = 32 bits		

6.2.77 Sensor record

An entity's sensor information shall be represented by one or more Sensor records. For each type or location of sensor, this record shall specify the type, location, quantity, and status of sensors that an entity contains.

The Sensor record may be used to represent passive sensors that are not otherwise represented in the simulation.

This record may be used (optionally) in a transfer transaction to send internal state data from the divesting simulation to the acquiring simulation (see 5.9.4). This record may also be used for other purposes.

The Sensor record can be used to represent either non-location-specific or location-specific sensor information, as follows:

— *Nonspecific Stations*. One Sensor record is used for each type of sensor when stations are not specified (i.e., the *Station* field is zero).

— *Specific Stations*. One Sensor record is used to indicate the quantity of a sensor type located at one specific station when the *Station* field value is nonzero.

All Sensor records associated with an entity shall be either Nonspecific Stations or Specific Stations. There shall be no mixing of the two types.

The Sensor record shall contain the following fields:

a) *Sensor Type Source*. This field shall specify the source of the Sensor Type field. It shall be represented by an 8-bit enumeration (see [UID 414]).

b) *Sensor On/Off Status*. This field shall specify the on/off status of the sensor. It shall be represented by an 8-bit enumeration (see [UID 331]).

c) *Sensor Type*. This field shall specify the sensor type and shall be represented by a 16-bit enumeration. Values for this field depend on the value of the Sensor Type Source field. See Table 109 for a reference to the section of SISO-REF-010 containing the Sensor Type enumeration to be used for each Sensor Type Source.

d) *Station*. This field shall indicate the station to which the sensor is assigned and shall be represented by a 32-bit enumeration. A zero value shall indicate that this Sensor record is not associated with any particular station. To specify a particular station, the enumerations represented by the Parameter Type field of either an attached or an articulated part shall be used. See 6.2.94.2, 6.2.94.3, and Annex I.

e) *Quantity*. This field shall identify the quantity of this sensor and shall be represented by a 16-bit unsigned integer. When the Station field is set to zero, the quantity represents the total quantity of this sensor type for this entity. Otherwise, this field indicates the quantity of the sensor type at the specified station.

The format of the Sensor record shall be as shown in Table 108.

Table 108—Sensor record

Field size (bits)	Field name	Data type
8	Sensor Type Source	8-bit enumeration
8	Sensor On/Off Status	8-bit enumeration
16	Sensor Type	16-bit enumeration
32	Station	32-bit enumeration

Table 108—Sensor record *(continued)*

Field size (bits)	Field name	Data type
16	Quantity	16-bit unsigned integer
16	Padding	16 bits unused
Total Sensor record size = 96 bits		

Table 109—Sensor Type sources

Sensor Type source	Category	Associated PDU	PDU Field name	Sensor Type (SISO-REF-010 UID Reference)
0	Other Active Sensors	None		[UID 325]
1	Electromagnetic	Electromagnetic Emission	Emitter Name	[UID 75]
2	Passive Sensors	None		[UID 326]
3	Minefield Sensors	Minefield Query Minefield Data	Sensor Type	See 6.2.57
4	Underwater Acoustics	Underwater Acoustic	Acoustic System Name	[UID 144]
5	Lasers	Designator	Code Name	[UID 80]

6.2.78 Service type

Service types shall be specified by an 8-bit enumeration (see [UID 63]).

6.2.79 Silent Entity System record

This record contains information about an entity not producing Entity State PDUs. It shall contain the following fields:

a) *Number of Entities*. This field shall specify the number of entities that have the type specified by the Entity Type field and shall be represented by a 16-bit unsigned integer.

b) *Number of Appearance Records*. This field shall specify the number of Entity Appearance records that follow (see 5.9.2.2.3). This number shall be represented by a 16-bit unsigned integer.

c) *Entity Type*. This field shall specify the entity type common to the entities in this system list and shall be represented by an Entity Type record (see 6.2.30).

d) *Entity Appearances*. These fields shall specify the entity appearances of entities in the aggregate that deviate from the default. These fields shall be represented by Entity Appearance records (see 6.2.26).

Table 110—Silent Entity System record

Field size (bits)	Field name	Data type
16	Number of Entities	16-bit unsigned integer
16	Number of Appearance Records (*A*)	16-bit unsigned integer
64	Entity Type	Entity Type record
32	Entity Appearance #1	Entity Appearance record
• • •		
32	Entity Appearance #*A*	Entity Appearance record
Total Silent Entity System record size = 96 + 32*A* bits where *A* is the number of Entity Appearance records		

6.2.80 Simulation Address record

6.2.80.1 General

A simulation's designation associated with all object identifiers except those contained in Live Entity PDUs shall be specified by a Simulation Address record. A Simulation Address record shall consist of the Site Number and the Application Number.

The format of the Simulation Address record shall be as shown in Table 111.

Table 111—Simulation Address record

Field size (bits)	Field name	Data type
16	Site Number	16-bit unsigned integer
16	Application Number	16-bit unsigned integer
Total Simulation Address record size = 32 bits		

6.2.80.2 Site number

A site is defined as a facility, an installation, an organizational unit, or a geographic location that has one or more simulation applications capable of participating in a distributed event. A facility, an installation, an organizational unit, or a geographic location may have multiple sites associated with it. Each site participating in an event (e.g., training exercise, experiment, and test) shall be assigned a unique Site Number that is different from any other site that is part of the same event.

The Site Number is the first component of the simulation address, which defines a simulation. A simulation site is a site that generates simulated objects based on simulated data. A live site is a site that is associated with producing live objects from live sources such as producing entities representing live aircraft flying in a live training range. A simulation associated with a live site may issue Live Entity PDUs (e.g., the TSPI PDU) for the live objects or may issue Entity State PDUs for them.

See 6.1.8 for the values associated with symbolic names contained herein.

The following requirements shall apply:

a) The range of valid Site Numbers that may be assigned to a specific site is from 1 to 65 534.
b) A simulation site (i.e., the simulation applications at a site) shall be able to use any Site Number assignment (i.e., configuration) in the range specified in item a) above. In addition, it shall be able to receive and process any Site Number in the range of valid Site Numbers, as well as the special Site Numbers specified in item c) and item d) below.
c) A Site Number of NO_SITE shall indicate no specific site.
d) A Site Number of ALL_SITES shall indicate all sites.

The manner in which Site Numbers are assigned is outside the scope of this standard. There is no requirement that a site's Site Number be changed for each event in which it is participating.

6.2.80.3 Application number

An application is defined as a software program that is used to generate and process distributed simulation data including live, virtual, and constructive data. An application is the second component of simulation address. The application will issue and/or receive PDUs in the course of an event. Each application participating in an event (e.g., training exercise) shall be assigned a unique Application Number for the site with which the application is associated. See also Serial Simulations (4.2.3.3).

The following requirements shall apply:

a) The range of valid Application Numbers that may be assigned to a specific application is from 1 to 65 534.
b) A simulation application shall be able to use any Application Number assignment (i.e., configuration) in the range specified in item a) above. In addition, it shall be able to receive and process any Application Number in the range of valid Application Numbers, as well as the special Application Numbers specified in item c) and item d) below.
c) An Application Number of NO_APPLIC shall indicate no specific application.
d) An Application Number of ALL_APPLIC shall indicate all applications.

The manner in which Application Numbers are assigned is outside the scope of this standard. There is no requirement that an application's Application Number be changed for each event in which it is participating.

6.2.81 Simulation Identifier record

The unique designation of a simulation when using the 48-bit identifier format shall be specified by the Simulation Identifier record. The reason that the 48-bit format is required in addition to the 32-bit simulation address format that actually identifies a specific simulation is because some 48-bit identifier fields in PDUs may contain either an Object Identifier, such as an Entity ID, or a Simulation Identifier.

The Simulation Identifier shall consist of a Simulation Address record (see 6.2.80), and the Reference Number shall be set to zero. If a simulation is represented as an entity in the exercise, the entity shall be uniquely identified using the Entity Identifier record.

The Simulation Identifier record shall contain the following fields:

a) *Simulation Address*. This field shall be represented by a Simulation Address record (see 6.2.80).
b) *Reference Number*. This field shall be set to zero as there is no reference number associated with a Simulation Identifier. This shall be represented by a 16-bit unsigned integer.

The format of the Simulation Identifier record shall be as shown in Table 112.

Table 112—Simulation Identifier record

Field size (bits)	Field name	Data type
32	Simulation Address	Site number—16-bit unsigned integer
		Application number—16-bit unsigned integer
16	Reference Number	Not Applicable. This field shall be set to zero —16-bit unsigned integer
Total Simulation Identifier record size = 48 bits		

6.2.82 Simulation Management PDU Header record

A Simulation Management PDU Header record shall be the first part of each Simulation Management (SIMAN) and Simulation Management with Reliability (SIMAN-R) PDU. The PDU Header described in 6.2.66 is part of this Header record. The Simulation Management PDU Header record contains information required for all Simulation Management PDUs. The fields of this record are as follows:

a) *PDU Header*. This field contains data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record described in 6.2.66.
b) *Originating ID*. This field identifies the simulation or entity that is issuing the SIMAN or SIMAN-R PDU. This field shall be represented by either a Simulation Identifier record (see 6.2.81) or an Entity Identifier record (see 6.2.28), whichever is applicable (see 5.6.2.3).
c) *Receiving ID*. This field identifies the intended recipient of the SIMAN or SIMAN-R PDU (see 5.6.2.4).

The format of the Simulation Management PDU Header record shall be as shown in Table 113.

Table 113—Simulation Management PDU Header record

Field size (bits)	Field name	Data type
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration
		Protocol Family—8-bit enumeration
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit enumeration
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
Total Simulation Management PDU Header record size = 192 bits		

6.2.83 Standard Variable Specification record

The Standard Variable Specification record shall be used when a PDU requires a variable record section. This record section shall provide the issuing simulation the ability to include any number of Standard Variable records (also called SV records) that are either PDU-specific or are applicable to the processing of the PDU per exercise agreement. All PDUs that contain a Standard Variable Specification record section shall use the format specified herein as the last record section of the PDU. The names of the Standard Variable records for a specific PDU are customized to identify the PDU with which they are associated. For example, the SV records designed for use in an Entity Damage Status PDU are called “Damage Description records.”

The issuance and receipt rules for a Standard Variable record are included in either the 6.2 paragraph for that record or the Issuance section of the associated PDU, whichever is deemed appropriate. An SV record may be applicable to more than one PDU Type, and the issuance and receipt rules may be different for different PDU Types. In addition, any number of SV records, with either the same or different Record Types, may be included in a Standard Variable Specification record section up to the maximum size of the PDU. Record Types that are not recognized by the receiving simulation shall be ignored.

NOTE 1—The inclusion of an SV record in a PDU other than those specifically identified as being associated with that PDU can occur for several reasons. This includes a new prototype record designed specifically for a PDU or the need to include a record in a PDU to meet a simulation’s requirements or to support an exercise agreement.

The Record Type enumerated value assigned to an SV record shall be unique from all other SV records, including those that are applicable to other PDU types. In addition, it shall be unique from all Datum IDs associated with Fixed or Variable Datum records contained in a Datum Specification record. A Fixed or Variable Datum record may be transmitted as an SV record so long as the Record Type is set equal to the Datum ID and the Datum Value field formats remain identical when inserted into the SV Record-Specific field(s), excluding any additional padding required to meet SV record alignment requirements.

The first SV record of a Standard Variable Specification record shall start on a 64-bit boundary. Therefore, the Number of Standard Variable Records field starts 16 bits before a 64-bit boundary.

The Standard Variable Specification record shall contain the following fields:

a) *Number of Standard Variable Records*. This field shall specify the number of Standard Variable records and shall be represented by a 16-bit unsigned integer. If no records are present for an issuance of the PDU, this field shall be set to zero and no records shall be included.

b) *Standard Variable records*. Each Standard Variable record shall contain the following fields:

   1) Record Type. This field shall indicate the unique record number assigned to this Standard Variable record. It shall be represented by a 32-bit enumeration (see [UID 66]).

   2) Record Length. This field shall indicate the record length expressed as the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 64-bit boundary. All Record Length values shall be a multiple of 8. The Record Length shall be represented by a 16-bit unsigned integer.

   3) Record-Specific fields. These are the data fields of the record. Any number and types of data fields may be included.

   4) Padding. Padding shall be explicitly included in each record as necessary to make the record length a multiple of 8 octets (64 bits) so that the following record is automatically aligned. The record length requirement may be achieved by placing padding fields anywhere in the Standard Variable record as deemed appropriate, not necessarily at the end of the record.

NOTE 2—PDUs that use the Standard Variable Specification record format do not include padding between Standard Variable records since those records are always a multiple of 8 octets in length.

The format of the Standard Variable Specification record is shown in Table 114.

Table 114—Standard Variable Specification record

<table>
<tr><th>Field size (bits)</th><th>Field name</th><th>Data type</th></tr>
<tr><td>16</td><td>Number of Standard Variable Records (N)</td><td>16-bit unsigned integer</td></tr>
<tr><td rowspan="4">$48 + 8K_1 + 8P_1$</td><td rowspan="4">Standard Variable record #1</td><td>Record Type—32-bit enumeration</td></tr>
<tr><td>Record Length—16-bit unsigned integer ($6 + K_1 + P_1$)</td></tr>
<tr><td>Record-Specific fields—K_1 octets</td></tr>
<tr><td>Padding to 64-bit boundary—P_1 octets</td></tr>
<tr><td colspan="3">•
•
•</td></tr>
</table>

Table 114—Standard Variable Specification record *(continued)*

<table>
<tr><th>Field size (bits)</th><th>Field name</th><th>Data type</th></tr>
<tr><td rowspan="4">$48 + 8K_N + 8P_N$</td><td rowspan="4">Standard Variable record #N</td><td>Record Type—32-bit enumeration</td></tr>
<tr><td>Record Length—16-bit unsigned integer
$(6 + K_N + P_N)$</td></tr>
<tr><td>Record-Specific fields—K_N octets</td></tr>
<tr><td>Padding to 64-bit boundary—P_N octets</td></tr>
<tr><td colspan="3">Standard Variable Specification record size = $16 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

N is the number of Standard Variable records
K_i is the length of the Record-Specific field in Standard Variable record #i in octets
P_i is the number of padding octets in Standard Variable record #i, which is $\lceil (6 + K_i)/8 \rceil 8 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.</td></tr>
</table>

6.2.84 Storage Fuel record

An entity's storage fuel information shall be represented by one or more Storage Fuel records. For each type or location of storage fuel, this record shall specify the type, location, and quantity of storage fuel that an entity contains.

This record may be used (optionally) in a transfer transaction to send internal state data from the divesting simulation to the acquiring simulation (see 5.9.4). This record may also be used for other purposes.

The Storage Fuel record can be used to represent either non-location-specific or location-specific storage fuel information, as follows:

— *Nonspecific Fuel Location*. One Storage Fuel record is used for all storage fuel associated with the entity if the *Fuel Location* field is zero.

— *Specific Fuel Location*. One Storage Fuel record may be used to indicate the quantity of fuel remaining for each fuel location specified in the *Fuel Location* field associated with the entity.

The Storage Fuel records associated with an entity shall be either Nonspecific Fuel Location (one record) or Specific Fuel Location (one or more records). There shall be no mixing of specific and nonspecific fuel locations in a given transfer.

The Storage Fuel record shall contain the following fields:

a) *Fuel Quantity.* This field shall indicate the quantity of fuel remaining, in increments, specified by the Fuel Measurement Units field. When the Fuel Location field is set to zero, the fuel quantity applies to all storage locations associated with the entity. This field shall be represented by a 32-bit unsigned integer.

b) *Fuel Measurement Units*. This field shall identify the fuel measurement units. It shall be represented by an 8-bit enumeration (see [UID 328]).

c) *Fuel Type*. This field shall identify the type of fuel. It shall be represented by an 8-bit enumeration (see [UID 413]).

d) *Fuel Location.* This field shall indicate the location of the fuel as related to the entity. A zero indicates no specific location. It shall be represented by an 8-bit enumeration (see [UID 329]).

The format of the Storage Fuel record shall be as shown in Table 115.

Table 115—Storage Fuel record

Field size (bits)	Field name	Data type
32	Fuel Quantity	32-bit unsigned integer
8	Fuel Measurement Units	8-bit enumeration
8	Fuel Type	8-bit enumeration
8	Fuel Location	8-bit enumeration
8	Padding	8 bits unused
Total Storage Fuel record size = 64 bits		

6.2.85 Storage Fuel Reload record

An entity's storage fuel reload information shall be represented by one or more Storage Fuel Reload records. For each type or location of storage fuel, this record shall specify the type, location, fuel measurement units, reload quantity, and maximum quantity for storage fuel for either the whole entity or a specific storage fuel location (tank).

This record shall be required to be sent if one or more Storage Fuel records are included in a transfer transaction. One Storage Fuel Reload record shall be sent for each Storage Fuel record that is sent and shall mirror the type of station representation (nonspecific or specific) found in that record. This record may also be used for other purposes. This record is optional for receipt. If processed, the data may be used or ignored as deemed appropriate by the simulation (e.g., an instant reload feature would override reload times, and an infinite quantity feature would override quantity values).

The Storage Fuel Reload record can be used to represent either non-location-specific or location-specific storage fuel information, as follows:

— *Nonspecific Fuel Location*. One Storage Fuel Reload record is used for all storage fuel associated with the entity if the Fuel Location field is zero.

— *Specific Fuel Location*. One Storage Fuel Reload record may be used to indicate the standard and maximum quantities and reload times of fuel remaining for each fuel location (tank) specified in the Fuel Location field associated with the entity.

The Storage Fuel Reload records associated with an entity shall be either Nonspecific Fuel Location (one record) or Specific Fuel Location (one or more records). There shall be no mixing of specific and nonspecific fuel locations in a given transfer.

The reload times may vary for different fuel locations that can accept a specific storage fuel type, so only an overall reload time is given for loading the standard or maximum quantity at either a fuel location or for the entire entity, whichever is specified. How quantities and reload times are calculated is beyond the scope of this standard.

The Storage Fuel Reload record shall contain the following fields:

a) *Standard Quantity*. This field shall identify the standard quantity of this fuel type normally loaded at this fuel location if a fuel location is specified. If the Fuel Location field is set to zero, then this is the total quantity of this fuel type that would be present in a standard reload of all applicable fuel locations associated with this entity. This field shall be represented by a 32-bit unsigned integer.

b) *Maximum Quantity*. This field shall specify the maximum quantity of this fuel type that this fuel location is capable of holding when a fuel location is specified. This would be the value used when a maximum reload was desired to be set for this fuel location. If the Fuel Location field is set to zero, then this is the maximum quantity of this fuel type that would be present on this entity at all fuel locations that can accept this fuel type. This field shall be represented by a 32-bit unsigned integer.

c) *Standard Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the standard quantity of this fuel type at this specific fuel location. When the Fuel Location field is set to zero, this shall be the number of seconds of simulation time required to perform a standard quantity reload of this fuel type at all applicable fuel locations for this entity. This field shall be represented by a 32-bit unsigned integer.

d) *Maximum Quantity Reload Time*. This field shall specify the number of seconds of simulation time normally required to reload the maximum possible quantity of this fuel type at this fuel location. When the Fuel Location field is set to zero, this shall be the number of seconds of simulation time required to perform a maximum quantity load/reload of this fuel type at all applicable fuel locations for this entity. This field shall be represented by a 32-bit unsigned integer.

e) *Fuel Measurement Units*. This field shall identify the fuel measurement units. It shall be represented by an 8-bit enumeration (see [UID 328]).

f) *Fuel Type*. This field shall identify the type of fuel. It shall be represented by an 8-bit enumeration (see [UID 413]).

g) *Fuel Location*. This field shall indicate the location of the fuel as related to the entity. A zero indicates no specific location. It shall be represented by an 8-bit enumeration (see [UID 329]).

The format of the Storage Fuel Reload record shall be as shown in Table 116.

Table 116—Storage Fuel Reload record

Field size (bits)	Field name	Data type
32	Standard Quantity	32-bit unsigned integer
32	Maximum Quantity	32-bit unsigned integer
32	Standard Quantity Reload Time	32-bit unsigned integer
32	Maximum Quantity Reload Time	32-bit unsigned integer
8	Fuel Measurement Units	8-bit enumeration
8	Fuel Type	8-bit enumeration
8	Fuel Location	8-bit enumeration
8	Padding	8 bits unused
Total Storage Fuel Reload record size = 160 bits		

6.2.86 Supply Quantity record

Supply quantity shall be represented by the Supply Quantity record. This record shall contain fields specifying the type of supply and the quantity of that supply. The fields of this record are as follows:

a) *Supply Type*. This field shall specify by an Entity Type record (see 6.2.30).

b) *Quantity*. This field shall specify a 32-bit floating point number representing the number of units of a supply type. The unit measure depends on the supply type and shall use the SI units of measurement used for such supplies.

The format of the Supply Quantity record shall be as shown in Table 117.

Table 117—Supply Quantity record

Field size (bits)	Field name	Data type
64	Supply Type	Entity type—64-bit record
32	Quantity	32-bit floating point
Total Supply Quantity record size = 96 bits		

6.2.87 System Identifier record

The identification of the IFF emitting system shall be specified by a System Identifier record. This record shall specify the change status of this PDU, the system type, name, and mode. The fields of this record are as follows:

a) *System Type.* This field shall specify the general type of emitting system and shall be represented by a 16-bit enumeration (see [UID 82]).

b) *System Name.* This field shall specify a particular named type of system and shall be represented by a 16-bit enumeration (see [UID 83]).

c) *System Mode.* This field shall specify a mode of operation for the named system and shall be represented by an 8-bit enumeration (see [UID 84]).

d) *Change/Options.* This field shall specify the status of this PDU (existence of changes from previous, indicators for alternate fields, etc.). This field shall be represented by the Change/Options record (see 6.2.13).

The format of the System Identifier record shall be as shown in Table 118.

Table 118—System Identifier record

Field size (bits)	Field name	Data type
16	System Type	16-bit enumeration
16	System Name	16-bit enumeration

Table 118—System Identifier record *(continued)*

Field size (bits)	Field name	Data type
8	System Mode	8-bit enumeration
8	Change/Options	8-bit record
Total System Identifier record size = 48 bits		

6.2.88 Timestamp

6.2.88.1 General

Timestamping shall be used to indicate the reference time at which the data contained in the PDU was generated. The timestamp shall be calculated using a 32-bit unsigned integer representing units of time passed since the beginning of the current hour in the selected time reference. The least significant bit shall indicate whether the timestamp is absolute or relative. Additional requirements are found in 4.6.3.

6.2.88.2 Timestamp format

6.2.88.2.1 Absolute timestamp

An absolute timestamp should be used when the host clocks are synchronized, to reduce the processing required to compensate for time. The use of absolute time shall be signified by the least significant bit set to one. The reference time corresponding to the simulation time represented by the timestamp is the time when the data is valid.

6.2.88.2.2 Relative timestamp

A relative timestamp shall be used when simulation application clocks are not synchronized. Each simulation application shall keep relative time beginning with an arbitrary starting point. The time indicated by the timestamp shall be relative to the simulation application issuing the PDU. The use of the relative timestamp shall be signified by the least significant bit set to zero. Relative timestamps may be used when clocks are synchronized, but processing overhead is increased.

6.2.88.2.3 Scale

The scale of the time value contained in the most significant 31 bits of the timestamp shall be determined by letting zero represent the start of the hour and letting $2^{31} - 1$ represent one time unit before the start of the next hour. The next hour then starts back at zero. This results in each time unit representing exactly $3600/(2^{31})$ s (approximately 1.67638063 µs).

6.2.89 Total Record Sets record

The total number of record sets contained in a logical set of one or more PDUs shall be represented by the Total Record Sets record.

This record shall be used in a transfer transaction when all the record sets for a Transfer Ownership, Set Record-R, or Record-R PDU (see 5.9.4) cannot fit within the size limitations of a single PDU. This record may also be used for other purposes.

This record shall be included in each of the PDUs of a logical set of PDUs containing record sets. Although the corresponding record set is included in each of the PDUs, it shall only be counted once in the Total Record Sets field. All other types of record sets shall appear only once in the PDUs. This information shall be used by the receiving application to determine when the complete set of record sets has been received (i.e., when all PDUs in the logical set have been received). The record set associated with the Total Record Sets record may be placed anywhere within the record sets area of a PDU; i.e., the record set may be the first or last record set in a PDU, or it may be in the middle. The Total Record Sets record is optional if all record sets fit within a single PDU.

The Total Record Sets record shall contain the following fields:

a) *Total Record Sets*. This field shall specify the total number of record sets, possibly spanning multiple PDUs, which are being transferred in conjunction with the current Transfer Ownership, Set Record-R, or Record-R PDU.
b) *Padding*. This field shall consist of 16 bits of padding.

The format of the Total Record Sets record shall be as shown in Table 119.

Table 119—Total Record Sets record

Field size (bits)	Field name	Data type
16	Total Record Sets	16-bit unsigned integer
16	Padding	16 bits unused
Total Record Sets record size = 32 bits		

6.2.90 Track/Jam Data record

The Track/Jam Data record identifies an entity tracked or illuminated or an emitter beam targeted with jamming. Use of the Track/Jam Data record in the EE PDU is described in 5.7.3 and 7.6.2. The record consists of an Entity ID, an Emitter Number, and a Beam Number:

a) *Entity ID*. This field shall specify an entity by Simulation Address and Entity Number and shall be represented by an Entity Identifier record (see 6.2.28).
b) *Emitter Number*. This field shall specify an emitter system associated with the entity. This field shall be represented by an 8-bit unsigned integer.
c) *Beam Number*. This field shall specify a beam associated with the emitter system. This field shall be represented by an 8-bit unsigned integer.

The format of the Track/Jam Data record shall be as shown in Table 120.

Table 120—Track/Jam Data record

Field size (bits)	Field name	Data type
48	Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Emitter Number	8-bit unsigned integer
8	Beam Number	8-bit unsigned integer
Total Track/Jam Data record size = 64 bits		

6.2.91 UA Fundamental Parameter Data record

The UA Fundamental Parameter Data record contains regeneration parameters for active emission systems that are variable throughout a scenario. The variations of these parameters are dependent on the actions of the operators in the emitting entity simulation. The fields of this record are as follows:

a) *Active Emission Parameter Index (Filename).* This field indicates which database record (database file) shall be used in the definition of intentional emissions of the entity. For example, a parameter index (filename) from the Active Emissions Database (AEDB) (providing active emitter pulse data such as pulse type, pulse duration, pulse interval, pulse frequency content, source power, and pulse beam pattern) would be referenced here. This field shall be represented by a 16-bit enumeration (see [UID 146]).

b) *Scan Pattern.* This field shall specify the type of Scan Pattern. If scan patterns are not used, this field shall be zero. This field shall be represented by a 16-bit enumeration (see [UID 147]).

c) *Beam Center Azimuth (Horizontal Bearing).* This field shall specify the center azimuthal bearing of the main beam (as opposed to sidelobes) in relation to the emitter coordinate system. The azimuth center for 360° bearing shall be 0°. This field shall be represented by a 32-bit floating point number representing units of radians. Omnidirectional beams shall have a bearing of 0°.

d) *Azimuthal Beamwidth (Horizontal Beamwidth).* This field shall specify the horizontal beamwidth of the main beam. Beamwidth will be measured at the 3 dB down point of peak radiated power. This field shall be represented by a 32-bit floating point number representing units of radians. Omnidirectional beams shall have a beam width of 0°.

e) *Beam Center Depression/Elevation (D/E).* This field shall specify the center of the D/E angle of the main beam relative to the stabilized D/E angle of the target (i.e., assume the target is stabilized and parallel to Earth for determining D/E angle). This field shall be represented by a 32-bit floating point number representing units of radians. A depression angle shall be defined as a downward angle and assigned a negative value. An elevation angle shall be defined as an upward angle and assigned a positive value. Omnidirectional beams shall have a beam center D/E angle of 0°.

f) *D/E Beamwidth (Vertical Beamwidth).* This field shall specify the vertical beamwidth of the main beam. Beamwidth shall be measured at the 3 dB down point of peak radiated power. This field shall be represented by a 32-bit floating point number representing units of radians. Omnidirectional beams shall have a beamwidth of 0°.

The format of the UA Fundamental Parameter Data record shall be as shown in Table 121.

Table 121—UA Fundamental Parameter Data record

Field size (bits)	Field name	Data type
16	Active Emission Parameter Index (Filename)	16-bit enumeration
16	Scan Pattern	16-bit enumeration
32	Beam Center Azimuth (Horizontal Bearing)	32-bit floating point
32	Azimuthal Beamwidth (Horizontal Beamwidth)	32-bit floating point
32	Beam Center D/E	32-bit floating point
32	D/E Beamwidth (Vertical Beamwidth)	32-bit floating point
Total UA Fundamental Parameter Data record size = 160 bits		

6.2.92 Unattached Identifier record

The unique designation of one or more unattached radios in an event or exercise shall be specified by an Unattached Identifier record. The Unattached Identifier record shall consist of a Simulation Address (see 6.2.80) and a 16-bit Reference Number. For intercom simulations using either of the two methods mentioned in 5.8.1, the unattached identifier refers to the intercom communications device. The format of the Unattached Identifier record shall be as shown in Table 122.

Table 122—Unattached Identifier record

Field size (bits)	Field name	Data type
32	Simulation Address	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
16	Reference Number	16-bit unsigned integer
Total Unattached Identifier record size = 48 bits		

6.2.93 Variable Datum record

Variable datum information shall be represented using the Variable Datum record. This record shall specify the variable datum type, the datum length, and the value for that variable datum type. The fields of this record are as follows:

a) *Variable Datum ID*. This field shall specify the type of variable datum to be communicated. This field shall consist of a 32-bit enumeration (see [UID 66]).

b) *Variable Datum Length*. This field shall specify the number of bits used in the Variable Datum Value. The length shall be specified using a 32-bit unsigned integer field.

c) *Variable Datum Value*. This field shall specify the value for a particular Variable Datum ID. The field format shall depend on the type of datum as specified by the Variable Datum ID. Variable datums can be any number of bits long.

d) *Padding*. Padding shall be added to end the record on a 64-bit boundary. Padding bits are not included in the Variable Datum Length.

The format of the Variable Datum record shall be as shown in Table 123.

Table 123—Variable Datum record

Field size (bits)	Field name	Data type
32	Variable Datum ID	32-bit enumeration
32	Variable Datum Length (K)	32-bit unsigned integer
K	Variable Datum Value	K bits
P	Padding	Padding to 64-bit boundary—P bits
Total Variable Datum record size = 64 + K + P bits where K is the variable datum length P is the number of padding bits, which is $\lceil K/64 \rceil 64 - K$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

6.2.94 Variable Parameter record

6.2.94.1 General

The specification of additional information associated with an entity or detonation, not otherwise accounted for in a PDU, shall be represented by a Variable Parameter record. This record shall provide information on articulated parts, attached parts, and other data associated with an entity or detonation. This record shall be contained in the Variable Parameter records field of PDUs that have that field. (See Annex I for further explanation, figures, and examples.) One or more Variable Parameter records may be contained in the Variable Parameter records field up to the maximum size of the PDU.

The Variable Parameter record contains the following fields:

a) *Record Type.* This field shall designate the identification of the Variable Parameter record. This field shall be represented by an 8-bit enumeration (see [UID 56]).

b) *Record-Specific fields*. The remainder of the Variable Parameter record shall vary in format depending on whether it is an articulated or an attached part or some other type of Variable Parameter record.

The Variable Parameter record format is defined in Table 124.

Table 124—Variable Parameter record

Field size (bits)	Field name	Data type
8	Record Type	8-bit enumeration
120	Record-Specific fields	120 bits
Total Variable Parameter record size = 128 bits		

6.2.94.2 Articulated Part VP record

The specification of articulated parts for movable parts and a combination of moveable/attached parts of an entity shall be represented by an Articulated Part VP record. This record shall specify whether or not a change has occurred, the part identification of the articulated part to which it is attached, and the type and value of each parameter (see Annex I for further explanation, figures, and examples). The Articulated Part VP record is a type of Variable Parameter record. It shall contain the following fields:

a) *Record Type*. This field shall designate the identification of an Articulated Part VP record. This field shall be represented by an 8-bit enumeration (see [UID 56]).

b) *Change Indicator*. This field shall indicate the change of any parameter for any articulated part. This field shall be specified by an 8-bit unsigned integer. This field shall be set to zero for each exercise and sequentially incremented by one for each change in articulation parameters. In the case where all possible values are exhausted, the numbers shall be reused beginning at zero.

c) *ID—Part Attached To*. This field shall specify the identification of the articulated part to which this articulation parameter is attached. This field shall be specified by a 16-bit unsigned integer. This field shall contain the value zero if the articulated part is attached directly to the entity. See I.2.2.2.

d) *Parameter Type*. This field shall specify the type of parameter represented and shall be specified by a 32-bit enumeration (see [UID 58-59]). Parameter types are defined in Annex I.

e) *Parameter Value*. This field shall specify the parameter value and shall be specified by a 32-bit floating point number. The definition of this field shall be determined based on the type of parameter specified in the Parameter Type field [see item d) above].

The format of the Articulated Part VP record shall be as shown in Table 125.

Table 125—Articulated Part VP record

Field size (bits)	Field name	Data type
8	Record Type = 0	8-bit enumeration
8	Change Indicator	8-bit unsigned integer
16	ID—Part Attached To	16-bit unsigned integer
32	Parameter Type	32-bit enumeration

Table 125—Articulated Part VP record *(continued)*

Field size (bits)	Field name	Data type
32	Parameter Value	32-bit floating point
32	Padding	32 bits unused
Total Articulated Part VP record size = 128 bits		

6.2.94.3 Attached Part VP record

The specification of removable parts that may be attached to an entity shall be represented by an Attached Part VP record. This record shall specify whether or not the part is currently attached, the part identification, and the type and value of each parameter (see Annex I for further explanation, figures, and examples). The Attached Part VP record is a type of Variable Parameter record. It shall contain the following fields:

a) *Record Type.* This field shall designate the identification of an Attached Part VP record. This field shall be represented by an 8-bit enumeration (see [UID 56]).

b) *Detached Indicator.* This field shall indicate whether an attached part is attached or detached. This field shall be specified by an 8-bit enumeration (see [UID 415]). This field shall be set to Attached (0) to indicate the part is attached and to Detached (1) if the part becomes detached. (See I.2.3.1 for transitions between the attached and detached states.)

c) *ID—Part Attached To.* This field shall specify the identification of the articulated or attached part to which this attached part is attached. This field shall contain the value zero if the attached part is attached directly to the entity. This field shall be specified by a 16-bit unsigned integer. See I.2.2.2.

d) *Parameter Type.* This field shall specify the location (or station) to which the part is attached and shall be specified by a 32-bit enumeration (see [UID 57]). Parameter types are defined in Annex I.

e) *Attached Part Type*. This field shall specify the Entity Type record enumeration of the attached part.

The format of the Attached Part VP record shall be as shown in Table 126.

Table 126—Attached Part VP record

Field size (bits)	Field name	Data type
8	Record Type = 1	8-bit enumeration
8	Detached Indicator	8-bit enumeration
16	ID—Part Attached To	16-bit unsigned integer
32	Parameter Type	32-bit enumeration
64	Attached Part Type	64-bit Entity Type record
Total Attached Part VP record size = 128 bits		

6.2.94.4 Entity Association VP Record

6.2.94.4.1 Purpose

The association or disassociation of two entities, or an entity with another object, shall be communicated using the Entity Association VP record. The association may be physical or nonphysical in nature. The Entity Association VP record shall be included in the Entity State PDU of each entity involved in an association as defined herein. For example, if an aircraft entity is towing a decoy entity, the Entity Association VP record shall be contained in both the aircraft and the towed or air-launched standoff decoy Entity State PDUs. Reference to an Entity State PDU shall also mean an Entity State Update PDU.

6.2.94.4.2 General rules

The following general rules shall apply:

a) *Physical Association*. A physical association is where two entities, or an entity and another object, are physically connected to each other. This connection may be direct where the surface of one entity is in physical contact with the surface of another entity or object, or it may be indirect; in which case, the two entities, or entity and another object, are connected by a physical linkage of some sort such as a cable or rope. There may be several physical linkages present between two entities, or an entity and another object, such as a ship moored at a pier. An Entity Association VP record may be included in the Entity State PDU when it is desired to indicate each of the other entities or objects with which it is in physical contact.

b) *Functional Association*. A functional association is where two or more entities are associated with each other to perform a task or mission. Examples include aircraft and tanks in formation, the trucks of a convoy on a road, the ships in a naval battle group, or a platoon of soldiers. It also includes where one entity is controlling another entity, such as a soldier entity that is controlling an Unmanned Aerial Vehicle (UAV) entity. An Entity Association VP record may be included in the Entity State PDU for each entity that is functionally associated with a group of entities to indicate its position in the group and to identify the group.

c) *Disassociation*. When an entity is no longer associated with another entity or object, the following requirements shall be met:

   1) An Entity State PDU Entity shall be issued with the Association Status field set to Association Broken (3).

   2) The next subsequent Entity State PDU shall not contain the Entity Association VP record.

d) Associations and disassociations for an entity may occur more than once during the course of an exercise as reflected by the inclusion and removal of Entity Association VP records in an Entity State PDU.

6.2.94.4.3 Fields

The Entity Association VP record shall contain the following fields:

a) *Record Type*. This field shall identify the record as an Entity Association VP record. It shall be represented by an 8-bit enumeration (see [UID 56]).

b) *Change Indicator*. This field shall indicate whether this VP record has changed since the last issuance. It shall be represented by an 8-bit enumeration (see [UID 320]).

c) *Association Status*. This field shall indicate the association status between two entities, or an entity with another object. It shall be represented by an 8-bit enumeration (see [UID 319]).

d) *Association Type*. This field shall indicate the type of association that exists between two entities, or an entity with another object. It shall be represented by an 8-bit enumeration (see [UID 323]).

e) *Entity/Object ID*. This field shall indicate the Object ID of the entity or other object associated with this entity. It shall be represented by an Entity Identifier record (see 6.2.28) or Object Identifier record (see 6.2.63).

f) *Own Station Location*. This field shall indicate the station location on one's own entity where a physical connection is attached, if known. If there is more than one physical connection, then each connection that is described shall be included in a separate Entity Association VP record. This field shall be represented by a 16-bit enumeration (see [UID 212]).

g) *Physical Connection Type*. This field shall indicate the type of physical connection, if any, between this entity and another entity or object. If the association does not involve a physical connection, this field shall be set to Not Specified (0). It shall be represented by an 8-bit enumeration (see [UID 324]).

h) *Group Member Type*. This field shall indicate the type of member the entity is within the group. If this entity is not part of a group, this field shall be set to zero. It shall be represented by an 8-bit enumeration (see [UID 321]).

i) *Group Number*. This field shall indicate the group, if any, to which an entity belongs. If this entity is not part of a group, this field shall be set to zero. The manner in which group numbers are assigned is outside the scope of this standard. It shall be represented by a 16-bit unsigned integer.

The format of the Entity Association VP record shall be as shown in Table 127.

Table 127—Entity Association VP record

Field size (bits)	Field name	Data type
8	Record Type = 4	8-bit enumeration
8	Change Indicator	8-bit enumeration
8	Association Status	8-bit enumeration
8	Association Type	8-bit enumeration
48	Entity/Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity/Object Number—16-bit unsigned integer
16	Own Station Location	16-bit enumeration
8	Physical Connection Type	8-bit enumeration
8	Group Member Type	8-bit enumeration
16	Group Number	16-bit unsigned integer
Total Entity Association VP record size = 128 bits		

6.2.94.5 Entity Type VP record

The Entity Type for an entity when it is different than the entity type value contained in the Entity Type field of the Entity State PDU shall be communicated using the Entity Type VP record.

The Entity Type VP record shall contain the following fields:

a) *Record Type*. This field shall identify the record as an Entity Type VP record. It shall be represented by an 8-bit enumeration (see [UID 56]).

b) *Change Indicator*. This field shall indicate whether this VP record has changed since the last issuance. It shall be represented by an 8-bit enumeration (see [UID 320]).

c) *Entity Type*. This field shall contain an Entity Type record for the Entity Type associated with an entity (see 6.2.30).

The format of the Entity Type VP record shall be as shown in Table 128.

Table 128—Entity Type VP record

Field size (bits)	Field name	Data type
8	Record Type = 3	8-bit enumeration
8	Change Indicator	8-bit enumeration
64	Entity Type	64-bit Entity Type record
16	Padding	16 bits unused
32	Padding	32 bits unused
Total Entity Type VP record size = 128 bits		

6.2.94.6 Separation VP record

The physical separation of an entity from another entity shall be communicated using the Separation VP record. The Separation VP record also provides the capability to specifically identify the station location on the parent entity that the entity separated from. Examples include the launch of an air-to-surface missile from the wing station of a fighter aircraft, the separation of a stage of a multistage missile, and the separation of a component of a smart weapon.

The Separation VP record shall contain the following fields:

a) *Record Type*. This field shall identify the record as a Separation VP record. It shall be represented by an 8-bit enumeration (see [UID 56]).

b) *Reason for Separation*. This field shall indicate the reason for the separation. It shall be represented by an 8-bit enumeration (see [UID 282]).

c) *Pre-Entity Indicator*. This field shall indicate whether the entity existed prior to the separation and, if so, in what manner. It shall be represented by an 8-bit enumeration (see [UID 283]).

d) *Parent Entity ID*. This field shall indicate the Entity ID of the parent entity. It shall be represented by an Entity Identifier record (see 6.2.28).

e) *Station Location*. This field shall indicate the station that this entity was attached to prior to separation if known. It shall be represented by the Named Location Identification record (see 6.2.62).

The format of the Separation VP record shall be as shown in Table 129.

Table 129—Separation VP record

Field size (bits)	Field name	Data type
8	Record Type = 2	8-bit enumeration
8	Reason for Separation	8-bit enumeration
8	Pre-Entity Indicator	8-bit enumeration
8	Padding	8 bits unused
48	Parent Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Padding	16 bits unused
32	Station Location	Station Name—16-bit enumeration
		Station Number—16-bit unsigned integer
Total Separation VP record size = 128 bits		

6.2.95 Variable Transmitter Parameters record

A Variable Transmitter Parameters (VTP) record shall be identified by a record type. One or more VTP records may be associated with a radio system, and the same VTP record may be associated with multiple radio systems. Specific VTP records applicable to a radio system are identified in the subclause that defines the radio system's unique requirements in Annex C. The fields of the VTP record shall be as follows:

a) *Record Type*. This field shall specify the type of VTP record. This field shall be represented by a 32-bit enumeration (see [UID 66]).

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. The value of the Record Length shall be the sum of the sizes of the Record Type field, the Record Length field, all Record-Specific fields, and any padding required to end the record on a 64-bit boundary. All Record Length values shall be a multiple of 8. The Record Length shall be represented by a 16-bit unsigned integer.

c) *Record-Specific Fields*. The specific transmitter parameters associated with a VTP record are based on the record type and contained in these field(s).

d) *Padding*. A Padding field shall be added as needed to reach a 64-bit boundary.

The format of the Variable Transmitter Parameters record shall be as shown in Table 130.

Table 130—Variable Transmitter Parameters record

Field size (bits)	Field name	Data type
32	Record Type	32-bit enumeration
16	Record Length	16-bit unsigned integer ($6 + K + P$)
$8K$	Record-Specific fields	K octets
$8P$	Padding	Padding to 64-bit boundary—P octets
Total Variable Transmitter Parameters record size = $48 + 8K + 8P$ bits where K is the number of octets in the Record-Specific fields P is the number of padding octets, which is $\lceil (6 + K) / 8 \rceil 8 - (6 + K)$ $\lceil x \rceil$ is the largest integeris the largest integer $< x + 1$.		

6.2.96 Vector record

Vector values for entity coordinates, linear acceleration, and linear velocity shall be represented using a Vector record. This record shall consist of three fields, each a 32-bit floating point number. The unit of measure represented by these fields shall depend on the information represented. The values utilizing the Vector record are as follows:

a) *Entity Coordinate Vector.* Location relative to a particular entity shall be specified with respect to the origin of the entity's coordinate system. The entity coordinate system shall be as specified in 1.6.3.

b) *Linear Acceleration Vector.* Linear acceleration shall be represented as a vector with components in either world coordinate system or entity's coordinate system depending on the value in the Dead Reckoning Algorithm field. Each vector component shall represent acceleration in meters per second squared.

c) *Linear Velocity Vector.* Linear velocity shall be represented as a vector with three components in either world coordinate system or entity's coordinate system depending on the value in the Dead Reckoning Algorithm field. Each vector component shall represent velocity in meters per second.

The format of the Vector record shall be as shown in Table 131.

Table 131—Vector record

Field size (bits)	Field name	Data type
32	First vector component	32-bit floating point

Table 131—Vector record *(continued)*

Field size (bits)	Field name	Data type
32	Second vector component	32-bit floating point
32	Third vector component	32-bit floating point
Total Vector record size = 96 bits		

6.2.97 Vectoring Nozzle System Data record

The SEES Vectoring Nozzle System Data record contains operational data for describing the vectoring nozzle systems being simulated. This information shall be provided for each nozzle system defined. The fields of this record are as follows:

a) *Horizontal Deflection Angle.* This field shall specify the nozzle deflection angle (in degrees) in the horizontal body axis of the entity. The field is represented by a 32-bit floating point number.

b) *Vertical Deflection Angle.* This field shall specify the nozzle deflection angle (in degrees) in the vertical body axis of the entity. The field is represented by a 32-bit floating point number.

The format of the Vectoring Nozzle System Data record shall be as shown in Table 132.

Table 132—Vectoring Nozzle System Data record

Field size (bits)	Field name	Data type
32	Horizontal Deflection Angle	32-bit floating point
32	Vertical Deflection Angle	32-bit floating point
Total Vectoring Nozzle System Data record size = 64 bits		

6.2.98 World Coordinates record

Location of the origin of the entity's or object's coordinate system, target locations, detonation locations, and other points shall be specified by a set of three coordinates: X, Y, and Z, represented by 64-bit floating point numbers. The world coordinate system shall be as specified in 1.6.3. The format of the World Coordinates record shall be as shown in Table 133.

Table 133—World Coordinates record

Field size (bits)	Field name	Data type
64	X-coordinate	64-bit floating point
64	Y-coordinate	64-bit floating point
64	Z-coordinate	64-bit floating point
Total World Coordinates record size = 192 bits		

6.3 General requirements

6.3.1 General

General requirements related to the transmission and receipt of PDUs not otherwise specified in IEEE Std 1278.2 are contained in this subclause.

6.3.2 Alignment

Each field in a PDU starts at an offset from the beginning of the PDU that is a multiple of its size. Each record in a PDU shall start at an offset that is a multiple of the size of the largest field in the record. If the PDU contains a 64-bit field, then the length of the PDU shall be a multiple of 64 bits. Otherwise, the length of the PDU shall be a multiple of 32 bits. Padding bits shall be used to achieve these offsets.

6.3.3 Maximum PDU size

The maximum size allowed for an individual PDU shall be:

MAX_PDU_SIZE_OCTETS = 8192

MAX_PDU_SIZE_BITS = 65 536

A DIS simulation shall not send PDUs that exceed the Maximum PDU Size. There is no requirement for simulations to have the capability to send PDUs of the maximum size. A DIS simulation shall be able to receive and process PDUs up to the Maximum PDU Size.

6.3.4 Smallest Maximum Transmission Unit (MTU)

Because most networks do not handle PDUs of the maximum size as efficiently as they do smaller PDUs, the use of large PDUs requires careful consideration. The most efficient maximum size for a PDU is the smallest MTU (Maximum Transmission Unit) of all the networks being used for a given DIS exercise. Simulations shall implement a Smallest MTU Parameter. This parameter shall be set to the smallest MTU in the network minus the size of network and encryptor protocol headers. The Smallest MTU Parameter shall be identified by the symbolic name SMALLEST_MTU_OCTETS. (See 6.1.8 for parameter details and default values.)

If the PDU size exceeds the Smallest MTU Parameter value, this will cause the network to fragment and reassemble the PDU. While this may be handled efficiently by most networks, it should usually be avoided. The exception is a PDU that contains information best sent as a coherent set of data because it would be too unwieldy to break it into multiple smaller PDUs. An example is a naval ship entity with many radar systems on board that are required to be reported in an EE PDU. It is usually better to transmit all the radars in one EE PDU even if it exceeds the Smallest MTU Parameter value so that the data is received as a single full PDU update rather than as multiple PDUs. Otherwise, if the data is broken into multiple PDUs, their may be coherency problems as the PDUs may arrive out of order or some of them be lost in transit, which complicates the processing task.

However, other large sets of information can easily be broken into multiple PDUs without causing coherency problems. For example, a large amount of Signal PDU or Set Data PDU information should be broken into multiple PDUs if it does not affect data coherency. When a large set of PDU information is divided into smaller PDUs for transmission, each PDU’s size should not exceed the Smallest MTU Parameter value.

6.3.5 PDU bundling

Network efficiency may be enhanced with PDU bundling. This is the process of concatenating two or more PDUs into a single network datagram so that they may be transmitted and relayed through the network in a single operation. The requirements related to PDU bundling are specified in IEEE Std 1278.2. Detailed information on bundling as it relates to DIS PDUs is described in Murray [B12].

7 DIS PDU contents

7.1 Introduction

The contents of the PDUs for DIS are described in 7.2 through 7.12 and Clause 9.

7.2 Entity Information/Interaction protocol family

7.2.1 General

The PDUs of the Entity Information/Interaction protocol family are described in 7.2.2 through 7.2.6.

7.2.2 Entity State PDU

Information about a particular entity shall be communicated by issuing an Entity State PDU. See 5.3.2 for specific requirements on the use of the Entity State PDU. The Entity State PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Entity ID.* This field shall identify the entity issuing the PDU and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Force ID.* This field shall identify the force to which the issuing entity belongs and shall be represented by an 8-bit enumeration (see [UID 6]).

d) *Number of Variable Parameter Records.* This field shall specify the number of Variable Parameter records present. This field shall be represented by an 8-bit unsigned integer (see Annex I).

e) *Entity Type.* This field shall identify the entity type to be displayed by members of the same force as the issuing entity. This field shall be represented by an Entity Type record (see 6.2.30).

f) *Alternate Entity Type.* This field shall identify the entity type to be displayed by members of forces other than that of the issuing entity and shall be represented by an Entity Type record (see 6.2.30).

g) *Entity Linear Velocity.* This field shall specify an entity's linear velocity. The coordinate system for an entity's linear velocity depends on the dead reckoning algorithm used. This field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96].

h) *Entity Location.* This field shall specify an entity's physical location in the simulated world and shall be represented by a World Coordinates record (see 6.2.98).

i) *Entity Orientation.* This field shall specify an entity's orientation and shall be represented by an Euler Angles record (see 6.2.32).

j) *Entity Appearance.* This field shall specify the dynamic changes to the entity's appearance attributes. This field shall be represented by an Entity Appearance record (see 6.2.26).

k) *Dead Reckoning Parameters.* This field will be used to provide parameters for dead reckoning the position and orientation of the entity. The dead reckoning algorithm in use, entity acceleration, and angular velocity shall be included as part of the dead reckoning parameters. One-hundred-twenty bits are reserved for other parameters as described in item f) in 5.3.2.3.2:

1) Dead Reckoning Algorithm. This field shall specify the dead reckoning algorithm in use by the issuing entity and shall be represented by an 8-bit enumeration (see [UID 44]).
2) Other Parameters. This field shall specify other required dead reckoning parameters and shall consist of 120 bits. See E.8 for rules associated with the use of the Other Parameters field.
3) Entity Linear Acceleration. This field shall specify an entity's linear acceleration. This field shall be represented by a Linear Acceleration Vector record [see item b) in 6.2.96].
4) Entity Angular Velocity. This field shall specify an entity's angular velocity and shall be represented by an Angular Velocity Vector record (see 6.2.7).

l) *Entity Marking*. This field shall identify any unique markings on an entity (for example, a bumper number or country symbol). This field shall be represented by an Entity Marking record (see 6.2.29).

m) *Capabilities*. This field shall specify the entity's capabilities. This field shall be represented by an Entity Capabilities record (see 6.2.27).

n) *Variable Parameter Records*. This field shall specify the parameter values for each Variable Parameter record that is included (see 6.2.94 and Annex I).

The format of the Entity State PDU shall be as shown in Table 134.

Table 134—Entity State PDU

Field size (bits)	**Entity State PDU fields**	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 1
		Protocol Family—8-bit enumeration = 1
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Force ID	8-bit enumeration
8	Number of Variable Parameter Records (*N*)	8-bit unsigned integer

Table 134—Entity State PDU *(continued)*

Field size (bits)	Entity State PDU fields	
64	Entity Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
64	Alternate Entity Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
96	Entity Linear Velocity	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
192	Entity Location	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
96	Entity Orientation	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
32	Entity Appearance	32-bit record
320	Dead Reckoning Parameters	Dead Reckoning Algorithm—8-bit enumeration
		Other Parameters—120 bits
		Entity Linear Acceleration—3 × 32-bit floating point
		Entity Angular Velocity—3 × 32-bit floating point
96	Entity Marking	Character Set—8-bit enumeration
		11, 8-bit unsigned integers
32	Capabilities	32-bit record
128	Variable Parameter record #1	Record Type—8-bit enumeration
		Record-Specific fields—120 bits

Table 134—Entity State PDU *(continued)*

Field size (bits)	Entity State PDU fields	
• • •		
128	Variable Parameter record #*N*	Record Type—8-bit enumeration
		Record-Specific fields—120 bits
Total Entity State PDU size = 1152 + 128*N* bits where *N* is the number of Variable Parameter records		

7.2.3 Collision PDU

Collisions between entities shall be communicated by issuing a Collision PDU. See 5.3.3 for specific requirements on the use of the Collision PDU. The Collision PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Issuing Entity ID*. This field shall identify the entity that is issuing the PDU and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Colliding Entity ID*. This field shall identify the entity that has collided with the issuing entity (see 5.3.3.4). This field shall be represented by an Entity Identifier record (see 6.2.28).

d) *Event ID*. This field shall contain an identification generated by the issuing simulation application to associate related collision events. This field shall be represented by an Event Identifier record (see 6.2.33).

e) *Collision Type*. This field shall identify the type of collision. The Collision Type field shall be represented by an 8-bit enumeration (see [UID 189]).

f) *Velocity*. This field shall contain the velocity (at the simulation time the collision is detected) of the issuing entity. The velocity shall be represented in world coordinates. This field shall be represented by the Linear Velocity Vector record [see item c) in 6.2.96].

g) *Mass*. This field shall contain the mass of the issuing entity and shall be represented by a 32-bit floating point number representing kilograms.

h) *Location*. This field shall specify the location of the collision with respect to the entity with which the issuing entity collided. The Location field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

The format of the Collision PDU shall be as shown in Table 135.

Table 135—Collision PDU

Field size (bits)	Collision PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 4
		Protocol Family—8-bit enumeration = 1
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Issuing Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Colliding Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
8	Collision Type	8-bit enumeration
8	Padding	8 bits unused
96	Velocity	*X*-component—32-bit floating point
		Y-component—32-bit floating point
		Z-component—32-bit floating point
32	Mass	32-bit floating point
96	Location (with respect to entity)	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
Total Collision PDU size = 480 bits		

7.2.4 Collision-Elastic PDU

Information about elastic collisions in a DIS exercise shall be communicated using a Collision-Elastic PDU. See 5.3.4 for specific requirements on the use of the Collision-Elastic PDU. The Collision-Elastic PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Issuing Entity ID.* This field shall identify the entity that is issuing the PDU and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Colliding Entity ID.* This field shall identify the entity that has collided with the issuing entity. This field shall be a valid identifier of an entity or server capable of responding to the receipt of this Collision-Elastic PDU. This field shall be represented by an Entity Identifier record (see 6.2.28).

d) *Event ID.* This field shall contain an identification generated by the issuing simulation application to associate related collision events. This field shall be represented by an Event Identifier record (see 6.2.33).

e) *Contact Velocity.* This field shall contain the velocity at the simulation time the collision is detected at the point the collision is detected. The velocity shall be represented in world coordinates. This field shall be represented by the Linear Velocity Vector record [see item c) in 6.2.96].

f) *Mass.* This field shall contain the mass of the issuing entity and shall be represented by a 32-bit floating point number representing kilograms.

g) *Location of Impact.* This field shall specify the location of the collision with respect to the entity with which the issuing entity collided. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

h) *Collision Intermediate Result.* These six records represent the six independent components of a positive semidefinite matrix formed by premultiplying and postmultiplying the tensor of inertia, by the antisymmetric matrix generated by the moment arm and shall be represented by 32-bit floating point numbers (see 5.3.4.4).

i) *Unit Surface Normal.* This record shall represent the normal vector to the surface at the point of collision detection. The surface normal shall be represented in world coordinates. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

j) *Coefficient of Restitution.* This field shall represent the degree to which energy is conserved in a collision and shall be represented by a 32-bit floating point number. In addition, it represents a free parameter by which simulation application developers may “tune” their collision interactions.

The format of the Collision-Elastic PDU shall be as shown in Table 136.

Table 136—Collision-Elastic PDU

Field size (bits)	Collision-Elastic PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 66
		Protocol Family—8-bit enumeration = 1
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Issuing Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Colliding Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
16	Padding	16 bits unused
96	Contact Velocity	*X*-component—32-bit floating point
		Y-component—32-bit floating point
		Z-component—32-bit floating point
32	Mass	32-bit floating point
96	Location of Impact	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
32	Collision Intermediate Result-*XX*	32-bit floating point
32	Collision Intermediate Result-*XY*	32-bit floating point
32	Collision Intermediate Result-*XZ*	32-bit floating point
32	Collision Intermediate Result-*YY*	32-bit floating point
32	Collision Intermediate Result-*YZ*	32-bit floating point
32	Collision Intermediate Result-*ZZ*	32-bit floating point

Table 136—Collision-Elastic PDU ***(continued)***

Field size (bits)	Collision-Elastic PDU fields	
96	Unit Surface Normal	*X*-component—32-bit floating point
		Y-component—32-bit floating point
		Z-component—32-bit floating point
32	Coefficient of Restitution	32-bit floating point
Total Collision-Elastic PDU size = 800 bits		

7.2.5 Entity State Update PDU

Nonstatic information about a particular entity may be communicated by issuing an Entity State Update PDU. See 5.3.5 for specific requirements on the use of the Entity State Update PDU. The Entity State Update PDU shall contain the following fields and may be used when a simulation application is using dead reckoning algorithm DRM (FPW) [defined in Annex E]:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).
b) *Entity ID.* This field shall identify the entity issuing the PDU and shall be represented by an Entity Identifier record (see 6.2.28).
c) *Number of Variable Parameter Records.* This field shall specify the number of Variable Parameter records present. This field shall be represented by an 8-bit unsigned integer (see Annex I).
d) *Entity Linear Velocity.* This field shall specify an entity's linear velocity. The coordinate system for an entity's linear velocity depends on the dead reckoning algorithm used. This field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96].
e) *Entity Location.* This field shall specify an entity's physical location in the simulated world and shall be represented by a World Coordinates record (see 6.2.98).
f) *Entity Orientation.* This field shall specify an entity's orientation and shall be represented by an Euler Angles record (see 6.2.32).
g) *Entity Appearance.* This field shall specify the dynamic changes to the entity's appearance attributes. This field shall be represented by an Entity Appearance record (see 6.2.26).
h) *Variable Parameter Records.* This field shall specify the parameter values for each Variable Parameter record that is included (see 6.2.94 and Annex I).

The format of the Entity State Update PDU shall be as shown in Table 137.

Table 137—Entity State Update PDU

Field size (bits)	**Entity State Update PDU fields**	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 67
		Protocol Family—8-bit enumeration = 1
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Padding	8 bits unused
8	Number of Variable Parameter Records (*N*)	8-bit unsigned integer
96	Entity Linear Velocity	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
192	Entity Location	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
96	Entity Orientation	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
32	Entity Appearance	32-bit record
128	Variable Parameter record #1	Record Type—8-bit enumeration
		Record-Specific fields—120 bits
	• • •	

Table 137—Entity State Update PDU *(continued)*

Field size (bits)	Entity State Update PDU fields	
128	Variable Parameter record #*N*	Record Type—8-bit enumeration
		Record-Specific fields—120 bits
Total Entity State Update PDU size = 576 + 128*N* bits where *N* is the number of Variable Parameter records		

7.2.6 Attribute PDU

Information about individual attributes for a particular entity, other object, or event may be communicated using an Attribute PDU. The Attribute PDU shall not be used to exchange data available in any other PDU except where explicitly mentioned in the PDU issuance instructions within this standard. See 5.3.6 for specific requirements on the use of the Attribute PDU.

The Attribute PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Originating Simulation Address*. This field shall identify the simulation issuing the Attribute PDU. It shall be represented by a Simulation Address record (see 6.2.80).

c) *Attribute Record PDU Type.* This field shall represent the type of the PDU that is being extended or updated, if applicable. It shall be represented by an 8-bit enumeration (see [UID 4]).

d) *Attribute Record Protocol Version.* This field shall indicate the Protocol Version associated with the Attribute Record PDU Type. It shall be represented by an 8-bit enumeration (see [UID 5]).

e) *Master Attribute Record Type.* This field shall contain the Attribute record type of the Attribute records in the PDU if they all have the same Attribute record type. It shall be represented by a 32-bit enumeration (see [UID 66]).

f) *Action Code.* This field shall identify the action code applicable to this Attribute PDU. The Action Code shall apply to all Attribute records contained in the PDU. It shall be represented by an 8-bit enumeration (see [UID 295]).

g) *Number of Attribute Record Sets.* This field shall specify the number of Attribute Record Sets that make up the remainder of the PDU. It shall be represented by a 16-bit unsigned integer.

h) *Attribute Record Sets*: Each Attribute Record Set (see 5.3.6.3) shall contain the following fields:

   1) Entity/Object ID. This record shall indicate the entity or object to which all Attribute records in an Attribute Record Set apply and shall be represented by an Entity Identifier record (see 6.2.28) or an Object Identifier record (see 6.2.63).

   2) Number of Attribute Records. This field shall specify the number of Attribute records in this record set and shall be represented by a 16-bit unsigned integer.

   3) Attribute Records. The types of attributes and their values to be communicated shall be represented by one or more Attribute records (see 6.2.10).

The format of the Attribute PDU shall be as shown in Table 138.

Table 138—Attribute PDU

Field size (bits)	**Attribute PDU**	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 72
		Protocol Family—8-bit enumeration = 1
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
32	Originating Simulation Address	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
32	Padding	32 bits unused
16	Padding	16 bits unused
8	Attribute Record PDU Type	8-bit enumeration
8	Attribute Record Protocol Version	8-bit enumeration
32	Master Attribute Record Type	32-bit enumeration
8	Action Code	8-bit enumeration
8	Padding	8 bits unused
16	Number of Attribute Record Sets (S)	16-bit unsigned integer
Attribute Record Set # 1		
48	Entity/Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity/Object Number—16-bit unsigned integer
16	Number of Attribute Records (N_1)	16-bit unsigned integer
$48 + 8K_{11} + 8P_{11}$	Attribute Record # 11	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6 + K_{11} + P_{11}$)
		Record-Specific fields—K_{11} octets
		Padding to 64-bit boundary—P_{11} octets
• • •		

Table 138—Attribute PDU *(continued)*

Field size (bits)	Attribute PDU	
$48 + 8K_{1N_1} + 8P_{1N_1}$	Attribute Record #1N_1	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_{1N_1} + P_{1N_1})$
		Record-Specific fields—K_{1N_1} octets
		Padding to 64-bit boundary—P_{1N_1} octets
	• • •	
	Attribute Record Set # S	
48	Entity/Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity/Object Number—16-bit unsigned integer
16	Number of Attribute Records (N_S)	16-bit unsigned integer
$48 + 8K_{S1} + 8P_{S1}$	Attribute Record # $S1$	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_{S1} + P_{S1})$
		Record-Specific fields—K_{S1} octets
		Padding to 64-bit boundary—P_{S1} octets
	• • •	
$48 + 8K_{SN_S} + 8P_{SN_S}$	Attribute Record #$S\ N_S$	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_{SN_S} + P_{SN_S})$
		Record-Specific fields—K_{SN_S} octets
		Padding to 64-bit boundary—P_{SN_S} octets

Total Attribute PDU size = $256 + 8\sum_{i=1}^{S}\left[8 + \sum_{j=1}^{N_i}(6 + K_{ij} + P_{ij})\right]$ bits

where

S is the number of Record Sets
N_i is the number of Attribute records in Record Set i
K_{ij} is the length of the Record-Specific field in Attribute Record j (within Record Set i) in octets
P_{ij} is the number of padding octets in Attribute Record j (within Record Set i), which is $\lceil(6 + K_{ij})/8\rceil 8 - (6 + K_{ij})$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.3 Warfare protocol family

7.3.1 General

The PDUs of the Warfare protocol family are described in 7.3.2 through 7.3.5.

7.3.2 Fire PDU

The firing of a weapon or expendable shall be communicated by issuing a Fire PDU. See 5.4.3 for specific requirements on the use of the Fire PDU. The Fire PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Firing Entity ID*. This field shall identify the firing entity and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Target Entity ID*. This field shall identify the intended target (see 5.4.3.3). This field shall be represented by an Entity Identifier record (see 6.2.28).

d) *Munition/Expendable Entity ID*. This field shall specify the entity identification of the fired munition or expendable. This field shall be represented by an Entity Identifier record (see 6.2.28).

e) *Event ID*. This field shall contain an identification generated by the firing entity to associate related firing and detonation events. This field shall be represented by an Event Identifier record (see 6.2.33).

f) *Fire Mission Index*. This field shall identify the fire mission (see 5.4.3.3). This field shall be represented by a 32-bit unsigned integer.

g) *Location in World Coordinates*. This field shall specify the location, in world coordinates, from which the munition was launched and shall be represented by a World Coordinates record (see 6.2.98).

h) *Descriptor*. This field shall describe the firing or launch of a munition or expendable represented by one of the following types of Descriptor records: Munition Descriptor (6.2.19.2) or Expendable Descriptor (6.2.19.4).

i) *Velocity*. This field shall specify the velocity of the fired munition at the point when the issuing simulation application intends the externally visible effects of the launch (e.g., exhaust plume or muzzle blast) to first become apparent. The velocity shall be represented in world coordinates. This field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96].

j) *Range*. This field shall specify the range that an entity's fire control system has assumed in computing the fire control solution. This field shall be represented by a 32-bit floating point number in meters. For systems where range is unknown or unavailable, this field shall contain a value of zero.

The format of the Fire PDU shall be as shown in Table 139.

Table 139—Fire PDU

Field size (bits)	Fire PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 2
		Protocol Family—8-bit enumeration = 2
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Firing Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Target Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Munition/Expendable Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
32	Fire Mission Index	32-bit unsigned integer
192	Location in World Coordinates	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
128	Descriptor	Entity Type record—64 bits
		Descriptor record fields—64 bits
96	Velocity	*X*-component—32-bit floating point
		Y-component—32-bit floating point
		Z-component—32-bit floating point
32	Range	32-bit floating point
Total Fire PDU size = 768 bits		

7.3.3 Detonation PDU

The detonation or impact of munitions, as well as non-munition explosions, the burst or initial bloom of chaff, and the ignition of a flare shall be indicated using the Detonation PDU. See 5.4.4 for specific requirements on the use of the Detonation PDU. The Detonation PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Source Entity ID*. This field shall identify the entity that fired the munition, the entity that launched the expendable, or the entity that caused the non-munition entity, or portion thereof, to explode as specified in 5.4.4.3. This field shall be represented by an Entity Identifier record (see 6.2.28).

c) *Target Entity ID*. This field shall identify the target entity of the munition, the target threat entity to the entity that launched the expendable, or the non-munition entity that exploded as specified in 5.4.4.3. This field shall be represented by an Entity Identifier record (see 6.2.28).

d) *Exploding Entity ID*. This field shall identify the munition entity or the expendable entity. This field is not applicable for non-munition detonations. Detailed requirements for setting this field are in 5.4.4.3. This field shall be represented by an Entity Identifier record (see 6.2.28).

e) *Event ID*. This field shall contain the same data as in the Event ID field of the Fire PDU that communicated the launch of the munition or expendable. If the detonation is not preceded by a corresponding fire event, then the Event Number field of the Event Identifier record shall be zero (e.g., land mines detonation). This field shall be represented by an Event Identifier record (see 6.2.33).

f) *Velocity*. This field shall specify the velocity of the munition immediately before detonation/impact, the velocity of a non-munition entity immediately before exploding, or the velocity of an expendable immediately before a chaff burst or ignition of a flare. The velocity shall be represented in world coordinates. This field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96].

g) *Location in World Coordinates*. This field shall specify the location of the detonation in world coordinates. This field shall be represented by a World Coordinates record (see 6.2.98).

h) *Descriptor*. This field shall describe the detonation represented by one of the following types of Descriptor records: Munition Descriptor (6.2.19.2), Explosion Descriptor (6.2.19.3), or Expendable Descriptor (6.2.19.4).

i) *Location in Entity's Coordinates*. This field shall specify the location of the munition detonation, the expendable detonation, or the non-munition explosion as specified in sections 5.4.4.3. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

j) *Detonation Result*. This field shall specify the result of the detonation and shall be represented by an 8-bit enumeration (see [UID 62]).

k) *Number of Variable Parameter Records*. This field shall specify the number of Variable Parameter records present. This field shall be represented by an 8-bit unsigned integer (see Annex I).

l) *Variable Parameter records*. This field shall specify the parameter values for each Variable Parameter record that is included (see 6.2.94 and Annex I).

The format of the Detonation PDU shall be as shown in Table 140.

Table 140—Detonation PDU

Field size (bits)	Detonation PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 3
		Protocol Family—8-bit enumeration = 2
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Source Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Target Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Exploding Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
96	Velocity	*X*-component—32-bit floating point
		Y-component—32-bit floating point
		Z-component—32-bit floating point
192	Location in World Coordinates	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
128	Descriptor	Entity Type record—64 bits
		Descriptor record fields—64 bits
96	Location in Entity's Coordinates	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point

Table 140—Detonation PDU *(continued)*

Field size (bits)	Detonation PDU fields	
8	Detonation Result	8-bit enumeration
8	Number of Variable Parameter Records (*N*)	8-bit unsigned integer
16	Padding	16 bits unused
128	Variable Parameter record #1	Record Type—8-bit enumeration
		Record-Specific fields—120 bits
⋮		
128	Variable Parameter record #*N*	Record Type—8-bit enumeration
		Record-Specific fields—120 bits
Total Detonation PDU size = 832 + 128*N* bits where *N* is the number of Variable Parameter records		

7.3.4 Directed Energy (DE) Fire PDU

The firing of a directed energy weapon shall be communicated by issuing a DE Fire PDU. See 5.4.5 for specific requirements on the use of the DE Fire PDU. The DE Fire PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Firing Entity ID*. This field shall identify the firing entity and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Event ID*. This field shall contain a number generated by the issuing simulator to associate DE Fire PDUs related to the same shot. This field shall be represented by an Event Identifier record (see 6.2.33).

d) *Munition Type*. This field shall identify the munition type enumeration for the DE weapon beam. This field shall be represented by an Entity Type record (see 6.2.30).

e) *Shot Start Time*. This field shall indicate the simulation time at start of the shot. This field shall be represented by a Clock Time record (see 6.2.14).

f) *Cumulative Shot Time*. This field shall indicate the current cumulative duration of the shot, expressed in units of seconds, and shall be represented by a 32-bit floating point number.

g) *Aperture/Emitter Location in Firing Entity Coordinates*. This field shall identify the location of the DE weapon aperture/emitter and shall be represented by an Entity Coordinate Vector (see 6.2.96), expressed in units of meters.

h) *Aperture Diameter*. This field shall identify the beam diameter at the aperture/emitter, expressed in units of meters, and shall be represented by a 32-bit floating point number.

i) *Wavelength*. This field shall identify the emissions wavelength in units of meters and shall be represented by a 32-bit floating point number.

j) *Pulse Repetition Frequency*. This field shall identify the current pulse repetition frequency in units of cycles per second (Hertz) and shall be represented by a 32-bit floating point number.

k) *Pulse Width*. This field shall identify the pulse width emissions in units of seconds and shall be represented by a 32-bit floating point number.

l) *Flags*. This field shall contain various flags to indicate status information needed to process a DE Fire PDU. It shall be a 16-bit record (see [UID 313]).

m) *Pulse Shape*. This field shall identify the pulse shape and shall be represented as an 8-bit enumeration (see [UID 312]).

n) *Number of DE Records*. This field shall specify the number of DE records and shall be represented by a 16-bit unsigned integer.

o) *DE records*. These fields shall contain one or more DE records (see 6.2.20) that conform to the variable record format of a Standard Variable Specification record (see 6.2.83) and may contain other Standard Variable records.

The format of the DE Fire PDU shall be as shown in Table 141.

Table 141—DE Fire PDU

Field size (bits)	DE Fire PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 68
		Protocol Family—8-bit enumeration = 2
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Firing Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
64	Munition Type	Entity Type record—64 bits
64	Shot Start Time	Hour—32-bit integer
		Time Past the Hour—32-bit unsigned integer
32	Cumulative Shot Time	32-bit floating point
96	Aperture/Emitter Location in Firing Entity Coordinates	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
32	Aperture Diameter	32-bit floating point

Table 141—DE Fire PDU *(continued)*

Field size (bits)	DE Fire PDU fields	
32	Wavelength	32-bit floating point
32	Padding	32 bits unused
32	Pulse Repetition Frequency	32-bit floating point
32	Pulse Width	32-bit floating point
16	Flags	16-bit record
8	Pulse Shape	8-bit enumeration
8	Padding	8 bits unused
32	Padding	32 bits unused
16	Padding	16 bits unused
16	Number of DE Records (N)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	DE record # 1	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_1 + P_1)$
		Record-Specific fields—K_1 octets
		Padding to 64-bit boundary—P_1 octets
	• • •	
$48 + 8K_N + 8P_N$	DE record # N	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_N + P_N)$
		Record-Specific fields—K_N octets
		Padding to 64-bit boundary—P_N octets

Total DE Fire PDU size = $704 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

N is the number of DE records
K_i is the length of the Record-Specific field in DE record i in octets
P_i is the number of padding octets in DE record i, which is $\lceil (6 + K_i)/8 \rceil 8 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.3.5 Entity Damage Status PDU

The Entity Damage Status PDU shall be used to communicate detailed damage information sustained by an entity regardless of the source of the damage. See 5.4.6 for specific requirements on the use of the Entity Damage Status PDU. The Entity Damage Status PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Damaged Entity ID*. This field shall identify the damaged entity and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Number of Damage Description Records*. This field shall specify the number of Damage Description records and shall be represented by a 16-bit unsigned integer.

d) *Damage Description records*. These fields shall contain one or more Damage Description records (see 6.2.15) and may contain other Standard Variable records. These records shall conform to the Standard Variable record format of the Standard Variable Specification record (see 6.2.83).

The format of the Entity Damage Status PDU shall be as shown in Table 142.

Table 142—Entity Damage Status PDU

Field size (bits)	**Entity Damage Status PDU fields**	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 69
		Protocol Family—8-bit enumeration = 2
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Damaged Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Padding	16 bits unused
16	Padding	16 bits unused
16	Number of Damage Description Records (*N*)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	Damage Description record #1	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6 + K_1 + P_1$)
		Record-Specific fields—K_1 octets
		Padding to 64-bit boundary—P_1 octets
	⋮	

Table 142—Entity Damage Status PDU *(continued)*

Field size (bits)	Entity Damage Status PDU fields	
$48 + 8K_N + 8P_N$	Damage Description record #*N*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_N + P_N)$
		Record-Specific fields—K_N octets
		Padding to 64-bit boundary—P_N octets

Total Entity Damage Status Report PDU = $192 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

N is the number of Damage Description records
K_i is the length of the Record-Specific field in Damage Description record i in octets
P_i is the number of padding octets in Damage Description record i, which is $\lceil(6 + K_i)/8\rceil 8 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.4 Logistics protocol family

7.4.1 General

The PDUs of the Logistics protocol family are described in 7.4.2 through 7.4.7.

7.4.2 Service Request PDU

A request for logistics support shall be communicated by issuing a Service Request PDU. See 5.5.5 for specific requirements on the use of the Service Request PDU. The Service Request PDU shall consist of the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Requesting Entity ID.* This field shall identify the entity that is requesting the service and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Servicing Entity ID.* This field shall identify the entity that is able to provide the service requested and shall be represented by an Entity Identifier record (see 6.2.28).

d) *Service Type Requested.* This field shall describe the type of service being requested and shall be represented by an 8-bit enumeration (see [UID 63]).

e) *Number of Supply Types.* For a service of resupply, this field shall specify the number of different supplies being requested. If the service requested is not resupply, this field shall contain the value zero. This field shall be represented by an 8-bit unsigned integer.

f) *Supplies.* For a service of resupply, this field shall specify the type of supply and the amount of that supply for the number of supplies specified above. If the service requested is not resupply, this field shall not be present. This field shall be represented by a Supply Quantity record (see 6.2.86).

The format of the Service Request PDU shall be as shown in Table 143.

Table 143—Service Request PDU

Field size (bits)	**Service Request PDU fields**	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 5
		Protocol Family—8-bit enumeration = 3
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Requesting Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Servicing Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Service Type Requested	8-bit enumeration
8	Number of Supply Types (*N*)	8-bit unsigned integer
16	Padding	16 bits unused
96	Supplies #1	Supply Type—Entity Type record
		Quantity—32-bit floating point
⋮		
96	Supplies #*N*	Supply Type—Entity Type record
		Quantity—32-bit floating point
Total Service Request PDU size = 224 + 96*N* bits where *N* is the number of supply types		

7.4.3 Resupply Offer PDU

The offering of supplies shall be communicated by issuing a Resupply Offer PDU. See 5.5.6 for specific requirements on the use of the Resupply Offer PDU. The Resupply Offer PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Receiving Entity ID.* This field shall identify the receiving entity and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Supplying Entity ID.* This field shall identify the supplying entity and shall be represented by an Entity Identifier record (see 6.2.28).

d) *Number of Supply Types.* This field shall specify the number of different supply types being offered and shall be represented by an 8-bit unsigned integer.

e) *Supplies.* This field shall specify the type of supply and the amount of that supply for each of the supply types specified above. This field shall not be present if the Number of Supply Types field value is zero. This field shall be represented by a Supply Quantity record (see 6.2.86).

The format of the Resupply Offer PDU shall be as shown in Table 144.

Table 144—Resupply Offer PDU

Field size (bits)	Resupply Offer PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 6
		Protocol Family—8-bit enumeration = 3
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Receiving Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Supplying Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Number of Supply Types (*N*)	8-bit unsigned integer
8	Padding	8 bits unused
16	Padding	16 bits unused

Table 144—Resupply Offer PDU *(continued)*

Field size (bits)	Resupply Offer PDU fields	
96	Supplies #1	Supply Type—Entity Type record
		Quantity—32-bit floating point
	• • •	
96	Supplies #*N*	Supply Type—Entity Type record
		Quantity—32-bit floating point
Total Resupply Offer PDU size = 224 + 96*N* bits where *N* is the number of supply types		

7.4.4 Resupply Received PDU

The receipt of supplies shall be communicated by issuing a Resupply Received PDU. See 5.5.7 for specific requirements on the use of the Resupply Received PDU. The Resupply Received PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Receiving Entity ID*. This field shall identify the receiving entity and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Supplying Entity ID*. This field shall identify the supplying entity and shall be represented by an Entity Identifier record (see 6.2.28).

d) *Number of Supply Types*. This field shall specify the number of different supplies taken by the receiving entity and shall be represented by an 8-bit unsigned integer.

e) *Supplies*. This field shall specify the type of supply and the amount of that supply for each of the supply types specified above. This field shall not be present if the Number of Supply Types field value is zero. This field shall be represented by a Supply Quantity record (see 6.2.86).

The format of the Resupply Received PDU shall be as shown in Table 145.

Table 145—Resupply Received PDU

Field size (bits)	Resupply Received PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 7
		Protocol Family—8-bit enumeration = 3
		Time stamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Receiving Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Supplying Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Number of Supply Types (*N*)	8-bit unsigned integer
8	Padding	8 bits unused
16	Padding	16 bits unused
96	Supplies #1	Supply Type—Entity Type record
		Quantity—32-bit floating point
	• • •	
96	Supplies #*N*	Supply Type—Entity Type record
		Quantity—32-bit floating point
Total Resupply Received PDU size = 224 + 96*N* bits where *N* is the number of supply types		

7.4.5 Resupply Cancel PDU

The canceling of a service function by either the receiving or the supplying entity shall be communicated by issuing a Resupply Cancel PDU. See 5.5.8 for specific requirements on the use of the Resupply Cancel PDU. The Resupply Cancel PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Receiving Entity ID*. This field shall identify the entity that has requested the resupply service and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Supplying Entity ID*. This field shall identify the supplying entity and shall be represented by an Entity Identifier record (see 6.2.28).

The format of the Resupply Cancel PDU shall be as shown in Table 146.

Table 146—Resupply Cancel PDU

Field size (bits)	Resupply Cancel PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 8
		Protocol Family—8-bit enumeration = 3
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Receiving Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Supplying Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
Total Resupply Cancel PDU size = 192 bits		

7.4.6 Repair Complete PDU

When a Service Request PDU has been received and the repairing entity has completed a requested repair, the repairing entity shall notify the receiving entity of the repair by issuing a Repair Complete PDU. See 5.5.10 for specific requirements on the use of the Repair Complete PDU. The Repair Complete PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Receiving Entity ID*. This field shall identify the entity that is requesting repairs and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Repairing Entity ID*. This field shall identify the repairing entity and shall be represented by an Entity Identifier record (see 6.2.28).

d) *Repair*. This field shall describe the repair performed and shall be represented by a 16-bit enumeration repair type (see [UID 64]).

The format of the Repair Complete PDU shall be as shown in Table 147.

Table 147—Repair Complete PDU

Field size (bits)	Repair Complete PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU type—8-bit enumeration = 9
		Protocol Family—8-bit enumeration= 3
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Receiving Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Repairing Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Repair	16-bit enumeration
16	Padding	16 bits unused
Total Repair Complete PDU size = 224 bits		

7.4.7 Repair Response PDU

When a receiving entity receives a Repair Complete PDU from its repairing entity, the receiving entity shall acknowledge the receipt of the repair by issuing a Repair Response PDU. See 5.5.11 for specific requirements on the use of the Repair Response PDU. The Repair Response PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Receiving Entity ID.* This field shall identify the entity that requested repairs and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Repairing Entity ID.* This field shall identify the repairing entity and shall be represented by an Entity Identifier record (see 6.2.28).

d) *Repair Result.* This field shall specify the result of the repair specified in the Repair Complete PDU and shall be represented by an 8-bit enumeration (see [UID 65]).

The format of the Repair Response PDU shall be as shown in Table 148.

Table 148—Repair response PDU

Field size (bits)	Repair Response PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 10
		Protocol Family—8-bit enumeration = 3
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Receiving Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Repairing Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Repair Result	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
Total Repair Response PDU size = 224 bits		

7.5 Simulation Management protocol family

7.5.1 General

The PDUs of the Simulation Management protocol family are described in 7.5.2 through 7.5.13.

7.5.2 Create Entity PDU

The creation of a new entity shall be communicated using a Create Entity PDU. See 5.6.5.2 for specific requirements on the use of the Create Entity PDU. The Create Entity PDU shall contain the following fields:

a) *Simulation Management PDU Header*. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID*. This field shall identify the entity creation request being made by the SM and shall be represented by a 32-bit unsigned integer (see 6.2.75).

The format of the Create Entity PDU shall be as shown in Table 149.

Table 149—Create entity PDU

Field size (bits)	Create Entity PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 11
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
Total Create Entity PDU size = 224 bits		

7.5.3 Remove Entity PDU

The removal of an entity from an exercise shall be communicated with a Remove Entity PDU. See 5.6.5.3 for specific requirements on the use of the Remove Entity PDU. The Remove Entity PDU shall consist of the following fields:

a) *Simulation Management PDU Header*. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the specific and unique entity removal request being made by the SM. This field shall be represented by a 32-bit unsigned integer.

The format of the Remove Entity PDU shall be as shown in Table 150.

Table 150—Remove Entity PDU

Field size (bits)	Remove Entity PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 12
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
Total Remove Entity PDU size = 224 bits		

7.5.4 Start/Resume PDU

The Start/Resume of an entity/exercise shall be communicated using a Start/Resume PDU. See 5.6.5.4 for specific requirements on the use of the Start/Resume PDU. The Start/Resume PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Real-World Time.* This field shall specify the real-world time at which the entity is to start/resume in the exercise. This information shall be used by the participating simulation applications to start/resume an exercise synchronously. This field shall be represented by a Clock Time record (see 6.2.14).

c) *Simulation Time.* The shared time being simulated within a simulation exercise. This time is established ahead of time by simulation management and is common to all participants in a particular exercise. Simulation time may be either Absolute Time or Relative Time. This field shall be represented by a Clock Time record (see 6.2.14).

d) *Request ID.* This field shall identify the specific and unique start/resume request being made by the SM and shall be represented by a 32-bit unsigned integer.

The format of the Start/Resume PDU shall be as shown in Table 151.

Table 151—Start/Resume PDU

Field size (bits)	Start/Resume PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 13
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
64	Real-World Time	Hour—32-bit integer
		Time Past Hour—32-bit unsigned integer
64	Simulation Time	Hour—32-bit integer
		Time Past Hour—32-bit unsigned integer
32	Request ID	32-bit unsigned integer
Total Start/Resume PDU size = 352 bits		

7.5.5 Stop/Freeze PDU

The stopping or freezing of an entity/exercise shall be communicated using a Stop/Freeze PDU. See 5.6.5.5 for specific requirements on the use of the Stop/Freeze PDU. The Stop/Freeze PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Real-World Time.* This field shall specify the real-world time at which the entity is to stop/freeze in the exercise and shall be represented by a Clock Time record (see 6.2.14).

c) *Reason.* This field shall specify the reason that an entity or exercise was stopped/frozen and shall be represented by an 8-bit enumeration (see [UID 67]).

d) *Frozen Behavior.* This field shall specify the internal behavior of the simulation and its appearance while frozen to the other participants of the exercise and shall be represented by an 8-bit record (see [UID 68]).

e) *Request ID.* This field shall identify the specific and unique stop/freeze request being made by the SM and shall be represented by a 32-bit unsigned integer.

The format of the Stop/Freeze PDU shall be as shown in Table 152.

Table 152—Stop/Freeze PDU

Field size (bits)	Stop/Freeze PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 14
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
64	Real-World Time	Hour—32-bit integer
		Time Past Hour—32-bit unsigned integer
8	Reason	8-bit enumeration
8	Frozen Behavior	8-bit record
16	Padding	16 bits unused
32	Request ID	32-bit unsigned integer
Total Stop/Freeze PDU size = 320 bits		

7.5.6 Acknowledge PDU

The acknowledgment of the receipt of a Start/Resume PDU, Stop/Freeze PDU, Create Entity PDU, or Remove Entity PDU shall be communicated by issuing an Acknowledge PDU. See 5.6.5.6 for specific requirements on the use of the Acknowledge PDU. The Acknowledge PDU shall contain the following fields:

a) *Simulation Management PDU Header*. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Acknowledge Flag.* This field shall indicate what type of message has been acknowledged and shall be represented by a 16-bit enumeration (see [UID 69]).

c) *Response Flag.* This field shall indicate whether or not the receiving entity was able to comply with the request and shall be represented by a 16-bit enumeration (see [UID 70]).

d) *Request ID.* This field shall identify the matching response to the specific Start/Resume, Stop/Freeze, Create Entity, or Remove Entity PDU sent by the SM. This field shall be represented by a 32-bit unsigned integer.

The format of the Acknowledge PDU shall be as shown in Table 153.

Table 153—Acknowledge PDU

Field size (bits)	Acknowledge PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 15
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Acknowledge Flag	16-bit enumeration
16	Response Flag	16-bit enumeration
32	Request ID	32-bit unsigned integer
Total Acknowledge PDU size = 256 bits		

7.5.7 Action Request PDU

A request from an SM to a managed entity to perform a specified action shall be communicated using an Action Request PDU. See 5.6.5.7 for specific requirements on the use of the Action Request PDU. The Action Request PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the request being made by the SM and shall be represented by a 32-bit unsigned integer.

c) *Action ID.* This field shall specify the particular action that is requested by the SM and shall be represented by a 32-bit enumeration (see [UID 71]).

d) *Datum Information.* This field shall specify the types of datum and the value of the datum to be communicated and shall be represented by a Datum Specification record (see 6.2.18).

The format of the Action Request PDU shall be as shown in Table 154.

Table 154—Action Request PDU

Field size (bits)	Action Request PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 16
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
32	Action ID	32-bit enumeration
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer

Table 154—Action Request PDU *(continued)*

Field size (bits)	Action Request PDU fields	
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum #*N*	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
	• • •	
$64 + K_M + P_M$	Variable Datum #*M*	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Action Request PDU size = $320 + 64N + \sum_{i=1}^{M}(64 + K_i + P_i)$ bits

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i/64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.5.8 Action Response PDU

When an entity receives an Action Request PDU, that entity shall acknowledge the receipt of the Action Request PDU with an Action Response PDU. See 5.6.5.8 for specific requirements on the use of the Action Response PDU. The Action Response PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the matching response to a request made by the SM and shall be represented by a 32-bit unsigned integer.

c) *Request Status.* This field shall identify the status of the requested action and shall be represented by a 32-bit enumeration (see [UID 72]).

d) *Datum Information.* This field shall specify the types of datum and the value of the datum to be communicated and shall be represented by a Datum Specification record (see 6.2.18).

The format of the Action Response PDU shall be as shown in Table 155.

Table 155—Action Response PDU

Field size (bits)	Action Response PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 17
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
32	Request Status	32-bit enumeration
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum #*N*	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits

Table 155—Action Response PDU *(continued)*

Field size (bits)	Action Response PDU fields	
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
	• • •	
$64 + K_M + P_M$	Variable Datum #*M*	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits
Total Action Response PDU size = $320 + 64N + \sum_{i=1}^{M}(64 + K_i + P_i)$ bits where N is the number of fixed datum records M is the number of variable datum records K_i is the length of variable datum value i in bits P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

7.5.9 Data Query PDU

A request for data from an entity shall be communicated by issuing a Data Query PDU. See 5.6.5.9 for specific requirements on the use of the Data Query PDU. The Data Query PDU shall contain the following fields:

a) *Simulation Management PDU Header*. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID*. This field shall identify the data query request being made by the SM and shall be represented by a 32-bit unsigned integer.

c) *Time Interval*. This field shall specify the reference time interval between issues of Data PDUs (see 5.6.5.9.3). This field shall be represented by a timestamp (see 6.2.88).

d) *Datum Information*. This field shall specify the types of datum for which information is required. Datum IDs, but not datum values, are listed in the Data Query Datum Specification record (see 6.2.17).

The format of the Data Query PDU shall be as shown in Table 156.

Table 156—Data Query PDU

Field size (bits)	Data Query PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 18
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
32	Time Interval	32-bit unsigned integer
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
32	Fixed Datum ID #1	32-bit enumeration
	• • •	
32	Fixed Datum ID #*N*	32-bit enumeration
32	Variable Datum ID #1	32-bit enumeration

Table 156—Data Query PDU *(continued)*

Field size (bits)	Data Query PDU fields	
⋮		
32	Variable Datum ID #*M*	32-bit enumeration
Total Data Query PDU size = $320 + 32N + 32M$ bits where N is the number of fixed datum records M is the number of variable datum records		

7.5.10 Set Data PDU

Initializing or changing internal state information shall be communicated using a Set Data PDU. See 5.6.5.10 for specific requirements on the use of the Set Data PDU. The Set Data PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the set data request being made by the SM and shall be represented by a 32-bit unsigned integer.

c) *Datum Information.* This field shall specify the types of datum and their value to be communicated and shall be represented by a Datum Specification record (see 6.2.18).

The format of the Set Data PDU shall be as shown in Table 157.

Table 157—Set Data PDU

Field size (bits)	Set Data PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 19
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused

Table 157—Set Data PDU (continued)

Field size (bits)	Set Data PDU fields	
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
32	Padding	32 bits unused
32	Number of Fixed Datum Records (N)	32-bit unsigned integer
32	Number of Variable Datum Records (M)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
• • •		
64	Fixed Datum #N	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
• • •		

Table 157—Set Data PDU *(continued)*

Field size (bits)	Set Data PDU fields	
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Set Data PDU size = $320 + 64N + \sum_{i=1}^{M} (64 + K_i + P_i)$ bits

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.5.11 Data PDU

Information issued in response to a Data Query PDU or Set Data PDU shall be communicated using a Data PDU. See 5.6.5.11 for specific requirements on the use of the Data PDU. The Data PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the matching response to a Data Query PDU or Set Data PDU made by the SM and shall be represented by a 32-bit unsigned integer.

c) *Datum Information.* This field shall specify the types of datum and their value to be communicated and shall be represented by a Datum Specification record (see 6.2.18).

The format of the Data PDU shall be as shown in Table 158.

Table 158—Data PDU

Field size (bits)	Data PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 20
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
32	Padding	32 bits unused
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum #*N*	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits

Table 158—Data PDU *(continued)*

Field size (bits)	Data PDU fields	
⋮		
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Data PDU size $= 320 + 64N + \sum_{i=1}^{M} (64 + K_i + P_i)$ bits

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.5.12 Event Report PDU

A managed entity shall report the occurrence of a significant event to the SM using an Event Report PDU. See 5.6.5.12 for specific requirements on the use of the Event Report PDU. The Event Report PDU shall consist of the following fields:

a) *Simulation Management PDU Header*. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Event Type*. This field shall specify the type of event that caused the issue of an Event PDU and shall be represented by a 32-bit enumeration (see [UID 73]).

c) *Datum Information.* This field shall specify the types of datum and their value to be communicated and shall be represented by a Datum Specification record (see 6.2.18).

The format of the Event Report PDU shall be as shown in Table 159.

Table 159—Event Report PDU

Field size (bits)	Event Report PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 21
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Event Type	32-bit enumeration
32	Padding	32 bits unused
32	Number of Fixed Datum Records (N)	32-bit unsigned integer
32	Number of Variable Datum Records (M)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum #N	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits

Table 159—Event Report PDU *(continued)*

Field size (bits)	Event Report PDU fields	
⋮		
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Event Report PDU size = $320 + 64N + \sum_{i=1}^{M} (64 + K_i + P_i)$ bits

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.5.13 Comment PDU

Arbitrary messages (character strings, for example) shall be entered into the data stream by using a Comment PDU. See 5.6.5.13 for specific requirements on the use of the Comment PDU. The Comment PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Number of Fixed Datum Records*. This field shall specify the number of Fixed Datum records in the Comment PDU and shall be represented by a 32-bit unsigned integer. The value of this field shall be set to zero.

c) *Number of Variable Datum Records.* This field shall specify the number of Variable Datum records required to supply database names or character fields that exceed 32 bits and shall be represented by a 32-bit unsigned integer.

d) *Variable Datum records.* These fields shall specify the types of variable datum, their length, and their value and shall be represented by a Variable Datum record (see 6.2.93).

The format of the Comment PDU shall be as shown in Table 160.

Table 160—Comment PDU

Field size (bits)	Comment PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 22
		Protocol Family—8-bit enumeration = 5
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Number of Fixed Datum Records	32-bit unsigned integer
32	Number of Variable Datum Records (M)	32-bit unsigned integer
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
• • •		

Table 160—Comment PDU *(continued)*

Field size (bits)	Comment PDU fields	
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

$$\text{Total Comment PDU size} = 256 + \sum_{i=1}^{M} (64 + K_i + P_i) \text{ bits}$$

where

M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.6 Distributed Emission Regeneration protocol family

7.6.1 General

The PDUs of the Distributed Emission Regeneration protocol family are described in 7.6.2 through 7.6.6.

7.6.2 Electromagnetic Emission (EE) PDU

Information about active electromagnetic emissions shall be communicated using an EE PDU. See 5.7.3 for specific requirements on the use of the EE PDU. The EE PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Emitting Entity ID*. This field shall identify the entity that is the source of the emissions and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Event ID*. This field shall contain an identification generated by the issuing simulation application to associate related events and shall be represented by an Event Identifier record (see 6.2.33).

d) *State Update Indicator*. Use of this field is optional (see 5.7.3.3.3). It shall be represented by an 8-bit enumeration (see [UID 77]).

e) *Number of Systems*. This field shall specify the number of emitter systems described in the current PDU. This field shall be represented by an 8-bit unsigned integer.

f) The following information shall be provided for each emitter system in the PDU:

    1) System Data Length. If the length of this emitter system's data (including the System Data Length field and beam and track/jam information) does not exceed 1020 octets (255, 32-bit words), then this field shall specify the length of this emitter system's data in 32-bit words. Otherwise, the size of the emitter system is considered large and this field shall be set to zero. This field shall be represented by an 8-bit unsigned integer.

2) Number of Beams. This field shall specify the number of beams being described in the current PDU for the emitter system being described. This field shall be represented by an 8-bit unsigned integer.

3) Emitter System. This field shall specify information about a particular emitter system and shall be represented by an Emitter System record (see 6.2.23).

4) Location. This field shall specify the location of the antenna beam source with respect to the emitting entity's coordinate system. This location shall be the origin of the emitter coordinate system that shall have the same orientation as the entity coordinate system. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

5) The following information shall be provided for each active beam:

   i) Beam Data Length. If the length of this beam's data (including the Beam Data Length field and track/jam information) does not exceed 1020 octets (255, 32-bit words), then this field shall specify the length of this beam's data in 32-bit words. Otherwise, the size of the beam is considered large and this field shall be set to zero. This field shall be represented by an 8-bit unsigned integer.

   ii) Beam Number. This field shall specify a unique number assigned to differentiate between otherwise similar or identical emitter beams within an emitter system. This field shall be represented by an 8-bit unsigned integer.

   iii) Beam Parameter Index. Used in conjunction with the Emitter Name field as a database primary key, this field shall specify a number by which receiving entities reference stored database parameters required to regenerate the beam. The mechanism by which beam parameter index values are assigned is outside the scope of this standard. This field shall be represented by a 16-bit unsigned integer.

   iv) Fundamental Parameter Data. This field shall specify dynamic parameters of the emitter and shall be represented by an EE Fundamental Parameter Data record (see 6.2.22).

   v) Beam Data. This field shall specify parameters of the beam and shall be represented by a Beam Data record (see 6.2.11)

   vi) Beam Function. This field shall specify the intended use of a particular beam. Typical functions include search, acquisition, tracking, illumination, jamming, and so on. This field is intended to help receiving entities determine the emission mode represented by the beam. This field shall be represented by an 8-bit enumeration (see [UID 78]).

   vii) Number of Targets. This field, in conjunction with the High-Density Track/Jam field, shall identify, for the current PDU and emitter beam, the number of entities tracked or under illumination (as appropriate for an emitter beam's function) or the number of targeted emitter beams (for jammers). This field shall be represented by an 8-bit unsigned integer.

   viii) High-Density Track/Jam. This field shall be used to indicate that receiving simulation applications can assume that all viable targets in the field of regard specified by the beam data are being tracked or jammed. This field shall be represented by an 8-bit enumeration (see [UID 79]).

   ix) Beam Status. This field shall indicate the status of the beam (e.g., the beam is active or deactivated) and shall be represented by the Beam Status record (see 6.2.12).

   x) Jamming Technique. This field shall be used to identify the jamming method or methods and shall be represented by a Jamming Technique record (see 6.2.49).

   xi) Track/Jam Data. This field is optional for any given beam. Rules for inclusion and use are provided in 5.7.3.3, 5.7.3.7, and 5.7.3.8. When included, this field shall be represented by a series of Track/Jam Data records (see 6.2.90).

The EE PDU shall contain emitter system, beam, and track/jam data in an order that includes all beam and track/jam data for a particular emitter system before any data from a subsequent emitter system.

The format of the EE PDU shall be as shown in Figure 49 and Table 161.

Figure 49—General form of emitter systems in the EE PDU

<table>
<tr><td rowspan="10">Emitter System #1</td><td colspan="2">Emitter System #1 specific fields</td></tr>
<tr><td rowspan="4">Beam #1</td><td>Beam #1 specific fields</td></tr>
<tr><td>Track/Jam Data #1</td></tr>
<tr><td>•
•
•</td></tr>
<tr><td>Track/Jam Data #P_{11}</td></tr>
<tr><td colspan="2">•
•
•</td></tr>
<tr><td rowspan="4">Beam #M_1</td><td>Beam #M_1 specific fields</td></tr>
<tr><td>Track/Jam Data #1</td></tr>
<tr><td>•
•
•</td></tr>
<tr><td>Track/Jam Data #P_{1M}</td></tr>
<tr><td colspan="3">•
•
•</td></tr>
<tr><td rowspan="10">Emitter System #N</td><td colspan="2">Emitter System #N specific fields</td></tr>
<tr><td rowspan="4">Beam #1</td><td>Beam #1 specific fields</td></tr>
<tr><td>Track/Jam Data #1</td></tr>
<tr><td>•
•
•</td></tr>
<tr><td>Track/Jam Data #P_{N1}</td></tr>
<tr><td colspan="2">•
•
•</td></tr>
<tr><td rowspan="4">Beam #M_N</td><td>Beam #M_N specific fields</td></tr>
<tr><td>Track/Jam Data #1</td></tr>
<tr><td>•
•
•</td></tr>
<tr><td>Track/Jam Data #P_{NM}</td></tr>
</table>

NOTE—In Figure 49, the Track/Jam Data labeled "Track/Jam Data #P_{NM}" is read as the *P*th Track/Jam Data of the *M*th Beam of the *N*th Emitter System.

Table 161—EE PDU

Field size (bits)	**EE PDU fields**	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 23
		Protocol Family—8-bit enumeration = 6
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Emitting Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
8	State Update Indicator	8-bit enumeration
8	Number of Systems (*N*)	8-bit unsigned integer
16	Padding	16 bits unused
Emitter system data (*N* emitter systems, where *i* = 1 to *N*)		

Table 161—EE PDU *(continued)*

Field size (bits)	EE PDU fields	
Varies	System Data Length	8-bit unsigned integer
	Number of Beams (M_i)	8-bit unsigned integer
	Padding	16 bits unused
	Emitter System	Emitter name—16-bit enumeration
		Emitter Function—8-bit enumeration
		Emitter Number—8-bit unsigned integer
	Location (with respect to entity)	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
	Beam data for each emitter system (M_i beams, where j = 1 to M_i)	
	Beam Data Length	8-bit unsigned integer
	Beam Number	8-bit unsigned integer
	Beam Parameter Index	16-bit unsigned integer
	Fundamental Parameter Data	Frequency—32-bit floating point
		Frequency Range—32-bit floating point
		Effective Radiated Power—32-bit floating point
		Pulse Repetition Frequency—32-bit floating point
		Pulse width—32-bit floating point
	Beam Data	Beam Azimuth Center—32-bit floating point
		Beam Azimuth Sweep—32-bit floating point
		Beam Elevation Center—32-bit floating point
		Beam Elevation Sweep—32-bit floating point
		Beam Sweep Sync—32-bit floating point
	Beam Function	8-bit enumeration
	Number of Targets (P_{ij})	8-bit unsigned integer
	High-Density Track/Jam	8-bit enumeration
	Beam Status	8-bit record
	Jamming Technique	Kind—8-bit enumeration
		Category—8-bit enumeration
		Sub-category—8-bit enumeration
		Specific—8-bit enumeration

Table 161—EE PDU *(continued)*

Field size (bits)	EE PDU fields	
	Track/Jam Data for each beam (P_{ij} targets)	
	Track/Jam Data	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
		Emitter Number—8-bit unsigned integer
		Beam Number—8-bit unsigned integer
Total EE PDU size = $224 + 160N + \sum_{i=1}^{N}\left(416M_i + \sum_{j=1}^{M_i} 64P_{ij}\right)$ bits where N is the number of systems M_i is the number of beams of system i P_{ij} is the number of targets in Track/Jam Data records in beam j of system i		

7.6.3 Designator PDU

Designating operations shall be communicated by issuing a Designator PDU. See 5.7.4 for specific requirements on the use of the Designator PDU. The Designator PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Designating Entity ID.* This field shall identify the entity that is positioning the designator and shall be represented by an Entity Identifier record (see 6.2.28).

c) *Code Name.* This field shall identify the code name for the designator system and shall be represented by a 16-bit enumeration (see [UID 80]).

d) *Designated Entity ID.* This field shall identify the entity that is currently being designated (see 5.7.4.3). This field shall be represented by an Entity Identifier record (see 6.2.28).

e) *Designator Code.* This field shall identify the designator code being used by the designating entity and shall be represented by a 16-bit enumeration (see [UID 81]).

f) *Designator Power.* This field shall identify the designator output power in Watts and shall be represented by a 32-bit floating point number.

g) *Designator Wavelength.* This field shall identify the designator wavelength in units of microns and shall be represented by a 32-bit floating point number.

h) *Designator Spot with Respect to Designated Entity.* This field shall specify the location of the designator spot with respect to the designated entity's coordinate system (see 5.7.4.3). This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

i) *Designator Spot Location.* This field shall identify the location of the designator spot with respect to the world coordinate system and shall be represented by a World Coordinates record (see 6.2.98).

j) *Dead Reckoning Parameters*. This field will be used to provide parameters for dead reckoning the position of the designator spot. Dead reckoning algorithm in use and entity acceleration shall be included as part of the dead reckoning parameters:

1) Dead Reckoning Algorithm. This field shall specify the dead reckoning algorithm in use by the issuing entity. This field shall be represented by an 8-bit enumeration (see [UID 44]).

2) Entity Linear Acceleration. This field shall specify the designator spot's linear acceleration and shall be represented by a Linear Acceleration Vector record [see item b) in 6.2.96].

The format of the Designator PDU shall be as shown in Table 162.

Table 162—Designator PDU

Field size (bits)	Designator PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 24
		Protocol Family—8-bit enumeration = 6
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Designating Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Code Name	16-bit enumeration
48	Designated Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Designator Code	16-bit enumeration
32	Designator Power	32-bit floating point
32	Designator Wavelength	32-bit floating point
96	Designator Spot with Respect to Designated Entity	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
192	Designator Spot Location	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
8	Dead Reckoning Algorithm	8-bit enumeration

Table 162—Designator PDU *(continued)*

Field size (bits)	Designator PDU fields	
8	Padding	8 bits unused
16	Padding	16 bits unused
96	Entity Linear Acceleration	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
Total Designator PDU size = 704 bits		

7.6.4 Underwater Acoustic (UA) PDU

Information about acoustic emissions shall be communicated using a UA PDU. See 5.7.5 for specific requirements on the use of the UA PDU. The UA PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Emitting Entity ID.* This field shall identify the entity that is the source of UA emission. This field shall be represented by an Entity Identifier record (see 6.2.28).

c) *Event ID.* This field shall contain a number generated by the issuing simulator to associate related events. This field shall be represented by an Event Identifier record (see 6.2.33).

d) *State/Change Update Indicator.* This field shall be used to indicate whether the data in the UA PDU represent a state update or data that have changed since issuance of the last UA PDU. This field shall be represented by an 8-bit enumeration (see [UID 143]).

e) *Passive Parameter Index.* This field indicates which database record (or file) shall be used in the definition of passive signature (unintentional) emissions of the entity. The indicated database record (or file) shall define all noise generated as a function of propulsion plant configurations and associated auxiliaries. For example, parameter indexes identifying a file in the Common Acoustic Database (CADB) would be used here. This field shall be represented by a 16-bit enumeration (see [UID 148]).

f) *Propulsion Plant Configuration.* This field shall specify the entity propulsion plant configuration. This field is used to determine the passive signature characteristics of an entity. This field shall be represented by an 8-bit record (see [UID 149]).

g) *Number of Shafts.* This field shall represent the number of shafts on a platform and shall be represented by an 8-bit unsigned integer.

h) *Number of APAs.* This field shall indicate the number of APAs described in the current UA PDU. This field shall be an 8-bit unsigned integer.

i) *Number of UA Emitter Systems.* This field shall specify the number of UA emitter systems being described in the current UA PDU. One, several, or all of the UA emitter systems on a particular entity may be described in a single UA PDU. This field shall be represented by an 8-bit unsigned integer. Item o) through item r) shall be provided for each system.

j) *Current Shaft RPM.* This field shall indicate the current shaft speed in RPM for each shaft. The sign of the field shall indicate the direction of rotation. A positive RPM shall indicate the shaft is rotating in a clockwise direction (reference view is from the stern to bow), and a negative RPM shall indicate the shaft is rotating in a counter-clockwise direction. Shafts are defined from port to starboard locations (reference view is from stern to bow). This field shall be represented by a 16-bit signed integer.

k) *Ordered Shaft RPM.* This field shall indicate the ordered shaft speed in RPM for each shaft. The sign of the field shall indicate the direction of rotation. A positive RPM shall indicate the shaft is rotating in a clockwise direction (reference view is from the stern to bow), and a negative RPM shall indicate the shaft is rotating in a counter-clockwise direction. Shafts are defined from port to starboard locations (reference view is from stern to bow). This field shall be represented by a 16-bit signed integer.

l) *Shaft RPM Rate of Change.* This field shall indicate the shaft RPM rate of change for each shaft. Shafts are identified from port to starboard. This field shall be represented by a 32-bit signed integer so that negative rates of change (i.e., the shaft is slowing down) can be represented.

m) *APA Parameter Index.* This field indicates which database record (or file) shall be used to describe an additional acoustic source such as transient effects, prelaunch data (i.e., torpedo tube floodings), additional sources, and the states available for each source type. This field shall be represented by a 16-bit APA Parameter Index record as described in Table 163:

   1) APA Parameter. This field shall be used to identify data from databases such as the Special Event Database (SEDB) and an Additional Narrowband Database (ANDB). This field shall be represented by a 14-bit enumeration (see [UID 150]).

   2) APA Status. This field shall be used to indicate on/off/change status of the record (see 5.7.5.3). This field shall be represented by a 2-bit enumeration (see [UID 281]).

Table 163—APA Parameter Index record

Field name	Bits	Data type
APA Parameter	0–13	14-bit enumeration
APA Status	14–15	2-bit enumeration
Total APA Parameter Index record size = 16 bits		

n) *APA Value.* This 16-bit signed integer field shall represent the value of a state change defined in the APA Parameter Index field. The values shall be in units corresponding to the state change (such as dB and RPM).

o) *Emitter System Data Length.* This field shall specify the length in 32-bit words of the data for each emitter system. The length calculation includes the Emitter System Data Length field and all subsequent fields up to and including the Fundamental Data Parameters field of the last beam in the system. This field shall be represented by an 8-bit unsigned integer.

p) *Number of Beams.* This field shall specify the number of main beams being described in the current UA PDU for the system being described. This field shall be represented by an 8-bit unsigned integer.

q) *Acoustic Emitter System.* This field shall specify information about a particular emitter's system. This field shall be represented by an Acoustic Emitter System record (see 6.2.2).

r) *Location.* This field shall specify the location of the acoustic source in the emitting entity's coordinate system. This location shall be the origin of the UA emission coordinate system, which has the same orientation as the entity coordinate system. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

s) The following information shall be provided for each beam of each system:

   1) Beam Data Length. This field shall specify the length in 32-bit words including the Beam Data Length field of the data for all the active emission beams for each system. This field shall be represented by an 8-bit unsigned integer.

2) Beam ID Number. This field shall specify a unique number assigned to differentiate between multiple emitter beams within a UA emitter system. This field shall be represented by an 8-bit unsigned integer.
3) Fundamental Data Parameters. This field shall specify dynamic parameters of the UA emitter. This field shall be represented by a UA Fundamental Parameter Data record (see 6.2.91).

The format of the UA PDU shall be as shown in Figure 50 and Table 164.

Figure 50—General form of emitter systems in the UA PDU

Emitter System #1	Emitter System #1 specific fields
	Beam #1
	• • •
	Beam #M_1
• • •	
Emitter System #N	Emitter System #N specific fields
	Beam #1
	• • •
	Beam #M_N

NOTE—In Figure 50, the beam data labeled "Beam #M_N" is read as the Mth beam of the Nth emitter system.

Table 164—UA PDU

Field size (bits)	UA PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 29
		Protocol Family—8-bit enumeration = 6
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused

Table 164—UA PDU *(continued)*

Field size (bits)	UA PDU fields	
48	Emitting Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
8	State/Change Update Indicator	8-bit enumeration
8	Padding	8 bits unused
16	Passive Parameter Index	16-bit enumeration
8	Propulsion Plant Configuration	8-bit record
8	Number of Shafts (*S*)	8-bit unsigned integer
8	Number of APAs (*P*)	8-bit unsigned integer
8	Number of UA Emitter Systems (*N*)	8-bit unsigned integer
64	Current Shaft RPM #1	16-bit signed integer
	Ordered Shaft RPM #1	16-bit signed integer
	Shaft RPM Rate of Change #1	32-bit signed integer
	• • •	
64	Current Shaft RPM #S	16-bit signed integer
	Ordered Shaft RPM #S	16-bit signed integer
	Shaft RPM Rate of Change #S	32-bit signed integer
32	APA Parameter Index #1	16-bit record
	APA Value #1	16-bit signed integer
	• • •	
32	APA Parameter Index #*P*	16-bit record
	APA Value #*P*	16-bit signed integer
Emitter system data (*N* emitter systems, where *i* = 1 to *N*)		

Table 164—UA PDU *(continued)*

Field size (bits)	UA PDU fields	
160N	Emitter System Data Length	8-bit unsigned integer
	Number of Beams (M_i)	8-bit unsigned integer
	Padding	16 bits unused
	Acoustic Emitter System	Acoustic System Name—16-bit enumeration
		Function—8-bit enumeration
		Acoustic ID Number—8-bit unsigned integer
	Location (with respect to entity)	x-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
Beam data for each emitter system (M_i beams)		
192M for each system 1–N	Beam Data Length	8-bit unsigned integer
	Beam ID Number	8-bit unsigned integer
	Padding	16 bits unused
	Fundamental Data Parameters	Active Emission Parameter Index—16-bit enumeration
		Scan Pattern—16-bit enumeration
		Beam Center Azimuth (Horizontal Bearing) —32-bit floating point
		Azimuthal Beamwidth (Horizontal Beamwidth)—32-bit floating point
		Beam Center D/E—32-bit floating point
		D/E Beamwidth (Vertical Beamwidth)—32-bit floating point

Total UA PDU size = $256 + 64S + 32P + 160N + \sum_{i=1}^{N} 192M_i$ bits

where

S is the number of shafts
P is the number of APAs
N is the number of emitter systems
M_i is the number of beams in system i

7.6.5 Identification Friend or Foe (IFF) PDU

7.6.5.1 General

Information about military and civilian interrogators, transponders, and specific other electronic systems shall be communicated using the IFF PDU. See 5.7.6 for specific requirements on the use of the IFF PDU.

The term "IFF" when used in this standard shall encompass both military IFF and other transponder and interrogator systems, such as the civilian Air Traffic Control Radar Beacon System (ATCRBS). Table 14 lists applicable systems. IFF records applicable to all system types are contained in 6.2. IFF records, information requirements, and issuance and receipt rules that are applicable to one or more, but not all, system types are contained in Annex B:

a) The IFF PDU shall consist of a maximum of seven possible Information Layers. An Information Layer (layer) is defined as a set of functional information either common to all interrogators and transponders or associated with specific system types. Layers shall be included in numeric order beginning with Layer 1. Duplicate layer numbers within an IFF PDU shall not be allowed.

b) Layer 1 shall always be included in any IFF PDU that is issued. The inclusion of other layers is dependent on the type of IFF data that needs to be conveyed. All layers applicable to an initial, update, or heartbeat issuance of a PDU shall be contained in a single IFF PDU. The format and detailed field requirements associated with each layer are defined in this subclause except that some fields may contain different data values depending on the system type. The detailed use of such fields is contained in Annex B for those system types that use that field.

c) Table 165 describes the overall structure of the IFF PDU with all layers depicted. Except for Layer 1, which is required to be included in all IFF PDUs, the other layers present will depend on the system type, IFF Simulation Mode, and a simulation's fidelity requirements. Only the layers that are required are present in an IFF PDU. Clause B.5 specifies the required and optional layer(s) for each system type. The overarching goal of the layer concept is to efficiently transmit all the data needed to be exchanged for a system type without having to send multiple IFF PDUs. Subsequent paragraphs in this subclause define the layer formats associated with each system type for an interrogator and a transponder.

d) The term "format," as used here, denotes the format for an entire layer, including all fields and records contained therein. A specific format is designed to provide detailed information in support of one or more system types. A separate format may be implemented to support transponders and another to support the interrogators for a system type (e.g., both a Mode S Transponder format and a Mode S Interrogator format use Layer 4, one in support of Mode S transponders and the other in support of Mode S interrogators). Layers 1, 2, and 5 each have a standard format that is designed to support any system type. These formats include some system-specific fields and records that may be used. Layers 3, 4, 6, and 7 are available for formats to support present and future system types. Table 166 lists the formats currently implemented.

   NOTE—The implementation of IFF interoperability requirements cannot be achieved solely by reference to the informative and normative information contained in this standard. Knowledge of the system type being implemented can be obtained by reference to system specifications, standards, user documentation, and through appropriate consultation with personnel who are technically or operationally familiar with that system.

Table 165—Overall structure of the IFF PDU

Field size (bits)	Layer	IFF data records	Description
480	Layer 1	—	Layer 1 is required for all system types
varies	Layer 2	—	Optional layer
varies	Layer 3	X	Layer 3 is required if this system type has a Mode 5 capability
varies	Layer 4	X	Layer 4 is required if this system type has a Mode S capability
varies	Layer 5	X	Optional layer
—	Layer 6, 7	N/A	Not defined. Reserved for future use
NOTE—X—Includes an IFF Data record section. NA—Not Applicable.			

e) Some system types may represent a combination of interrogators or transponders combined into a single system. In this case, different layers will be present in a single IFF PDU to reflect the individual system types that are part of a combined system. For example, a Mark XIIA transponder includes both a Mode 5 and a Mode S capability. The Mode 5 transponder characteristics are reflected in a portion of Layer 1 and use the Layer 3, Mode 5 Transponder Format. The Mode S transponder characteristics are reflected in Layer 1 and use the Layer 4, Mode S Transponder Format.

Table 166 summarizes the purpose of each of the layers. Unlike Layers 1 and 2, the other layers may have a different format depending on the system type and whether it is a transponder or an interrogator.

Table 166—IFF PDU layer summary

Layer	Description	System-Specific fields/ system formats
1	*Basic System Data*. This is the only layer that is required to be included in every IFF PDU regardless of the system type. It includes the standard PDU Header record. It supports all system types. The field formats of the two Data fields and six Parameter fields may vary depending on the system type.	*System-Specific Fields* System Specific Data Data Fields 1, 2 Parameters 1 through 6
2	*Basic Emissions Data*. This layer is used for basic emissions data when required to support simulations that need detailed IFF electromagnetic characteristics. The field formats of the two Operational Parameter fields and the System-Specific Data field may vary depending on the system type.	*System-Specific Fields* Operational Parameters 1, 2 System-Specific Data
3	*Mode 5 Functional Data*. The military Mode 5 system is the only system that currently uses Layer 3.	*System Formats* Mode 5 transponder format Mode 5 interrogator format
4	*Mode S Functional Data*. The civilian Mode S system is the only system that currently uses Layer 4.	*System Formats* Mode S transponder format Mode S interrogator format
5	*Data Communications*. This layer is used to extend a layer that does not have a standard variable record section (i.e., Layers 1 and 2) and to support the emulation of real-world transponder and interrogator data link messages.	Not applicable
6, 7	Not defined.	—

f) Layers 1, 2, and 5 are common layers that have standard formats. The fields within common layers are of two types: common and system-specific fields:

1) Common Fields. Common fields have the same content and enumerations regardless of the system type. However, not all enumeration values for a common field may be applicable for a system type.

2) System-Specific Fields. System-Specific field content and enumerations depend on the system type and are defined in Annex B.

g) More than one IFF system may be associated with the same entity. If two or more transponder or interrogator system types are associated with an entity, each one shall require a separate IFF PDU. See 5.7.6.1 for the requirements associated with multiple IFF system types when associated with a single entity.

7.6.5.2 Layer 1 basic system data

Layer 1 contains IFF basic data for interrogators and transponders. Layer 1 shall always be transmitted in an IFF PDU. Layer 1 fields include variable format fields that will vary in content depending on whether it is an interrogator or transponder and the specific system type. Not all enumerations defined for a field are necessarily applicable to both interrogators and transponders or for specific system types. The content and use requirements applicable to all system types are defined here. Those that are unique to a system type are defined in Annex B.

Layer 1 shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Emitting Entity ID*. This field shall identify the entity that is the source of the emissions. This field shall be represented by an Entity Identifier record (see 6.2.28).

c) *Event ID*. This field shall contain a number generated by the issuing simulator to associate related events. This field shall be represented by an Event Identifier record (see 6.2.33).

d) *Relative Antenna Location*. This field shall specify the relative location of a designated primary antenna for this IFF system. The antenna location shall be with respect to the emitting entity's coordinate system and shall be represented by an Entity Coordinate Vector record [item a) in 6.2.96]. See the Antenna Location IFF Data record (B.2.3) to indicate any auxiliary antenna locations associated with a system.

e) *System ID*. This field shall identify a specific interrogator or transponder system for the emitting system and shall be represented by a System Identifier record (see 6.2.87).

f) *System Designator*. This field shall contain a unique decimal number assigned to this interrogator or transponder to distinguish it from multiple interrogators or transponders that are associated with the same entity. It shall be represented by an 8-bit unsigned integer. [See item d2) in 5.7.6.1.]

g) *System-Specific Data*. This is a variable format field containing 8 bits of data whose meaning is defined for each specific system. See Clause B.5.

h) *Fundamental Operational Data*. This field shall identify certain basic operational data for the IFF emitting system. This field shall be represented by a Fundamental Operational Data record (see 6.2.39).

The format of Layer 1 shall be as shown in Table 167.

Table 167—Layer 1 basic data

Field size (bits)	IFF PDU fields (Layer 1)	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 28
		Protocol Family—8-bit enumeration = 6
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused

Table 167—Layer 1 basic data (continued)

Field size (bits)	IFF PDU fields (Layer 1)	
48	Emitting Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Event ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Event Number—16-bit unsigned integer
96	Relative Antenna Location	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
48	System ID	System Type—16-bit enumeration
		System Name—16-bit enumeration
		System Mode—8-bit enumeration
		Change/Options—8-bit record
8	System Designator	8-bit unsigned integer
8	System-Specific Data	8-bit record defined by system type
128	Fundamental Operational Data	System Status—8-bit record
		Data Field 1—8-bit record
		Information Layers—8-bit record
		Data Field 2—8-bit record
		Parameter 1—16-bit record
		Parameter 2—16-bit record
		Parameter 3—16-bit record
		Parameter 4—16-bit record
		Parameter 5—16-bit record
		Parameter 6—16-bit record
Layer 1 size = 480 bits		

7.6.5.3 Layer 2 emissions data

Layer 2 contains emissions data for an interrogator or a transponder. This is used to support high-fidelity representations of IFF signals. Layer 2 is optional. However, if it is implemented by a simulation, the simulation shall adhere to the requirements for Layer 2 specified herein.

Layer 2 shall contain the following fields:

a) *Layer Header*. This field shall be represented by the Layer Header record (see 6.2.51).

b) *Beam Data*. This field shall specify beam-specific data (see 5.7.6.3) and shall be represented by a Beam Data record (see 6.2.11).

c) *Secondary Operational Data*. This field shall identify certain secondary operational data for the interrogator or transponder emitting system. This field shall be represented by a Secondary Operational Data record (see 6.2.76).

d) *IFF Fundamental Parameter Data record*. The fundamental energy radiation emissions characteristics of the mode(s) for an IFF system type shall be represented by an IFF Fundamental Parameter Data record (see 6.2.44). At least one IFF Fundamental Parameter Data record is required.

The maximum number of Layer 2 IFF Fundamental Parameter Data records that can be included in an IFF PDU is a function of the maximum PDU size.

The format of Layer 2 shall be as shown in Table 168.

Table 168—Layer 2 emissions data

Field size (bits)	IFF PDU fields (Layer 2)	
32	Layer Header	Layer Number—8-bit unsigned integer
		Layer-Specific Information—8-bit enumeration
		Length—16-bit unsigned integer
160	Beam Data	Beam Azimuth Center—32-bit floating point
		Beam Azimuth Sweep—32-bit floating point
		Beam Elevation Center—32-bit floating point
		Beam Elevation Sweep—32-bit floating point
		Beam Sweep Sync—32-bit floating point
32	Secondary Operational Data	Operational Parameter 1—8-bit record
		Operational Parameter 2—8-bit record
		Number of IFF Fundamental Parameter Data Records (*N*)—16-bit unsigned integer
192	IFF Fundamental Parameter Data record #1	ERP—32-bit floating point
		Frequency—32-bit floating point
		PgRF—32-bit floating point
		Pulse Width—32-bit floating point
		Burst Length—32-bit unsigned integer
		Applicable Modes—8-bit enumeration
		System Specific Data—24 bits defined by system type
	• • •	

Table 168—Layer 2 emissions data *(continued)*

Field size (bits)	IFF PDU fields (Layer 2)	
192	IFF Fundamental Parameter Data record #*N*	ERP—32-bit floating point
		Frequency—32-bit floating point
		PgRF—32-bit floating point
		Pulse Width—32-bit floating point
		Burst Length—32-bit unsigned integer
		Applicable Modes—8-bit enumeration
		System Specific Data—24 bits defined by system type
Layer 2 size = 224 + 192*N* bits where *N* is the number of IFF Fundamental Parameter Data records		

7.6.5.4 Layer 3 Mode 5 formats

7.6.5.4.1 General

Layer 3 shall be used to convey Mode 5 interrogator and transponder functional data. The Layer 1 System Type field shall be used to determine whether Layer 3 represents a Mode 5 capable interrogator or transponder. The information contained in this standard for Layer 3 supports basic interoperability between simulations when exchanging simulated Mode 5 interrogator and transponder data. However, when operating in a classified simulated environment, additional Mode 5 transmit and receive requirements, as well as additional data formats, are contained in the AIMS 03-1000A Technical Standard and its cited applicable documents (see Clause 2). This is a U.S. DoD For Official Use Only (FOUO) document that is available to authorized U.S. and non-U.S. personnel. Layer 3 consists of two defined formats—the Mode 5 Interrogator Format and the Mode 5 Transponder Format. See 5.7.6.3, 5.7.6.4, and Annex B for requirements related to the setting of specific field values and associated issuance and receipt rules.

NOTE—Both the U.S. and North Atlantic Treaty Organization (NATO) military implementations require that, as a minimum, Mode 5 capable systems include a basic Mode S capability. It should also be noted that a U.S. or NATO Mode 5 system will also be capable of other military and civilian modes (see the system summaries contained in B.1.3.2.) The AIMS 03-1000A document is also the technical standard for the military implementation of Mode S (see 7.6.5.5).

7.6.5.4.2 Layer 3 Mode 5 Interrogator Format

The Layer 3 Mode 5 Interrogator Format shall have the following fields:

a) *Layer Header*. This field shall be represented by the Layer Header record (see 6.2.51).

b) *Reporting Simulation*. This field shall contain the simulation reporting this IFF PDU. It shall be represented by a Simulation Address record (see 6.2.80).

c) *Mode 5 Interrogator Basic Data*. This field shall contain basic Mode 5 interrogator data that is always required to be transmitted. It shall be represented by the Mode 5 Interrogator Basic Data record (see B.2.26).

d) *Number of IFF Data Records*. This field shall indicate the number of IFF Data records and shall be represented by a 16-bit unsigned integer. If there are no records, this field shall be set to zero.

e) *IFF Data records*. These records contain additional data for a Mode 5 interrogator. These records shall conform to the variable record format of the IFF Data Specification record (see 6.2.43). See B.2.1.1 for a list of IFF Data records.

The format of Layer 3 for a Mode 5 interrogator shall be as shown in Table 169.

Table 169—Layer 3 Mode 5 Interrogator Format

Field size (bits)	IFF PDU fields (Layer 3)	
32	Layer Header	Layer Number—8-bit unsigned integer
		Layer-Specific Information—8-bit enumeration
		Length—16-bit unsigned integer
32	Reporting Simulation	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
128	Mode 5 Interrogator Basic Data	Mode 5 Interrogator Status—8-bit record
		Padding—8 bits unused
		Padding—16 bits unused
		Mode 5 Message Formats Present—32-bit record
		Interrogated Entity ID—48-bit record
		Padding—16 bits unused
16	Padding	16 bits unused
16	Number of IFF Data Records (*N*)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IFF Data record #*1*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_1 + P_1)$
		Record-Specific fields—K_1 octets
		Padding to 32-bit boundary—P_1 octets
⋮		

Table 169—Layer 3 Mode 5 Interrogator Format *(continued)*

Field size (bits)	IFF PDU fields (Layer 3)	
$48 + 8K_N + 8P_N$	IFF Data record #*N*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6 + K_N + P_N$)
		Record-Specific fields—K_N octets
		Padding to 32-bit boundary—P_N octets

Layer 3 Mode 5 Interrogator size = $224 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

N is the number of IFF data records
K_i is the length of the Record-Specific field in IFF Data record i in octets
P_i is the number of padding octets in IFF Data record i, which is $\lceil (6 + K_i)/4 \rceil 4 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.6.5.4.3 Layer 3 Mode 5 Transponder Format

The Layer 3 Mode 5 Transponder Format shall have the following fields:

a) *Layer Header*. This field shall be represented by the Layer Header record (see 6.2.51).

b) *Reporting Simulation*. This field shall contain the simulation reporting this IFF PDU. It shall be represented by a Simulation Address record (see 6.2.80).

c) *Mode 5 Transponder Basic Data*. This field shall contain basic Mode 5 transponder data that is always required to be transmitted. It shall be represented by the Mode 5 Transponder Basic Data record (see B.2.29).

d) *Number of IFF Data Records*. This field shall indicate the number of IFF Data records and shall be represented by a 16-bit unsigned integer. If there are no records, this field shall be set to zero.

e) *IFF Data records*. These records contain additional data for a Mode 5 transponder. These records shall conform to the variable record format of the IFF Data Specification record (see 6.2.43). See B.2.1.1 for a list of IFF Data records.

The format of Layer 3 for a Mode 5 transponder shall be as shown in Table 170.

Table 170—Layer 3 Mode 5 Transponder Format

Field size (bits)	IFF PDU fields (Layer 3)	
32	Layer Header	Layer Number—8-bit unsigned integer
		Layer-Specific Information—8-bit enumeration
		Length—16-bit unsigned integer

Table 170—Layer 3 Mode 5 Transponder Format *(continued)*

Field size (bits)	IFF PDU fields (Layer 3)	
32	Reporting Simulation	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
128	Mode 5 Transponder Basic Data	Mode 5 Status—16-bit record
		Personal Identification Number (PIN)—16-bit unsigned integer
		Mode 5 Message Formats Present—32-bit record
		Enhanced Mode 1—16-bit record
		National Origin—16-bit enumeration
		Supplemental Data—8-bit record
		Navigation Source—8-bit enumeration
		Figure of Merit—8-bit unsigned integer
		Padding—8 bits unused
16	Padding	16 bits unused
16	Number of IFF Data Records (N)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IFF Data record #*1*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6 + K_1 + P_1$)
		Record-Specific fields—K_1 octets
		Padding to 32-bit boundary—P_1 octets
	• • •	
$48 + 8K_N + 8P_N$	IFF Data record #*N*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6 + K_N + P_N$)
		Record-Specific fields—K_N octets
		Padding to 32-bit boundary—P_N octets

$$\text{Layer 3 Mode 5 Transponder size} = 224 + 8\sum_{i=1}^{N}(6 + K_i + P_i) \text{ bits}$$

where

N is the number of IFF data records
K_i is the length of the Record-Specific field in IFF Data record i in octets
P_i is the number of padding octets in IFF Data record i, which is $\lceil (6 + K_i)/4 \rceil 4 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$

7.6.5.5 Layer 4 Mode S formats

7.6.5.5.1 General

Layer 4 shall be used to convey Mode S interrogator and transponder functional data. The Layer 1 System Type field shall be used to determine whether Layer 4 represents a Mode S capable interrogator or transponder. The information contained in this standard for Layer 4 supports basic interoperability between simulations when exchanging simulated Mode S interrogator and transponder data. Additional Mode S transmit and receive requirements, as well as additional data formats are contained in specific International Civil Aviation Organization (ICAO) publications. These are grouped in Clause 2 under the heading "International Civil Aviation Organization (ICAO) Mode S related publications." Layer 4 consists of two defined formats—the Mode S Interrogator Format and the Mode S Transponder Format. See 5.7.6.3, 5.7.6.4, and Annex B for requirements related to the setting of specific field values and associated issuance and receipt rules.

NOTE—Both the U.S. and NATO military implementation of Mode 5 capable systems include a basic Mode S capability (see 7.6.5.4.1).

7.6.5.5.2 Layer 4 Mode S Interrogator Format

The Layer 4 Mode S Interrogator Format shall have the following fields:

a) *Layer Header*. This field shall be represented by the Layer Header record (see 6.2.51).

b) *Reporting Simulation*. This field shall contain the simulation reporting this IFF PDU. It shall be represented by a Simulation Address record (see 6.2.80).

c) *Mode S Interrogator Basic Data*. This field shall contain basic Mode S interrogator data that is always required to be transmitted. It shall be represented by the Mode S Interrogator Basic Data record (see B.2.37).

d) *Number of IFF Data Records*. This field shall indicate the number of IFF Data records and shall be represented by a 16-bit unsigned integer. If there are no records, this field shall be set to zero.

e) *IFF Data Records*. These records contain additional data for a Mode S interrogator. These records shall conform to the variable record format of the IFF Data Specification record (see 6.2.43). See B.2.1.1 for a list of IFF Data records.

The format of Layer 4 for a Mode S interrogator shall be as shown in Table 171.

Table 171—Layer 4 Mode S Interrogator Format

Field size (bits)	IFF PDU fields (Layer 4)	
32	Layer Header	Layer Number—8-bit unsigned integer
		Layer-Specific Information—8-bit enumeration
		Length—16-bit unsigned integer
32	Reporting Simulation	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer

Table 171—Layer 4 Mode S Interrogator Format *(continued)*

Field size (bits)	IFF PDU fields (Layer 4)	
192	Mode S Interrogator Basic Data	Mode S Interrogator Status—8-bit record
		Padding—8 bits unused
		Mode S Levels Present—8-bit record
		Padding—8 bits unused
		Padding—5 × 32 bits unused (160 bits)
16	Padding	16 bits unused
16	Number of IFF Data Records (N)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IFF Data record #*1*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_1 + P_1)$
		Record-Specific fields—K_1 octets
		Padding to 32-bit boundary—P_1 octets
⋮		
$48 + 8K_N + 8P_N$	IFF Data record #*N*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_N + P_N)$
		Record-Specific fields—K_N octets
		Padding to 32-bit boundary—P_N octets

$$\text{Layer 4 Mode S Interrogator size} = 288 + 8\sum_{i=1}^{N}(6 + K_i + P_i) \quad \text{bits}$$

where

N is the number of IFF data records
K_i is the length of the Record-Specific field in IFF Data record i in octets
P_i is the number of padding octets in IFF Data record i, which is $\lceil (6 + K_i)/4 \rceil 4 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.6.5.5.3 Layer 4 Mode S Transponder Format

The Layer 4 Mode S Transponder Format shall have the following fields:

a) *Layer Header*. This field shall be represented by the Layer Header record (see 6.2.51).

b) *Reporting Simulation*. This field shall contain the simulation reporting this IFF PDU. It shall be represented by a Simulation Address record (see 6.2.80).

c) *Mode S Transponder Basic Data*. This field shall contain basic Mode S transponder data that is always required to be transmitted. It shall be represented by the Mode S Transponder Basic Data record (see B.2.41).

d) *Number of IFF Data Records*. This field shall indicate the number of IFF Data records and shall be represented by a 16-bit unsigned integer. If there are no records, this field shall be set to zero.

e) *IFF Data records*. These records contain additional data for a Mode S transponder. These records shall conform to the variable record format of the IFF Data Specification record (see 6.2.43). See B.2.1.1 for a list of IFF Data records.

The format of Layer 4 for a Mode S transponder shall be as shown in Table 172.

Table 172—Layer 4 Mode S Transponder Format

Field size (bits)	IFF PDU fields (Layer 4)	
32	Layer Header	Layer Number—8-bit unsigned integer
		Layer-Specific Information—8-bit enumeration
		Length—16-bit unsigned integer
32	Reporting Simulation	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
192	Mode S Transponder Basic Data	Mode S Status—16-bit record
		Mode S Levels Present—8-bit record
		Aircraft Present Domain—8-bit enumeration
		Aircraft Identification—64-bit record
		Aircraft Address—32-bit unsigned integer
		Aircraft ID Type—8-bit enumeration
		DAP Source—8-bit record
		Mode S Altitude—16-bit record
		Capability Report—8-bit enumeration
		Padding—8 bits unused
		Padding—16 bits unused
16	Padding	16 bits unused
16	Number of IFF Data Records (*N*)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IFF Data record #*1*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6 + K_1 + P_1$)
		Record-Specific fields—K_1 octets
		Padding to 32-bit boundary—P_1 octets

Table 172—Layer 4 Mode S Transponder Format *(continued)*

Field size (bits)	IFF PDU fields (Layer 4)	
⋮		
$48 + 8K_N + 8P_N$	IFF Data record #N	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6+K_N+P_N$)
		Record-Specific fields—K_N octets
		Padding to 32-bit boundary—P_N octets

Layer 4 Mode S Transponder size = $288 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

N is the number of IFF data records
K_i is the length of the Record-Specific field in IFF Data record i in octets
P_i is the number of padding octets in IFF Data record i, which is $\lceil (6 + K_i)/4 \rceil 4 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.6.5.6 Layer 5 data communications

Layer 5 is used to convey additional information for transponder and interrogator systems not provided for in other layers where those layers do not have a standard variable record section and to convey transponder/interrogator data link messages when emulating the exact message formats. The specific requirements associated with a system type will indicate whether Layer 5 is required and which IFF data records are applicable (see 5.7.6 and Annex B).

Layer 5 shall consist of the following fields:

a) *Layer Header*. This field shall be represented by the Layer Header record (see 6.2.51).
b) *Reporting Simulation*. This field shall contain the simulation reporting this IFF PDU. It shall be represented by a Simulation Address record (see 6.2.80).
c) *Applicable Layers*. This field shall indicate to which layer(s) the IFF data records contained in Layer 5 apply. This is to support data filtering at the receive simulation. It shall use the same format as the 8-bit Information Layers record (see 6.2.45).
d) *Data Category*. This field shall indicate the category of data represented by the included IFF data records. This is to support data filtering at the receive simulation. It shall be represented by an 8-bit enumeration (see [UID 369]).
e) *Number of IFF Data Records*. This field shall indicate the number of IFF Data records and shall be represented by a 16-bit unsigned integer. If there are no records, this field shall be set to zero.
f) *IFF Data records*. These records contain additional data for an IFF PDU. These records shall conform to the variable record format of the IFF Data Specification record (see 6.2.43). See B.2.1.1 for a list of IFF Data records.

The format of Layer 5 Data Communications is shown in Table 173.

Table 173—Layer 5—data communications

Field size (bits)	IFF PDU fields (Layer 5)	
32	Layer Header	Layer Number—8-bit unsigned integer
		Layer-Specific Information—8-bit enumeration
		Length—16-bit unsigned integer
32	Reporting Simulation	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
16	Padding	16 bits unused
8	Applicable Layers	8-bit record
8	Data Category	8-bit enumeration
16	Padding	16 bits unused
16	Number of IFF Data Records (*N*)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IFF Data record #*1*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_1 + P_1)$
		Record-Specific fields—K_1 octets
		Padding to 32-bit boundary—P_1 octets
• • •		
$48 + 8K_N + 8P_N$	IFF Data record #*N*	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_N + P_N)$
		Record-Specific fields—K_N octets
		Padding to 32-bit boundary—P_N octets

Layer 5 size = $128 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

N is the number of IFF data records
K_i is the length of the Record-Specific field in IFF Data record i in octets
P_i is the number of padding octets in IFF Data record i, which is $\lceil (6 + K_i)/4 \rceil 4 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.6.5.7 Layers 6 and 7

Layers 6 and 7 do not currently have formats defined for them.

7.6.6 Supplemental Emission/Entity State (SEES) PDU

Certain supplemental information on an entity's physical state and emissions shall be communicated using the SEES PDU. See 5.7.7 for specific requirements on the use of the SEES PDU. The SEES PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).
b) *Originating Entity ID.* This field shall identify the entity that is the source of the information contained in this PDU. The field shall be represented by an Entity Identifier record (see 6.2.28).
c) *Infrared Signature Representation Index.* This field represents an index to a specific value or an index pointer to a data table accessed with other information to obtain a specific value for the infrared signature state in which the entity is currently. This field shall be represented by a 16-bit unsigned integer.
d) *Acoustic Signature Representation Index.* This field represents an index to a specific value, or an index pointer to a data table accessed with other information to obtain a specific value for the acoustic signature state in which the entity is currently. This field shall be represented by a 16-bit unsigned integer.
e) *Radar Cross-Section Signature Representation Index.* This field represents an index to a specific value or an index pointer to a data table accessed with other information to obtain a specific value for the radar cross-section signature state in which the entity is currently. This field shall be represented by a 16-bit unsigned integer.
f) *Number of Propulsion Systems.* This field shall specify the number of operational propulsion systems aboard the entity that are described in the current SEES PDU. One, several, or all of the propulsion systems on a particular entity may be described in a single SEES PDU. This field shall be represented by a 16-bit unsigned integer.
g) *Number of Vectoring Nozzle Systems.* This field shall specify the number of operational vectoring nozzle systems aboard the entity. One, several, or all of the nozzle systems on a particular entity may be described in the SEES PDU. This field shall be represented by a 16-bit unsigned integer.
h) *Propulsion Systems Data.* This field shall identify basic operational data for the propulsion systems aboard the entity. This field shall be represented by a Propulsion System Data record (see 6.2.68).
i) *Vectoring Nozzle System Data.* This field shall identify basic operational data for the vectoring nozzle systems aboard the entity. This field shall be represented by a Vectoring Nozzle System Data record (see 6.2.97).

The format of the SEES PDU shall be as shown in Table 174.

Table 174—SEES PDU

Field size (bits)	SEES PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 30
		Protocol Family—8-bit enumeration = 6
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Infrared Signature Representation Index	16-bit unsigned integer
16	Acoustic Signature Representation Index	16-bit unsigned integer
16	Radar Cross-Section Signature Representation Index	16-bit unsigned integer
16	Number of Propulsion Systems (*N*)	16-bit unsigned integer
16	Number of Vectoring Nozzle Systems (*M*)	16-bit unsigned integer
64	Propulsion System Data #1	Power Setting—32-bit floating point
		Engine RPM—32-bit floating point
	• • •	
64	Propulsion System Data #*N*	Power Setting—32-bit floating point
		Engine RPM—32-bit floating point
64	Vectoring Nozzle System Data #1	Horizontal Deflection Angle—32-bit floating point
		Vertical Deflection Angle—32-bit floating point
	• • •	
64	Vectoring Nozzle System Data #*M*	Horizontal Deflection Angle—32-bit floating point
		Vertical Deflection Angle—32-bit floating point

Table 174—SEES PDU *(continued)*

Field size (bits)	SEES PDU fields
Total SEES PDU size = 224 + 64*N* + 64*M* bits where *N* is the number of propulsion systems *M* is the number of vectoring nozzle systems	

7.7 Radio Communications protocol family

7.7.1 General

The PDUs of the Radio Communications protocol family are described in 7.7.2 through 7.7.6.

7.7.2 Transmitter PDU

Detailed information about a radio transmitter shall be communicated by issuing a Transmitter PDU. See 5.8.3 for specific requirements on the use of the Transmitter PDU. The Transmitter PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record described in 6.2.66.

b) *Radio Reference ID*. For attached radios, this field shall identify the Entity Identifier record (see 6.2.28) or Object Identifier record (see 6.2.63) to which the radio is attached. For unattached radios, this field shall contain the Unattached Identifier record (see 6.2.92). Detailed requirements for setting this field are specified in item b2) in 5.8.3.3.

c) *Radio Number*. This field shall identify a particular radio that is either associated with an entity or object or is an unattached radio. The Radio Number field shall be represented by a 16-bit unsigned integer.

   NOTE—The combination of the Radio Reference ID and the Radio Number field uniquely identifies a particular radio within a simulation exercise. This combination is referred to as the Radio Identifier (see 6.2.70). The Radio Identifier is used to associate Transmitter, Signal, and Receiver PDUs with the same radio.

d) *Radio Type*. This field shall indicate the type of radio being simulated and shall be represented by a Radio Type record (see 6.2.71).

e) *Transmit State*. This field shall specify whether a radio is off, powered but not transmitting, or powered and transmitting and shall be represented by an 8-bit enumeration (see [UID 164]).

f) *Input Source*. This field shall specify which operator position or data port is using the radio associated with the entity, or that it represents an audio jamming source. This field shall be represented by an 8-bit enumeration (see [UID 165]).

g) *Number of Variable Transmitter Parameters Records*. This field shall specify the number of Variable Transmitter Parameters records contained in the Parameter Records section. This field shall be represented by a 16-bit unsigned integer.

h) *Antenna Location*. This field shall specify the location of the radiating portion of the antenna and shall be represented by a World Coordinates record (see 6.2.98). See 5.8.3.3 and 5.8.3.4 for detailed requirements.

i) *Relative Antenna Location*. This field shall specify the relative location of the radiating portion of the antenna with respect to the associated entity. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

j) *Antenna Pattern Type.* This field shall specify the type of representation used for the radiation pattern from the antenna. The value of this field shall determine the interpretation of the Antenna Pattern record contained in the Parameter Records section. This field shall be represented by a 16-bit enumeration (see [UID 167]).

k) *Antenna Pattern Length.* This field shall specify the length in octets of the Antenna Pattern record (see 6.2.8) contained in the Parameter Records section. This field shall be represented by a 16-bit unsigned integer.

l) *Frequency.* This field shall specify the center frequency being used by the radio for transmission if the radio is not capable of frequency hopping or it is not currently in such a mode. The frequency shall be expressed in units of Hertz and shall be represented by a 64-bit unsigned integer. This field shall also be used to indicate intercom transmissions. Detailed requirements for setting this field are specified in item b13) in 5.8.3.3.

m) *Transmit Frequency Bandwidth.* This field shall identify the bandpass, in Hertz, of the radio and shall be represented by a 32-bit floating point number.

n) *Power.* This field shall specify the power of the radio expressed as the average effective radiated power being transmitted, or that would be present if transmitting, in units of decibel-milliwatts. It shall be represented by a 32-bit floating point number. See item j) in 5.8.3.2 for additional details regarding the meaning of this field.

o) *Modulation Type.* This field shall specify the type of modulation used for radio transmission. The modulation type shall be represented by a Modulation Type record (see 6.2.59).

p) *Crypto System*. This field shall indicate the encryption capability of the transmitter regardless of whether it is operating in plain or secure communications mode. This field shall be represented by a 16-bit enumeration (see [UID 166]). Detailed requirements for setting this field are specified in item b8) in 5.8.3.3.

q) *Crypto Key ID.* This 16-bit record shall identify the crypto key and shall consist of the fields as identified in Table 175. Detailed requirements for setting this field are specified in item b9) in 5.8.3.3.

Table 175—Crypto Key ID record

Field name	Bits	Data type
Pseudo Crypto Key	0–14	15-bit unsigned integer
Crypto Mode	15	1-bit enumeration
Total Crypto Key ID record size = 16 bits		

r) *Length of Modulation Parameters*. This field shall specify the length in octets of the Modulation Parameters record contained in the Parameter Records section. This field shall be represented by an 8-bit unsigned integer.

s) *Parameter Records Section*. This section consists of any parameter records applicable to this radio system type. In order, it shall contain zero or one Modulation Parameters record (see 6.2.58), zero or one Antenna Pattern record (see 6.2.8), and zero or more Variable Transmitter Parameters records (see 6.2.95).

1) Modulation Parameters. This field shall contain an optional Modulation Parameters record (see 6.2.58) associated with the radio system identified in the Radio System field of the Modulation Type record (see 6.2.59). Modulation Parameters records are defined in Annex C.
2) Antenna Pattern. This field shall contain an optional Antenna Pattern record identified by the Antenna Pattern Type field (see 6.2.8).
3) Variable Transmitter Parameters records. These fields shall contain optional Variable Transmitter Parameters records (see 6.2.95).

The format of the Transmitter PDU shall be as shown in Table 176.

Table 176—Transmitter PDU

Field size (bits)	Transmitter PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 25
		Protocol Family—8-bit enumeration = 4
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Radio Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Radio Number	16-bit unsigned integer
64	Radio Type	64-bit record
8	Transmit State	8-bit enumeration
8	Input Source	8-bit enumeration
16	Number of Variable Transmitter Parameters Records (*N*)	16-bit unsigned integer
192	Antenna Location	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
96	Relative Antenna Location	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
16	Antenna Pattern Type	16-bit enumeration
16	Antenna Pattern Length (*A*)	16-bit unsigned integer

Table 176—Transmitter PDU *(continued)*

64	Frequency	64-bit unsigned integer
32	Transmit Frequency Bandwidth	32-bit floating point
32	Power	32-bit floating point
64	Modulation Type	Spread spectrum—16-bit record
		Major Modulation—16-bit enumeration
		Detail—16-bit enumeration
		Radio System—16-bit enumeration
16	Crypto System	16-bit enumeration
16	Crypto Key ID	16-bit record
8	Length of Modulation Parameters (M)	8-bit unsigned integer
8	Padding	8 bits unused
16	Padding	16 bits unused
Parameter records section		
$8M$	Modulation Parameters	Modulation Parameters record—M octets
$8A$	Antenna Pattern	Antenna Pattern record—A octets
$48 + 8K_1 + 8P_1$	Variable Transmitter Parameters record #1	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_1 + P_1)$
		Record-Specific fields—K_1 octets
		Padding to 64-bit boundary—P_1 octets
• • •		
$48 + 8K_N + 8P_N$	Variable Transmitter Parameters record #N	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_N + P_N)$
		Record-Specific fields—K_N octets
		Padding to 64-bit boundary—P_N octets

Total Transmitter PDU size = $832 + 8M + 8A + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits

where

M is the length of the Modulation Parameters record in octets
A is the length of the Antenna Pattern record in octets
N is the number of Variable Transmitter Parameters records
K_i is the length of the Record-Specific field in Variable Transmitter Parameters record i in octets
P_i is the number of padding octets in Variable Transmitter Parameters record i, which is
$\lceil (6 + K_i)/8 \rceil 8 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.7.3 Signal PDU

The actual transmission of voice, audio, or other data shall be communicated by issuing a Signal PDU. See 5.8.4 for specific requirements on the use of the Signal PDU. The Signal PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record described in 6.2.66.

b) *Radio Reference ID*. For attached radios, this field shall identify the Entity Identifier record (see 6.2.28) or Object Identifier record (see 6.2.63) to which the radio is attached. For unattached radios, this field shall contain the Unattached Identifier record (see 6.2.92). Detailed requirements for setting this field are specified in item b2) in 5.8.3.3.

c) *Radio Number*. This field shall identify a particular radio that is either associated with an entity or object or is an unattached radio. The Radio Number field shall be represented by a 16-bit unsigned integer.

NOTE—The combination of the Radio Reference ID and the Radio Number field uniquely identifies a particular radio within a simulation exercise. This combination is referred to as the Radio Identifier (see 6.2.70). The Radio Identifier is used to associate Transmitter, Signal, and Receiver PDUs with the same radio.

d) *Encoding Scheme*. This field specifies the encoding used in the Data field of this PDU and shall be represented by a 16-bit Encoding Scheme record as described in Table 177:

   1) Encoding Type or Number of TDL Messages. The definition of this 14-bit field depends on the value of the Encoding Class field and can be either an encoding type or the number of TDL messages in this Signal PDU. Encoding Type is a 14-bit enumeration (see [UID 271]). Number of TDL Messages is represented as a 14-bit unsigned integer.

   2) Encoding Class. This is a 2-bit enumeration (see [UID 270]).

   Detailed requirements for setting this field are specified in 5.8.4.3.2.

Table 177—Encoding Scheme record

Field name	Bits	Data type
Encoding Type or Number of TDL Messages	0–13	14-bit enumeration or 14-bit unsigned integer
Encoding Class	14–15	2-bit enumeration
Total Encoding Scheme record size = 16 bits		

e) *TDL Type*. This field shall specify the TDL Type and shall be represented as a 16-bit enumeration (see [UID 178]). Detailed requirements for setting this field are specified in 5.8.4.3.2.

f) *Sample Rate*. This field shall specify either the sample rate or the data rate and shall be represented by a 32-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.4.3.2.

g) *Data Length*. This field shall specify the number of bits of digital voice audio or digital data being sent in this Signal PDU and shall be represented by a 16-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.4.3.2.

h) *Samples*. This field shall specify the number of samples in this PDU and shall be represented by a 16-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.4.3.2.

i) *Data.* This field shall specify the audio or digital data conveyed by the radio transmission. The length of the valid data contained in this field shall be the value of the Data Length field. The Data field shall be zero-padded to comply with the overall PDU length requirements (see 6.3.2). Detailed requirements for setting this field are specified in 5.8.4.3.2.

The format of the Signal PDU shall be as shown in Table 178.

Table 178—Signal PDU

Field size (bits)	Signal PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 26
		Protocol Family—8-bit enumeration = 4
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Radio Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Radio Number	16-bit unsigned integer
16	Encoding Scheme	16-bit record
16	TDL Type	16-bit enumeration
32	Sample Rate	32-bit unsigned integer
16	Data Length (K)	16-bit unsigned integer
16	Samples	16-bit unsigned integer
K	Data	K bits
P	Padding	Padding to 32-bit boundary—P bits
Total Signal PDU size = 256 + K + P bits where K is the data length in bits P is the number of padding bits, which is $\lceil K/32 \rceil 32 - K$ $\lceil x \rceil$ is the largest integer < $x + 1$.		

7.7.4 Receiver PDU

Communication of the receiver state shall be communicated with a Receiver PDU. See 5.8.5 for specific requirements on the use of the Receiver PDU. The Receiver PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record described in 6.2.66.

b) *Radio Reference ID*. For attached radios, this field shall identify the Entity Identifier record (see 6.2.28) or Object Identifier record (see 6.2.63) to which the radio is attached. For unattached radios, this field shall contain the Unattached Identifier record (see 6.2.92). Detailed requirements for setting this field are specified in item b2) in 5.8.3.3.

c) *Radio Number*. This field shall identify a particular radio that is either associated with an entity or object or is an unattached radio. The Radio Number field shall be represented by a 16-bit unsigned integer.

   NOTE—The combination of the Radio Reference ID and the Radio Number field uniquely identifies a particular radio within a simulation exercise. This combination is referred to as the Radio Identifier (see 6.2.70). The Radio Identifier is used to associate Transmitter, Signal, and Receiver PDUs with the same radio.

d) *Receiver State.* This field shall indicate the state of the receiver, which shall be either idle or active and shall be represented by a 16-bit enumeration (see [UID 179]).

e) *Received Power.* This field shall indicate the radio frequency power received, after applying any propagation loss and antenna gain, and shall be represented by a 32-bit floating point number in units of decibel milliwatts.

f) *Transmitter Radio Reference ID.* This field shall identify the radio that is the source of the transmission that is currently being received. The selection of the received transmitter depends on the characteristics and state of the simulation receiver. For attached radios, this field shall identify the Entity Identifier record (see 6.2.28) or Object Identifier record (see 6.2.63) to which the radio is attached. For unattached radios, this field shall contain the Unattached Identifier record (see 6.2.92). Detailed requirements for setting this field are specified in item b2) in 5.8.3.3.

g) *Transmitter Radio Number.* This field shall identify the particular radio within the radio transmitter cited in item f) that is the source of the radio transmission. The Transmitter Radio Number shall be represented by a 16-bit unsigned integer.

   NOTE—The combination of the Transmitter Radio Reference ID and the Transmitter Radio Number field uniquely identifies a particular radio within a simulation exercise. This combination is referred to as the Radio Identifier (see 6.2.70). The Radio Identifier is used to associate Transmitter, Signal, and Receiver PDUs with the same radio.

The format of the Receiver PDU shall be as shown in Table 179.

Table 179—Receiver PDU

Field size (bits)	Receiver PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 27
		Protocol Family—8-bit enumeration = 4
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Radio Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Radio Number	16-bit unsigned integer
16	Receiver State	16-bit enumeration
16	Padding	16 bits unused
32	Received Power	32-bit floating point
48	Transmitter Radio Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Transmitter Radio Number	16-bit unsigned integer
Total Receiver PDU size = 288 bits		

7.7.5 Intercom Signal PDU

The actual transmission of intercom voice or data shall be communicated by using an Intercom Signal PDU. See 5.8.6 for specific requirements on the use of the Intercom Signal PDU. The Intercom Signal PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Intercom Reference ID*. For attached intercoms, this field shall identify the entity (see 6.2.28) or object identifier (see 6.2.63) to which the intercom is attached. For unattached intercoms, this field shall contain the Unattached Identifier record (see 6.2.92).

c) *Intercom Number*. This field shall identify a particular intercom that is either associated with an entity or object or is an unattached intercom. This field shall be represented by a 16-bit unsigned integer.

NOTE—The combination of the Intercom Reference ID and the Intercom Number field uniquely identifies a particular intercom within a simulation exercise. This combination is referred to as the Intercom Identifier (see 6.2.47). The Intercom Identifier is used to associate Intercom Signal and Intercom Control PDUs for the same intercom.

d) *Encoding Scheme.* This field shall specified by and represented by the Encoding Scheme field describe in item d) in 7.7.3.

e) *TDL Type.* This field shall be specified by and represented by the TDL Type field described in item e) in 7.7.3.

f) *Sample Rate.* This field shall be specified by and represented by the Sample Rate field described in item f) in 7.7.3.

g) *Data Length.* This field shall be specified by and represented by the Data Length field described in item g) in 7.7.3.

h) *Samples.* This field shall be specified by and represented by the Samples field described in item h) in 7.7.3.

i) *Data.* This field shall specify the audio or digital data conveyed by the transmission. This field shall be represented by the Data field described in item i) in 7.7.3.

The format of the Intercom Signal PDU shall be as shown in Table 180.

Table 180—Intercom Signal PDU

Field size (bits)	Intercom Signal PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 31
		Protocol Family—8-bit enumeration= 4
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Intercom Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Intercom Number	16-bit unsigned integer
16	Encoding Scheme	16-bit record
16	TDL Type	16-bit enumeration
32	Sample Rate	32-bit unsigned integer
16	Data Length (K)	16-bit unsigned integer
16	Samples	16-bit unsigned integer

Table 180—Intercom Signal PDU *(continued)*

Field size (bits)	Intercom Signal PDU fields	
K	Data	K bits
P	Padding	Padding to 32-bit boundary—P bits
Total Intercom Signal PDU size = $256 + K + P$ bits where K is the data length in bits P is the number of padding bits, which is $\lceil K/32 \rceil 32 - K$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

7.7.6 Intercom Control PDU

Detailed information about the state of an intercom device and the actions it is requesting of another intercom device or the response to a requested action shall be communicated by an Intercom Control PDU. See 5.8.7 for specific requirements on the use of the Intercom Control PDU. The Intercom Control PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Control Type.* This field shall specify the type of control requested in the PDU. This field shall be represented by an 8-bit enumeration (see [UID 180]).

c) *Communications Channel Type.* This 8-bit field shall identify the type of communications channel and shall be composed of a 1-bit enumeration specifying the communications class (see [UID 416]) and a 7-bit enumeration specifying the communications type (see [UID 181]). The format of the Communications Channel Type field shall be as shown in Table 181. Detailed requirements for setting this field are specified in 5.8.7.3.

Table 181—Communications Channel Type

Field name	Bits	Data type
Communications Type	0–6	7-bit enumeration
Communications Class	7	1-bit enumeration
Total Communications Channel Type size = 8 bits		

d) *Source Intercom Reference ID*. This field shall identify the source of the intercom. For attached intercoms, this field shall identify the entity (see 6.2.28) or object (see 6.2.63) to which the intercom is attached. For unattached intercoms, this field shall contain the Unattached Identifier record (see 6.2.92).

e) *Source Intercom Number*. This field shall identify the specific intercom device being interfaced and/or simulated within an intercom. This field shall be represented by a 16-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.7.3.

f) *Source Line ID.* This field shall identify the line number to which this intercom control refers and shall be represented by an 8-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.7.3.

g) *Transmit Priority.* This field shall identify the priority of this message relative to transmissions from other intercom devices on the same channel and shall be represented by an 8-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.7.3.

h) *Transmit Line State.* This field shall identify the current transmit state of the line at the intercom source. This field shall be represented by an 8-bit enumeration (see [UID 183]).

i) *Command.* This field shall specify details of a request or acknowledge and shall be represented by an 8-bit enumeration (see [UID 182]). Detailed requirements for setting this field are specified in 5.8.7.3.

j) *Master Intercom Reference ID*. This field shall specify the master identifier of the entity, object, or unattached intercom identifier that has created this intercom channel. For attached intercoms, this field shall identify the entity (see 6.2.28) or object (see 6.2.63) to which the intercom is attached. For unattached intercoms, this field shall contain the Unattached Identifier record (see 6.2.92).

k) *Master Intercom Number*. This field shall identify the specific intercom device that has created this intercom channel and shall be represented by a 16-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.7.3.

l) *Master Channel ID*. This field shall identify a unique intercom channel created by this Master Intercom Reference ID and Master Intercom Number pair and shall be represented by a 16-bit unsigned integer. Detailed requirements for setting this field are specified in 5.8.7.3.

m) *Intercom Parameters Length.* This field shall specify the length in octets of the optional Intercom Parameters. The field shall be represented by a 32-bit unsigned integer.

n) *Intercom Parameters.* This field shall specify intercom parameters to describe additional information, including destination(s) of the Intercom Control PDU, as required. This field shall be represented by an Intercom Communications Parameters record (see 6.2.46).

The format of the Intercom Control PDU shall be as shown in Table 182.

Table 182—Intercom Control PDU

Field size (bits)	Intercom Control PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 32
		Protocol Family—8-bit enumeration = 4
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
8	Control Type	8-bit enumeration
8	Communications Channel Type	8-bit record

Table 182—Intercom Control PDU *(continued)*

Field size (bits)	Intercom Control PDU fields	
48	Source Intercom Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Source Intercom Number	16-bit unsigned integer
8	Source Line ID	8-bit unsigned integer
8	Transmit Priority	8-bit unsigned integer
8	Transmit Line State	8-bit enumeration
8	Command	8-bit enumeration
48	Master Intercom Reference ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Master Intercom Number	16-bit unsigned integer
16	Master Channel ID	16-bit unsigned integer
32	Intercom Parameters Length (K)	32-bit unsigned integer
$8K$	Intercom Parameters	Record Type—16-bit enumeration
		Record Length—16-bit unsigned integer (R)
		Record-Specific fields—R octets
		Padding to 32-bit boundary—P octets
Total Intercom Control PDU size = 320 + $8K$ bits where K is the length of the Intercom Parameters record in octets, which is $4 + R + P$ R is the length of the Record-Specific field in the Intercom Parameters record in octets P is the number of padding octets, which is $\lceil R/4 \rceil 4 - R$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

7.8 Entity Management protocol family

7.8.1 General

The PDUs of the Entity Management protocol family are described in 7.8.2 through 7.8.5.

7.8.2 Aggregate State PDU

Detailed information about aggregating entities and communicating information about the aggregated entities shall be communicated using the Aggregate State PDU. See 5.9.2.2 for specific requirements on the use of the Aggregate State PDU. The Aggregate State PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by a PDU Header record (see 6.2.66).

b) *Aggregate ID.* This field shall identify the aggregate issuing the PDU and shall be represented by an Aggregate Identifier record (see 6.2.3).

c) *Force ID.* This field shall specify the common force to which the aggregate belongs. Aggregates shall not group together entities belonging to opposing forces. This field shall be represented by an 8-bit enumeration (see [UID 6]).

d) *Aggregate State.* This field shall identify the state of the aggregate and shall be represented by an 8-bit enumeration (see [UID 204]).

e) *Aggregate Type record.* This field shall specify the aggregate type and shall be represented by an Aggregate Type record (see 6.2.5).

f) *Formation.* This field shall identify the formation of the aggregate and shall be represented by a 32-bit enumeration (see [UID 205]).

g) *Aggregate Marking.* This field shall identify the unique marking associated with the aggregate. This field should not be used to encode other information. This field shall be represented by an Aggregate Marking record (see 6.2.4).

h) *Dimensions.* This field shall identify the bounding space, in meters, that the aggregate occupies and shall be represented by three 32-bit floating point numbers (x, y, and z). These measurements are taken along the orientation axes of the aggregate. As shown in Figure 51, measurements x, y, and z are taken from the center of mass of the aggregate to the shortest distance along an axis in the positive or negative direction that includes all of the constituent entities.

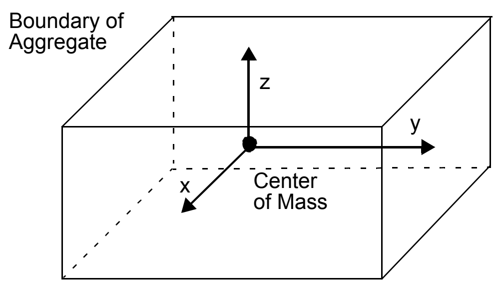


Figure 51—Aggregate dimensions

i) *Orientation.* This field shall specify the orientation of the aggregate. This field shall be represented by the Euler Angles record (see 6.2.32). An aggregate's orientation shall be calculated by taking the average orientation of its constituent entities.

j) *Center of Mass.* This field shall specify the location of the aggregate's center of mass in the simulated world and shall be represented by a World Coordinates record (see 6.2.98).

k) *Velocity.* This field shall specify an aggregate's linear velocity. The coordinate system for an aggregate's linear velocity depends on the dead reckoning algorithm used. This field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96]. As shown in Figure 52, an aggregate's velocity is calculated by taking the average velocity of its constituent entities. The velocity of the aggregate is directed from the center of mass.

l) *Number of Aggregate IDs.* This field shall specify the number of subaggregates that are transmitting Aggregate State PDUs and shall be represented by a 16-bit unsigned integer.

m) *Number of Entity IDs.* This field shall specify the number of constituent entities that are transmitting Entity State PDUs and shall be represented by a 16-bit unsigned integer.

n) *Number of Silent Aggregate Systems.* This field shall specify the number of subaggregates that are not transmitting Aggregate State PDUs and shall be represented by a 16-bit unsigned integer.

o) *Number of Silent Entity Systems.* This field shall specify the number of constituent entity systems that are not transmitting Entity State PDUs and shall be represented by a 16-bit unsigned integer.

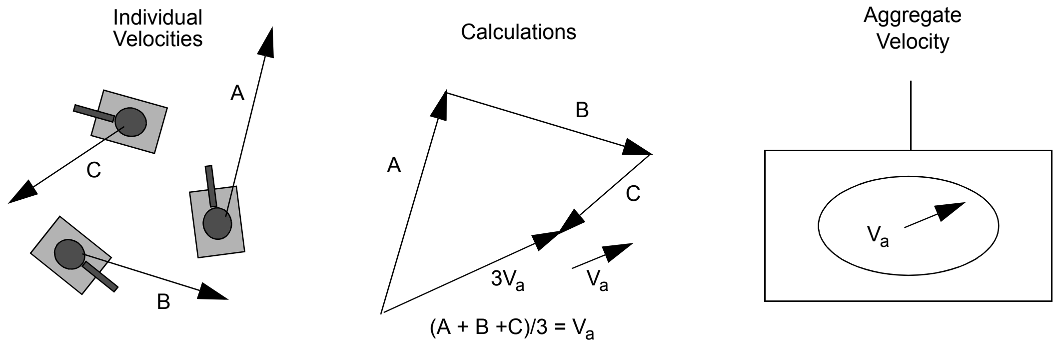


Figure 52—Aggregate velocity calculations

p) *Aggregate IDs.* These fields shall identify subaggregates that are transmitting Aggregate State PDUs. This field shall be represented by an Aggregate Identifier record (see 6.2.3).

q) *Entity IDs.* These fields shall identify the constituent entities that are transmitting Entity State PDUs. This field shall be represented by an Entity Identifier record (see 6.2.28).

r) *Silent Aggregate Systems.* These fields contain information about subaggregates not producing Aggregate State PDUs. Each element in the list shall contain the following information:

   1) Number of Aggregates. This field shall specify the number of aggregates that have the type specified by the following field and shall be represented by a 16-bit unsigned integer.

   2) Aggregate Type. This field shall specify the aggregates common to this system list. This field shall be represented by an Aggregate Type record (see 6.2.5).

s) *Silent Entity Systems.* These fields contain information about entities not producing Entity State PDUs. Each element in the list shall be represented by a Silent Entity System record (see 6.2.79).

t) *Number of Variable Datum Records.* This field shall specify the number of variable datum records to follow and shall be represented by a 32-bit unsigned integer.

u) *Variable Datum records.* This field shall specify extra data that is used by the entity-level and aggregate-level simulations to transfer control and correlate the simulation of entities in an aggregate and shall be represented by a Datum Specification record (see 6.2.18). The contents of these records shall be determined before each exercise. The first Variable Datum record may be used to indicate a flag that tells that the composition or formation of the unit has changed.

The format of the Aggregate State PDU shall be as shown in Table 183.

Table 183—Aggregate State PDU

Field size (bits)	Aggregate State PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 33
		Protocol Family—8-bit enumeration = 7
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Aggregate ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Aggregate Number—16-bit unsigned integer
8	Force ID	8-bit enumeration
8	Aggregate State	8-bit enumeration
64	Aggregate Type	Aggregate Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
32	Formation	32-bit enumeration
256	Aggregate Marking	Character Set—8-bit enumeration
		31 8-bit unsigned integers
96	Dimensions	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
96	Orientation	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point

Table 183—Aggregate State PDU *(continued)*

Field size (bits)	Aggregate State PDU fields	
192	Center of Mass (location)	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
96	Velocity	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
16	Number of Aggregates IDs (*W*)	16-bit unsigned integer
16	Number of Entity IDs (*J*)	16-bit unsigned integer
16	Number of Silent Aggregate Systems (*N*)	16-bit unsigned integer
16	Number of Silent Entity Systems (*S*)	16-bit unsigned integer
48	Aggregate ID #1	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Aggregate Number—16-bit unsigned integer
	⋮	
48	Aggregate ID #*W*	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Aggregate Number—16-bit unsigned integer
48	Entity ID #1	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
	⋮	
48	Entity ID #*J*	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
8*P*	Padding	Padding to 32-bit boundary—*P* octets
96	Silent Aggregate System #1	Number of Aggregates—16-bit unsigned integer
		Padding—16 bits unused
		Aggregate Type—Aggregate Type record
	⋮	

Table 183—Aggregate State PDU *(continued)*

Field size (bits)	Aggregate State PDU fields	
96	Silent Aggregate System #N	Number of Aggregates—16-bit unsigned integer
		Padding—16 bits unused
		Aggregate Type—Aggregate Type record
$96 + 32A_1$	Silent Entity System #1	Silent Entity System record
	• • •	
$96 + 32A_S$	Silent Entity System #S	Silent Entity System record
32	Number of Variable Datum Records (M)	32-bit unsigned integer
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
	• • •	
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Aggregate State PDU size = $1088 + 48W + 48J + 16P + 96N + \sum_{j=1}^{S} (96 + 32A_j) + \sum_{i=1}^{M} (64 + K_i + P_i)$ bits

where

W is the number of Aggregate IDs
J is the number of Entity IDs
P is the number of padding octets, which is $\lceil (W+J)/4 \rceil 4 - (W+J)$
N is the number of silent aggregate systems
S is the number of silent entity systems
A_j is number of appearance records in silent entity system j
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i/64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.8.3 IsGroupOf PDU

Information about a particular group of entities (grouped together for the purposes of network bandwidth reduction or aggregation) shall be communicated by issuing an IsGroupOf PDU. See 5.9.3 for specific requirements on the use of the IsGroupOf PDU. The IsGroupOf PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Group ID*. This field shall identify the entity that represents the group of entities and shall be represented by a Group Identifier record (see 6.2.42).

c) *Grouped Entity Category.* This field shall describe the type of entities constituting the group. This field shall be represented by an 8-bit enumeration (see [UID 213]).

d) *Number of Grouped Entities.* This field shall specify the number of individual entities constituting the group and shall be represented by an 8-bit unsigned integer.

e) *Group Reference Point.* This field shall specify the location of the group as a latitude and longitude, in radians, that will be used as the reference point from which the locations of all other grouped entities are based. The third coordinate of the Reference Point, which will not be transmitted in the IsGroupOf PDU, is defined to be 100 m below Adjusted Mean Sea Level to compensate for the lowest surface point on the Earth. The latitude and longitude values shall be represented by 64-bit floating point numbers.

f) *Group Entity Description (GED).* This field shall consist of GED records that specify information about each entity within the group. The type and size of the GED records depend on the type of group enumerated by the Grouped Entity Category field. See [UID 213] for the categories and cross-references to the GED records.

The format of the IsGroupOf PDU shall be as shown in Table 184.

Table 184—IsGroupOf PDU

Field size (bits)	IsGroupOf PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration
		Protocol Family—8-bit enumeration = 34
		Timestamp—32-bit unsigned integer = 7
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Group ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Group Number—16-bit unsigned integer
8	Grouped Entity Category	8-bit enumeration
8	Number of Grouped Entities (*N*)	8-bit unsigned integer
32	Padding	32 bits unused

Table 184—IsGroupOf PDU *(continued)*

<table>
<tr><th>Field size (bits)</th><th colspan="2">IsGroupOf PDU fields</th></tr>
<tr><td rowspan="2">128</td><td rowspan="2">Group Reference Point</td><td>Latitude—64-bit floating point</td></tr>
<tr><td>Longitude—64-bit floating point</td></tr>
<tr><td>8G</td><td>Group Entity Description #1</td><td>GED record - G octets</td></tr>
<tr><td colspan="3">•
•
•</td></tr>
<tr><td>8G</td><td>Group Entity Description #N</td><td>GED record - G octets</td></tr>
<tr><td colspan="3">Total IsGroupOf PDU size = 320 + 8NG bits

where

N is the number of grouped entities
G is the length of a GED record in octets</td></tr>
</table>

7.8.4 Transfer Ownership (TO) PDU

Information initiating the dynamic allocation and control of simulation entities between two simulation applications is initiated by a TO PDU. See 5.9.4 for specific requirements on the use of the TO PDU. The TO PDU shall contain the following fields:

a) *Simulation Management PDU Header.* These fields shall identify the PDU Header information, the originating entity, and the intended receiving entity. The Simulation Management PDU Header shall be represented by a Simulation Management PDU Header record (see 6.2.82).

b) *Request ID.* This field shall identify the transfer ownership request. This field shall be represented by a 32-bit unsigned integer.

c) *Required Reliability Service.* This field shall specify the required level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

d) *Transfer Type.* This field shall specify the type of transfer desired and shall be represented by an 8-bit enumeration (see [UID 224]).

e) *Transfer Entity ID.* This field shall identify the entity for which control is being requested to transfer. This field shall be represented by an Entity Identifier record (see 6.2.28).

f) *Record Information.* This field shall specify the number of record sets, the identification of the record, the serial number of the record, the record length, the record count, and the specific record values. This field shall be represented by a Record Specification record (see 6.2.73).

The format of the TO PDU shall be as shown in Table 185.

Table 185—TO PDU

Field size (bits)	TO PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 35
		Protocol Family—8-bit enumeration = 7
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Transfer Type	8-bit enumeration
48	Transfer Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
32	Number of Record Sets (R)	32-bit unsigned integer
$128 + L_1Q_1 + P_1$	Record Set #1	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_1)
		Record Count—16-bit unsigned integer (Q_1)
		Record Values—(L_1Q_1) bits
		Padding to 64-bit boundary—P_1 bits
	⋮	

Table 185—TO PDU *(continued)*

Field size (bits)	TO PDU fields	
$128 + L_R Q_R + P_R$	Record Set #R	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_R)
		Record Count—16-bit unsigned integer (Q_R)
		Record Values—($L_R Q_R$) bits
		Padding to 64-bit boundary—P_R bits

Total TO PDU size = $320 + \sum_{i=1}^{R} (128 + L_i Q_i + P_i)$ bits

where

R is the number of Record Sets
L_i is the length of the record in Record Set i in bits
Q_i is the number of records included in Record Set i
P_i is the number of padding bits in Record Set i, which is $\lceil (L_i Q_i)/64 \rceil 64 - L_i Q_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.8.5 IsPartOf PDU

The joining of two or more simulation entities forms a single entity with constituent parts. This joining is communicated by issuance and acknowledgment of the IsPartOf PDU. See 5.9.5 for specific requirements on the use of the IsPartOf PDU. The IsPartOf PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Originating Simulation ID.* This field shall identify the simulation issuing the PDU and shall be represented by a Simulation Identifier record (see 6.2.81).

c) *Receiving Entity ID.* This field shall identify the requested part entity and shall be represented by an Entity Identifier record (see 6.2.28).

d) *Relationship.* This field provides information that addresses the specific purpose (nature) for joining the entities and the relationship of the part entity to it host entity. This field shall be represented by a Relationship record (see 6.2.74).

e) *Part Location.* This field specifies the location of the part's centroid in the host entity's coordinate system (see 5.9.5.3 for details on how to set the fields of this record). This field shall be represented by three 32-bit floating point fields, as specified in Table 186.

f) *Named Location ID.* This field identifies the discrete positional relationship that the host entity will have with respect to the part entity upon receipt of an Acknowledgment PDU with a response indicating the requested simulation application is able to comply with the requested action. This field shall be represented by a Named Location Identification record (see 6.2.62).

g) *Part Type.* This field shall identify the part type and shall be represented by an Entity Type record (see 6.2.30).

The format of the IsPartOf PDU shall be as shown in Table 186.

Table 186—IsPartOf PDU

Field size (bits)	IsPartOf PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 36
		Protocol Family—8-bit enumeration = 7
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
32	Relationship	Nature—16-bit enumeration
		Position—16-bit enumeration
96	Part Location	Component A—32-bit floating point
		Component B—32-bit floating point
		Component C—32-bit floating point
32	Named Location ID	Station Name—16-bit enumeration
		Station Number—16-bit unsigned integer
64	Part Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
Total IsPartOf PDU size = 416 bits		

7.9 Minefield protocol family

7.9.1 General

The PDUs of the Minefield protocol family are described in 7.9.2 through 7.9.5.

7.9.2 Minefield State PDU

The Minefield State PDU shall provide information about the complete minefield. The minefield presence, perimeter, protocol mode, and types of mines contained within the minefield shall be communicated through the Minefield State PDU. See 5.10.2 for specific requirements on the use of the Minefield State PDU. The Minefield State PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Minefield ID.* This field shall identify the minefield issuing the Minefield State PDU. This field shall be represented by a Minefield Identifier record (see 6.2.56).

c) *Minefield Sequence Number.* This field shall specify a change in state of a minefield as a result of a change in minefield information or a change in the state, in accordance with the rules specified in 5.10.2.3, of any of the mines contained therein. This field shall be represented as a 16-bit unsigned integer.

d) *Force ID.* This field shall identify the force to which the issuing minefield belongs. This field shall be represented by an 8-bit enumeration (see [UID 6]).

e) *Number of Perimeter Points.* This field shall identify the number of points in the perimeter of the minefield. This field shall be defined by an 8-bit unsigned integer.

f) *Minefield Type.* This field shall identify the minefield type of the issuing minefield. This field shall be represented by an Entity Type record (see 6.2.30).

g) *Number of Mine Types.* This field shall specify the number of different mine types employed in the minefield. This field shall be represented by a 16-bit unsigned integer.

h) *Minefield Location.* This field shall indicate the location of the center of the minefield. This field shall be represented by a World Coordinates record (see 6.2.98).

i) *Minefield Orientation.* This field shall identify the orientation of the minefield. This field shall be represented by an Euler Angles record (see 6.2.32).

j) *Appearance.* This field shall specify the appearance information needed for displaying the symbology of the minefield as a doctrinal minefield graphic. This field shall be defined as a 16-bit record (see [UID 190]).

k) *Protocol Mode.* This field shall specify which protocol mode is being used to communicate the minefield data. This field shall be represented by a Protocol Mode record (see 6.2.69).

l) *Perimeter Point Coordinates.* This field shall specify the location of each perimeter point, relative to the Minefield Location field. Only the *x* and *y* coordinates of each perimeter point shall be specified. Each coordinate shall be represented by a 32-bit floating point number.

m) *Mine Type.* This field shall specify the type of each mine contained within the minefield. This field shall be represented by an Entity Type record (see 6.2.30).

The format of the Minefield State PDU shall be as shown in Table 187.

Table 187—Minefield State PDU

Field size (bits)	Minefield State PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 37
		Protocol Family—8-bit enumeration = 8
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Minefield ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Minefield Number—16-bit unsigned integer
16	Minefield Sequence Number	16-bit unsigned integer
8	Force ID	8-bit enumeration
8	Number of Perimeter Points (N)	8-bit unsigned integer
64	Minefield Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
16	Number of Mine Types (M)	16-bit unsigned integer
192	Minefield Location	X-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
96	Minefield Orientation	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
16	Appearance	16-bit record
16	Protocol Mode	16-bit record

Table 187—Minefield State PDU *(continued)*

Field size (bits)	Minefield State PDU fields	
64	Perimeter Point Coordinate #1	*x*-component—32-bit floating point
		y-component—32-bit floating point
	• • •	
64	Perimeter Point Coordinate #*N*	*x*-component—32-bit floating point
		y-component—32-bit floating point
64	Mine Type #1	Entity Type record—64 bits
	• • •	
64	Mine Type #M	Entity Type record—64 bits
Total Minefield State PDU size = 576 + 64N + 64M bits where N is the Number of Perimeter Points M is the Number of Mine Types		

7.9.3 Minefield Query PDU

The Minefield Query PDU shall provide the means by which a simulation shall query a minefield simulation for information on the individual mines contained within a minefield when operating in QRP mode. See 5.10.3 for specific requirements on the use of the Minefield Query PDU. The Minefield Query PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Minefield ID*. This field shall identify the minefield to which this query is addressed. This field shall be represented by a Minefield Identifier record (see 6.2.56).

c) *Requesting Simulation ID*. This field shall identify the simulation that requested the information from the minefield simulation. This field shall be represented by a Simulation Identifier record (see 6.2.81).

d) *Request ID*. This field shall identify the minefield query request. This field shall be represented by an 8-bit unsigned integer.

e) *Number of Perimeter Points*. This field shall identify the number of points in the perimeter of the area of interest defined by the requesting entity. This field shall be represented by an 8-bit unsigned integer.

f) *Number of Sensor Types*. This field shall identify the number of sensor types employed by the requesting simulation (see 5.10.3.3). This field shall be represented by an 8-bit unsigned integer.

g) *Data Filter*. This field shall identify which of the optional fields in the Minefield Data PDU are being requested. The data filter field shall be represented as a 32-bit record (see 6.2.16).

h) *Requested Mine Type*. This field shall specify the type of mine being queried by the requesting simulation (see 5.10.3.3). This field shall be represented by an Entity Type record (see 6.2.30).

i) *Requested Perimeter Point Coordinates.* This field shall specify the location of each perimeter point in the requested area, relative to the Minefield Location field from the Minefield State PDU. Only the *x* and *y* coordinates shall be specified. Each coordinate shall be represented by a 32-bit floating point number.

j) *Sensor Type.* This field shall specify the type of sensor that is requesting the data. The type of sensor determines which scalar detection coefficients shall be included in the Minefield Data PDU. This field shall be represented by a Minefield Sensor Type record (see 6.2.57).

The format of the Minefield Query PDU shall be as shown in Table 188.

Table 188—Minefield Query PDU

Field size (bits)	Minefield Query PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 38
		Protocol Family—8-bit enumeration = 8
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Minefield ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Minefield Number—16-bit unsigned integer
48	Requesting Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
8	Request ID	8-bit unsigned integer
8	Number of Perimeter Points (*N*)	8-bit unsigned integer
8	Padding	8 bits unused
8	Number of Sensor Types (*M*)	8-bit unsigned integer
32	Data Filter	32-bit record
64	Requested Mine Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration

Table 188—Minefield Query PDU *(continued)*

Field size (bits)	Minefield Query PDU fields	
64	Requested Perimeter Point Coordinate #1	x-component—32-bit floating point
		y-component—32-bit floating point
• • •		
64	Requested Perimeter Point Coordinate #N	x-component—32-bit floating point
		y-component—32-bit floating point
16	Sensor Type #1	Minefield Sensor Type record
• • •		
16	Sensor Type #M	Minefield Sensor Type record
$8P$	Padding	Padding to 32-bit boundary—P octets
Total Minefield Query PDU size = $320 + 64N + 16M + 8P$ bits where N is the Number of Perimeter Points M is the Number of Sensor Types P is the number of padding octets, which is $2(\lceil M/2 \rceil 2 - M)$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

7.9.4 Minefield Data PDU

The Minefield Data PDU shall provide information on individual mines contained within a minefield. See 5.10.4 for specific requirements on the use of the Minefield Data PDU. The Minefield Data PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Minefield ID*. This field shall identify the minefield to which the mines in this PDU belong. This field shall be represented by a Minefield Identifier record (see 6.2.56).

c) *Requesting Simulation ID*. This field shall identify the simulation that requested the information from the minefield simulation in QRP mode. In heartbeat mode, the Requesting Simulation ID shall contain the value NO_SITE: NO_APPLIC: NO_ENTITY. This field shall be represented by a Simulation Identifier record (see 6.2.81).

d) *Minefield Sequence Number*. This field shall identify the matching minefield sequence number from the associated Minefield State PDU. This field shall be represented as a 16-bit unsigned integer.

e) *Request ID*. This field shall identify the matching response to a request for mine information from the minefield simulation made by means of a Minefield Query PDU in QRP mode. In heartbeat mode, this field shall contain the value zero. This field shall be represented by an 8-bit unsigned integer.

f) *PDU Sequence Number*. This field shall specify the number of the current Minefield Data PDU in a sequence of Minefield Data PDUs sent in response to a Minefield Query PDU when operating in

QRP mode. This field shall be represented by an 8-bit unsigned integer. When operating in heartbeat mode, this field is unused and shall contain the value zero.

g) *Number of PDUs.* This field shall specify the total number of Minefield Data PDUs being sent in response to a Minefield Query PDU when operating in QRP mode. This field shall be represented by an 8-bit unsigned integer. When operating in heartbeat mode, this field is unused and shall contain the value zero.

h) *Number of Mines.* This field shall specify the number of mines of the same type contained in this Minefield Data PDU. This field shall be represented by an 8-bit unsigned integer.

i) *Number of Sensor Types.* In QRP mode, this field shall specify the number of sensor types employed by the requesting simulation as specified in the Minefield Query PDU. In heartbeat mode, this field shall specify the number of sensor types employed in the exercise. This field shall be represented by an 8-bit unsigned integer.

j) *Data Filter.* This field shall identify which of the optional data fields are contained in this Minefield Data PDU. The data filter field shall be represented as a 32-bit record (see 6.2.16).

k) *Mine Type.* This field shall identify the type of mine contained in this Minefield Data PDU. This field is represented by an Entity Type record (see 6.2.30).

l) *Sensor Type.* In QRP mode, this field shall specify the requesting sensor type that was specified in the Minefield Query PDU. In heartbeat mode, this field shall specify the sensor type that is being served by the minefield. This field shall be represented by a Minefield Sensor Type record (see 6.2.57).

m) *Mine Location.* This field shall specify the location of the mine, relative to the Minefield Location field, given in the corresponding Minefield State PDU field. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96].

n) *Ground Burial Depth Offset.* This field shall specify the offset of the origin of the mine coordinate system with respect to the ground surface. This offset can be used in conjunction with the mine orientation to determine the actual ground burial depth of the surfaces of a mine. Ground burial depth offset is specified as a positive measurement in meters below the terrain surface along the up vector. Although mine locations are specified in three dimensions in the Mine Location field, the ground burial depth offset field is provided so that the positions of mines relative to the terrain surface are accurately conveyed. Ground burial depth offset provides the frame of reference for the other two burial depth offsets (water and snow). If any of the three burial depth offsets are sent in the PDU, ground burial depth offset and mine orientation shall also be sent. If a terrain database does not include the ground surface (such as under a water feature), an arbitrary ground burial depth offset shall be specified. This field shall be represented by a 32-bit floating point number.

o) *Water Burial Depth Offset.* This field shall specify the offset of the origin of the mine coordinate system with respect to the water surface. This offset can be used in conjunction with the mine orientation to determine the actual water burial depth of the surfaces of a mine. Water burial depth offset is specified as a positive measurement in meters below the water surface along the up vector. Although mine locations are specified in three dimensions in the Mine Location field, the water burial depth offset field is provided so that the positions of mines relative to the water surface are accurately conveyed. Ground burial depth offset provides the frame of reference for this burial depth offset. If any of the three burial depth offsets are sent in the PDU, ground burial depth offset and mine orientation shall also be sent. The value of the water burial depth offset field shall be set to the value of the ground burial depth offset to indicate there is no water for the mine to be buried in. This field shall be represented by a 32-bit floating point number.

p) *Snow Burial Depth Offset.* This field shall specify the offset of the origin of the mine coordinate system with respect to the snow surface. This offset can be used in conjunction with the mine orientation to determine the actual snow burial depth of the surfaces of a mine. Snow burial depth offset is specified as a positive measurement in meters below the snow surface along the up vector. Although mine locations are specified in three dimensions in the Mine Location field, the snow burial depth offset field is provided so that the positions of mines relative to the snow surface are accurately conveyed. Ground burial depth offset provides the frame of reference for this burial depth offset. If any of the three burial depth offsets are sent in the PDU, ground burial depth offset and mine orientation shall also be sent. The value of the snow burial depth offset field shall be set to the value of the ground burial depth offset to indicate there is no snow for the mine to be buried in. This field shall be represented by a 32-bit floating point number.

q) *Mine Orientation.* This field shall specify the orientation of the center axis direction of fire of the mine, relative to the minefield Entity Coordinate System. If any of the three burial depth offsets is sent in the PDU, this field shall also be sent. This field shall be represented by an Euler Angles record (see 6.2.32).

r) *Thermal Contrast.* This field shall specify the temperature difference between the mine and the surrounding soil in degrees Centigrade. In the case of a buried mine, the delta temperature shall be measured between the ground surface above the mine and the surrounding ground surface temperature. This field shall be represented by a 32-bit floating point number.

s) *Reflectance.* This field shall specify the local dielectric difference between the mine and the surrounding soil. This field shall be represented by a 32-bit floating point number.

t) *Mine Emplacement Time.* This field shall specify the simulation time of emplacement of the mine. This field shall be represented as a Clock Time record (see 6.2.14).

u) *Mine Entity ID.* This field shall identify the mine entity and shall be represented by a 16-bit unsigned integer.

NOTE—The Minefield number in conjunction with the Site Number and Application Number values from the Minefield ID forms the unique identifier (per 6.2.28) for each mine.

v) *Fusing.* This field shall specify the primary and secondary fuse and antihandling device for each mine. The fusing field shall be represented as a 16-bit record (see [UID 192]).

w) *Scalar Detection Coefficient.* This field shall specify the coefficient to be utilized for proper correlation between detectors located on different simulation platforms. In statistically based detection system applications, the detection system simulation will generally compare a random number against an internally calculated probability of detection. The scalar detection coefficient, scaled appropriately to a 0 to 1 range, should be compared against the internally calculated probability of detection to determine whether a detection has occurred. If the scalar detection coefficient is equal to or less than the probability of detection, then a detection has occurred. If the scalar detection coefficient is greater than the probability of detection, then a detection has not occurred. Internally generated random numbers should not be used because they will not provide for proper correlation across distributed detection system simulations. Random processes within the detector that are not determined by external environmental factors can be incorporated into the calculation of the appropriate probability of detection prior to its comparison with the scalar detection coefficient. This field shall be represented as an 8-bit unsigned integer. See 5.10.4.5 for the ordering of elements in this field.

x) *Paint Scheme.* This field shall specify the camouflage scheme/color of the mine. The paint scheme field shall be represented as an 8-bit record (see [UID 202]).

y) *Number of Trip/Detonation Wires.* This field shall specify the number of trip/detonation wires of the mine. This field shall be represented by an 8-bit unsigned integer.

z) *Number of Vertices.* This field shall specify the number of vertices for a specific trip/detonation wire of the mine. This field shall be represented by an 8-bit unsigned integer. See 5.10.4.5 for the ordering of elements in this field.

aa) *Vertex.* This field shall specify the location of the trip/detonation wire vertex relative to the Minefield Location field given in the corresponding Minefield State PDU field. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96]. See 5.10.4.5 for the ordering of elements in this field.

The format of the Minefield Data PDU shall be as shown in Table 189.

Table 189—Minefield Data PDU

Field size (bits)	Minefield Data PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 39
		Protocol Family—8-bit enumeration = 8
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Minefield ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Minefield Number—16-bit unsigned integer
48	Requesting Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Minefield Sequence Number	16-bit unsigned integer
8	Request ID	8-bit unsigned integer
8	PDU Sequence Number	8-bit unsigned integer
8	Number of PDUs	8-bit unsigned integer
8	Number of Mines (*N*)	8-bit unsigned integer
8	Number of Sensor Types (*M*)	8-bit unsigned integer
8	Padding	8 bits unused
32	Data Filter	32-bit record
64	Mine Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
16	Sensor Type #1	Minefield Sensor Type record

Table 189—Minefield Data PDU *(continued)*

Field size (bits)	Minefield Data PDU fields	
⋮		
16	Sensor Type #*M*	Minefield Sensor Type record
$8P_1$	Padding	Padding to 32-bit boundary—P_1 octets
96	Mine Location #1	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
⋮		
96	Mine Location #*N*	*x*-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
$32G$	Ground Burial Depth Offset #1	32-bit floating point
⋮		
$32G$	Ground Burial Depth Offset #*N*	32-bit floating point
$32F_1$	Water Burial Depth Offset #1	32-bit floating point
⋮		
$32F_1$	Water Burial Depth Offset #*N*	32-bit floating point
$32F_2$	Snow Burial Depth Offset #1	32-bit floating point
⋮		
$32F_2$	Snow Burial Depth Offset #*N*	32-bit floating point
$96R$	Mine Orientation #1	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
⋮		

Table 189—Minefield Data PDU *(continued)*

Field size (bits)	Minefield Data PDU fields	
96*R*	Mine Orientation #*N*	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
$32F_4$	Thermal Contrast #1	32-bit floating point
	• • •	
$32F_4$	Thermal Contrast #*N*	32-bit floating point
$32F_5$	Reflectance #1	32-bit floating point
	• • •	
$32F_5$	Reflectance #*N*	32-bit floating point
$64F_6$	Mine Emplacement Time #1	Hour—32-bit integer
		Time Past the Hour—32-bit unsigned integer
	• • •	
$64F_6$	Mine Emplacement Time #*N*	Hour—32-bit integer
		Time Past the Hour—32-bit unsigned integer
16	Mine Entity Number #1	16-bit unsigned integer
	• • •	
16	Mine Entity Number #*N*	16-bit unsigned integer
$16F_8$	Fusing #1	16-bit record
	• • •	
$16F_8$	Fusing #*N*	16-bit record
$8F_9$	Scalar Detection Coefficient #1	8-bit unsigned integer
	• • •	
$8F_9$	Scalar Detection Coefficient #*S*	8-bit unsigned integer
$8F_{10}$	Paint Scheme #1	8-bit record

Table 189—Minefield Data PDU ***(continued)***

Field size (bits)	Minefield Data PDU fields	
	• • •	
$8F_{10}$	Paint Scheme #N	8-bit record
$8P_2$	Padding	Padding to 32-bit boundary—P_2 octets
$8F_7$	Number of Trip/Detonation Wires #1 (I_1)	8-bit unsigned integer
	• • •	
$8F_7$	Number of Trip/Detonation Wires #N (I_N)	8-bit unsigned integer
$8P_3$	Padding	Padding to 32-bit boundary—P_3 octets
$8F_7$	Number of Vertices #1 (J_1)	8-bit unsigned integer
	• • •	
$8F_7$	Number of Vertices #U (J_U)	8-bit unsigned integer
$8P_4$	Padding	Padding to 32-bit boundary—P_4 octets
$96F_7$	Vertex #1	x-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point
	• • •	
$96F_7$	Vertex #V	x-component—32-bit floating point
		y-component—32-bit floating point
		z-component—32-bit floating point

Table 189—Minefield Data PDU *(continued)*

Field size (bits)	Minefield Data PDU fields

Total Minefield Data PDU size =

$$352 + 16M + 8P_1 + 96N + 32NG + 32NF_1 + 32NF_2 + 96NR + 32NF_4 + 32NF_5$$
$$+64NF_6 + 16N + 16NF_8 + 8SF_9 + 8NF_{10} + 8P_2 + 8NF_7 + 8P_3 + 8UF_7 + 8P_4 + 96VF_7 \text{ bits}$$

where

N is the number of mines
M is the number of sensor types
S is the number of Scalar Detection Coefficient values in the PDU, which is MN
I_n is the number of trip/detonation wires for mine #n
J_i is the number of vertices for trip/detonation wire #i

U is the number of Number of Vertices values in the PDU, which is $\sum_{n=1}^{N} I_n$

V is the number of Vertex values in the PDU, which is $\sum_{n=1}^{N} \sum_{i=1}^{I_N} J_i$

P_1 is the number of padding octets in the first padding field, which is $2(\lceil M/2 \rceil 2 - M)$

P_2 is the number of padding octets in the second padding field, which is
$\lceil (2N + 2NF_8 + SF_9 + NF_{10})/4 \rceil 4 - (2N + 2NF_8 + SF_9 + NF_{10})$

P_3 is the number of padding octets in the third padding field, which is $\lceil NF_7/4 \rceil 4 - NF_7$

P_4 is the number of padding octets in the fourth padding field, which is $\lceil U/4 \rceil 4 - U$

$$G = \begin{cases} 1 \text{ if any of } F_0, F_1 \text{ or } F_2 \text{ are } 1 \\ 0 \text{ if } F_0, F_1 \text{ and } F_2 \text{ are all } 0 \end{cases}$$

$$R = \begin{cases} 1 \text{ if any of } F_0, F_1, F_2 \text{ or } F_3 \text{ are } 1 \\ 0 \text{ if } F_0, F_1, F_2 \text{ and } F_3 \text{ are all } 0 \end{cases}$$

F_x denotes the value of bit x in the Data Filter field (i.e., either 0 or 1)

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.9.5 Minefield Response Negative Acknowledgment (NACK) PDU

The Minefield Response NACK PDU shall provide the means by which a simulation shall request a minefield simulation to retransmit Minefield Data PDUs not received in response to a query when operating in QRP mode. See 5.10.5 for specific requirements on the use of the Minefield Response NACK PDU. The Minefield Response NACK PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Minefield ID*. This field shall identify the minefield to which this PDU is addressed. This field shall be represented by a Minefield Identifier record (see 6.2.56).

c) *Requesting Entity ID.* This field shall identify the simulation that generated the query and is requesting retransmission of information from the minefield simulation. This field shall be represented by a Simulation Identifier record (see 6.2.81).

d) *Request ID.* This field shall identify the minefield query request and shall be represented by an 8-bit unsigned integer.

e) *Number of Missing PDUs.* This field shall identify the number of PDUs that were not received by the requesting simulation in response to a minefield query request. This value is derived from the Minefield Data PDUs that were received in response to the minefield query request. This field shall be represented by an 8-bit unsigned integer.

f) *Missing PDU Sequence Numbers.* This field shall identify the sequence numbers of the missing PDUs. These values are derived from the Minefield Data PDUs that were received in response to the minefield query request. This field shall be represented by an 8-bit unsigned integer.

The format of the Minefield Response NACK PDU shall be as shown in Table 190.

Table 190—Minefield Response NACK PDU

Field size (bits)	Minefield Response NACK PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 40
		Protocol Family—8-bit enumeration = 8
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Minefield ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Minefield Number—16-bit unsigned integer
48	Requesting Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
8	Request ID	8-bit unsigned integer
8	Number of Missing PDUs (*N*)	8-bit unsigned integer
8	Missing PDU Sequence Number #1	8-bit unsigned integer
	⋮	
8	Missing PDU Sequence Number #*N*	8-bit unsigned integer
8*P*	Padding	Padding to 32-bit boundary—*P* octets

Table 190—Minefield Response NACK PDU *(continued)*

Field size (bits)	Minefield Response NACK PDU fields
Total Minefield Response NACK PDU size = $208 + 8N + 8P$ bits where N is the number of missing PDUs P is the number of padding octets, which is $\lceil (N+2)/4 \rceil 4 - (N+2)$ $\lceil x \rceil$ is the largest integer $< x + 1$.	

7.10 Synthetic Environment protocol family

7.10.1 General

The Synthetic Environment protocol family is a family of PDUs for distributing information about environmental conditions with three classes of PDUs: the Environmental Process PDU, Gridded Data PDU, and environment object PDUs. These PDUs are described in 7.10.2 through 7.10.6.

7.10.2 Environmental Process PDU

Information about environmental effects and processes shall be communicated using an Environmental Process PDU. See 5.11.2.2 for specific requirements on the use of the Environmental Process PDU. The Environmental Process PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Environmental Process ID.* This field shall identify the environmental process issuing the Environmental Process PDU. This field shall be represented by an Object Identifier record (see 6.2.63).

c) *Environment Type.* This field shall identify the type of environmental effect being described. This field shall be represented by an Entity Type record (see 6.2.30).

d) *Model Type.* This field shall specify the particular model being used to generate this environmental condition or entity. This field shall be represented by an 8-bit enumeration (see [UID 248]).

e) *Environment Status.* This field shall identify the status of the environmental process. This field shall be represented by an 8-bit record (see [UID 249]).

f) *Number of Environment Records.* This field shall specify the number of Environment records contained in the variable portion of this PDU. This field shall be represented by a 16-bit unsigned integer.

g) *Sequence Number.* This field shall indicate the PDU sequence number for the environmental process (see 5.11.2.2.4). This field shall be specified by a 16-bit unsigned integer.

h) *Environment record.* This field shall contain specific geometry, state, or bounding sphere information and shall be represented by an Environment record (see 6.2.31).

The format of the Environmental Process PDU shall be as shown in Table 191.

Table 191—Environmental Process PDU

Field size (bits)	Environmental Process PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 41
		Protocol Family—8-bit enumeration = 9
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Environmental Process ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Object Number—16-bit unsigned integer
64	Environment Type	Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
8	Model Type	8-bit enumeration
8	Environment Status	8-bit record
16	Number of Environment Records (M)	16-bit unsigned integer
16	Sequence Number	16-bit unsigned integer
$64 + K_1 + P_1$	Environment record #1	Type—32-bit enumeration
		Length—16-bit unsigned integer ($64 + K_1 + P_1$)
		Index—8-bit unsigned integer
		Padding—8 bits unused
		State or Geometry record (K_1 bits)
		Padding to 64-bit boundary—P_1 bits
⋮		

Table 191—Environmental Process PDU *(continued)*

Field size (bits)	Environmental Process PDU fields	
$64 + K_M + P_M$	Environment record #*M*	Type—32-bit enumeration
		Length—16-bit unsigned integer $(64 + K_M + P_M)$
		Index—8-bit unsigned integer
		Padding—8 bits unused
		State or Geometry record (K_M bits)
		Padding to 64-bit boundary—P_M bits

$$\text{Total Environmental Process PDU size} = 256 + \sum_{i=1}^{M} (64 + K_i + P_i) \text{ bits}$$

where

M is the number of environment records
K_i is the length of *i*th geometry or state record in bits
P_i is the number of padding bits in Environment Record *i*, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.10.3 Gridded Data PDU

Information about global, spatially varying environmental effects shall be communicated using one or more Gridded Data PDUs. See 5.11.2.3 for specific requirements on the use of the Gridded Data PDU. The Gridded Data PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Environmental Simulation ID.* This field shall identify the environmental simulation application issuing the PDU. This field shall be represented by a Simulation Identifier record (see 6.2.81).

c) *Field Number.* This field shall specify a unique identifier for each environmental variable transmitted during an exercise. This field shall be represented by a 16-bit unsigned integer.

d) *PDU Number.* This field shall specify an index number of the current PDU within the total number of PDUs used to transmit the environmental data. This field shall be represented by a 16-bit unsigned integer.

e) *PDU Total.* This field shall specify the total number of PDUs used to transmit the environmental data. This field shall be represented by a 16-bit unsigned integer.

f) *Coordinate System.* This field shall specify the coordinate system of the grid for the environmental data contained in the PDU. This field shall be represented by a 16-bit enumeration (see [UID 244]).

g) *Number of Grid Axes.* This field shall specify the number of grid axes for the environmental data contained in the PDU (e.g., three grid axes for an *x*, *y*, *z* coordinate system). This field shall be represented by an 8-bit unsigned integer.

h) *Constant Grid.* This field shall specify whether the domain grid axes are identical to those of the previous domain update grid for the environmental data sample contained in the PDU. This field shall be represented by an 8-bit enumeration (see [UID 245]).

i) *Environment Type.* This field shall identify the type of environmental entity being described. This field shall be represented by an Entity Type record (see 6.2.30).

j) *Orientation.* This field shall specify the orientation of the data grid and shall be represented by an Euler Angles record (6.2.32).

k) *Sample Time.* This field shall specify the valid simulation time of the environmental data sample contained in the PDU. This field shall be represented by a Clock Time record.

l) *Total Values.* This field shall specify the total number of data values for all PDUs for an environmental sample. This total includes vector-valued environmental data and equals the product of the vector dimension and the total number of grid points. This field shall be represented by a 32-bit unsigned integer.

m) *Vector Dimension (V).* This field shall specify the total number of data values at each grid point and accommodates scalar- or vector-valued environmental data. Vector Dimension shall be one for scalar data and shall be greater than one when multiple enumerated environmental data values are sent for each grid location (e.g., u, v, w wind components have $V = 3$). This field shall be represented by an 8-bit unsigned integer.

n) *Grid Axis Descriptor.* This field shall specify the detailed information about the grid dimensions (axes) and coordinates for environmental state variables. This field shall be represented by a Grid Axis Descriptor record(s) (see 6.2.40).

o) *Grid Data.* This field shall specify the environmental state data at the grid locations specified by parameters in the Grid Axis Descriptor record. More than one Grid Data record is allowed in a single Gridded Data PDU corresponding to each enumerated sample type. This field shall be represented by a Grid Data record(s) (see 6.2.41).

The format of the Gridded Data PDU shall be as shown in Table 192.

Table 192—Gridded Data PDU

Field size (bits)	Gridded Data PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 42
		Protocol Family—8-bit enumeration = 9
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Environmental Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Field Number	16-bit unsigned integer
16	PDU Number	16-bit unsigned integer
16	PDU Total	16-bit unsigned integer
16	Coordinate System	16-bit enumeration
8	Number of Grid Axes (N)	8-bit unsigned integer

Table 192—Gridded Data PDU *(continued)*

Field size (bits)	Gridded Data PDU fields	
8	Constant Grid	8-bit enumeration
64	Environment Type	Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
96	Orientation	Psi (ψ)—32-bit floating point
		Theta (θ)—32-bit floating point
		Phi (ϕ)—32-bit floating point
64	Sample Time	64-bit unsigned integer
32	Total Values (T)	32-bit unsigned integer
8	Vector Dimension (*V*)	8-bit unsigned integer
8	Padding	8 bits unused
16	Padding	16 bits unused
$8K_1$	Grid Axis Descriptor #1	Grid Axis Descriptor record—K_1 octets
	• • •	
$8K_N$	Grid Axis Descriptor #*N*	Grid Axis Descriptor record—K_N octets
$8D_1$	Grid Data #1	Grid Data record—D_1 octets
	• • •	
$8D_G$	Grid Data #*G*	Grid Data record—D_G octets

Table 192—Gridded Data PDU *(continued)*

Field size (bits)	Gridded Data PDU fields
	Total Gridded Data PDU size = $512 + \sum_{i=1}^{N} 8K_i + \sum_{j=1}^{G} 8D_j$ bits where N is the number of grid axes K_i is the length of Grid Axis Descriptor i in octets D_j is the length of Grid Data record j in octets G is the number of grid points in the PDU, which is $V \prod_{k=1}^{N} N_k$ V is the vector dimension N_k is the number of points in Axis k

7.10.4 Point Object State PDU

Detailed information about the addition/modification of a synthetic environment object that is geometrically anchored to the terrain with a single point shall be communicated by issuing a Point Object State PDU. See 5.11.3.2 for specific requirements on the use of the Point Object State PDU. The Point Object State PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by a PDU Header record (see 6.2.66).

b) *Object ID*. This field shall specify the unique identification of the object in the synthetic environment. The field shall be represented by an Object Identifier record (see 6.2.63).

c) *Referenced Object ID*. This field shall identify the synthetic environment object with which this point object is associated. The field shall be represented by an Object Identifier record (see 6.2.63).

d) *Update Number*. This field shall represent the unique update number, starting with 1, of each state transition of an individual object. This field shall be represented by a 16-bit unsigned integer.

e) *Force ID*. This field shall identify the force that created or modified the object. This field shall be represented by an 8-bit enumeration (see [UID 6]).

f) *Modifications*. This field shall identify whether a modification has been made to the point object's location or orientation. This field shall be represented by an 8-bit enumeration (see [UID 240]).

g) *Object Type*. This field shall identify the type of synthetic environment object. This field shall be represented by an Object Type record (see 6.2.64).

h) *Object Location*. This field shall specify the object's physical location in the simulated world. This field shall be represented by a World Coordinates record (see 6.2.98).

i) *Object Orientation*. This field shall specify an object's orientation. This field shall be represented by a Euler Angles record (see 6.2.32).

j) *Specific Object Appearance*. This field shall specify specific dynamic changes to an object's appearance attributes. This record shall be defined as a 32-bit record.

k) *General Object Appearance*. This field shall specify general dynamic changes to an object's appearance attributes. This record shall be defined as a 16-bit record (see [UID 229]).

l) *Requester Simulation ID*. This field shall specify the simulation application that is sending or has sent the Point Object State PDU to the Environment Manager. This field shall be represented by a Simulation Address record (see 6.2.80).

m) *Receiving Simulation ID*. This field shall specify the simulation application that is to receive the Point Object State PDU. This field shall be represented by a Simulation Address record (see 6.2.80).

The format of the Point Object State PDU shall be as shown in Table 193.

Table 193—Point Object State PDU

Field size (bits)	Point Object State PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 43
		Protocol Family—8-bit enumeration = 9
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Object Number—16-bit unsigned integer
48	Referenced Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Object Number—16-bit unsigned integer
16	Update Number	16-bit unsigned integer
8	Force ID	8-bit enumeration
8	Modifications	8-bit enumeration
32	Object Type	Domain—8-bit enumeration
		Object Kind—8-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
192	Object Location	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point

Table 193—Point Object State PDU *(continued)*

Field size (bits)	Point Object State PDU fields	
96	Object Orientation	Psi (ψ)—32-bit floating point
		Theta (θ) —32-bit floating point
		Phi (ϕ)—32-bit floating point
32	Specific Object Appearance	32-bit record
16	General Object Appearance	16-bit record
16	Padding	16 bits unused
32	Requester Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
32	Receiving Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
32	Padding	32 bits unused
Total Point Object State PDU size = 704 bits		

7.10.5 Linear Object State PDU

Detailed information about the addition/modification of a synthetic environment object that is geometrically anchored to the terrain with one point and has size and orientation shall be communicated by issuing a Linear Object State PDU. See 5.11.3.3 for specific requirements on the use of the Linear Object State PDU. The Linear Object State PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by a PDU Header record (see 6.2.66).

b) *Object ID*. This field shall specify the unique identification of the object in the synthetic environment. The field shall be represented by an Object Identifier record (see 6.2.63).

c) *Referenced Object ID*. This field shall identify the synthetic environment object with which this linear object is associated. The field shall be represented by an Object Identifier record (see 6.2.63).

d) *Update Number*. This field shall represent the unique update number, starting with 1, of each state transition of an individual object. This field shall be represented by a 16-bit unsigned integer.

e) *Force ID*. This field shall identify the force that created or modified the object. This field shall be represented by an 8-bit enumeration (see [UID 6]).

f) *Number of Linear Segments*. This field shall specify the number of Linear Segment Parameter records required for the specification of the linear object. This field shall be represented by an 8-bit unsigned integer.

g) *Requester Simulation ID*. This field shall specify the simulation application that is sending or has sent the Linear Object State PDU to the Environment Manager. This field shall be represented by a Simulation Address record (see 6.2.80).

h) *Receiving Simulation ID*. This field shall specify the simulation application that is to receive the Linear Object State PDU. This field shall be represented by a Simulation Address record (see 6.2.80).

i) *Object Type*. This field shall identify the type of synthetic environment object. This field shall be represented by an Object Type record (see 6.2.64).

j) *Linear Segment Parameters*. This field shall specify the parameter values for representing each linear segment. Each linear segment shall be represented by a Linear Segment Parameter record (see 6.2.52).

The format of the Linear Object State PDU shall be as shown in Table 194.

Table 194—Linear Object State PDU

Field size (bits)	Linear Object State PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 44
		Protocol Family—8-bit enumeration = 9
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Object Number—16-bit unsigned integer
48	Referenced Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Object Number—16-bit unsigned integer
16	Update Number	16-bit unsigned integer
8	Force ID	8-bit enumeration
8	Number of Linear Segments (*N*)	8-bit unsigned integer
32	Requester Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
32	Receiving Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
32	Object Type	Domain—8-bit enumeration
		Object Kind—8-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
512	Linear Segment Parameters #1	Linear Segment Parameter record
	⋮	

Table 194—Linear Object State PDU ***(continued)***

Field size (bits)	Linear Object State PDU fields	
512	Linear Segment Parameters #*N*	Linear Segment Parameter record
Total Linear Object State PDU size = 320 + 512*N* bits where *N* is the number of linear segments		

7.10.6 Areal Object State PDU

Detailed information about the addition/modification of a synthetic environment object that is geometrically anchored to the terrain with a set of three or more points that come to a closure shall be communicated by issuing a Areal Object State PDU. See 5.11.3.4 for specific requirements on the use of the Areal Object State PDU. The Areal Object State PDU shall contain the following fields:

a) *PDU Header.* This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by a PDU Header record (see 6.2.66).

b) *Object ID.* This field shall specify the unique identification of the object in the synthetic environment. The field shall be represented by an Object Identifier record (see 6.2.63).

c) *Referenced Object ID.* This field shall identify the synthetic environment object with which this areal object is associated. The field shall be represented by an Object Identifier record (see 6.2.63).

d) *Update Number.* This field shall represent the unique update number, starting with one, of each state transition of an individual object. This field shall be represented by a 16-bit unsigned integer.

e) *Force ID.* This field shall identify the force that created or modified the object. This field shall be represented by an 8-bit enumeration (see [UID 6]).

f) *Modifications.* This field shall identify whether a modification has been made to the areal object's location or orientation. This field shall be represented by an 8-bit enumeration (see [UID 242]).

g) *Object Type.* This field shall identify the type of synthetic environment object. This field shall be represented by an Object Type record (see 6.2.64).

h) *Specific Object Appearance.* This field shall specify specific dynamic changes to an object's appearance attributes. This record shall be defined as a 32-bit record.

i) *General Object Appearance.* This field shall specify general dynamic changes to an object's appearance attributes. This record shall be defined as a 16-bit record (see [UID 229]).

j) *Number of Points.* This field shall specify the total number of points making up the areal object. This field shall be represented by a 16-bit unsigned integer.

k) *Requester Simulation ID.* This field shall specify the simulation application that is sending or has sent the Areal Object State PDU to the Environment Manager. This field shall be represented by a Simulation Address record (see 6.2.80).

l) *Receiving Simulation ID.* This field shall specify the simulation application that is to receive the Areal Object State PDU. This field shall be represented by a Simulation Address record (see 6.2.80).

m) *Points.* This field shall specify the object's physical location in the simulated world. This field shall be represented by a World Coordinates record (see 6.2.98).

The format of the Areal Object State PDU shall be as shown in Table 195.

Table 195—Areal Object State PDU

Field size (bits)	Areal Object State PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 45
		Protocol Family—8-bit enumeration = 9
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Object Number—16-bit unsigned integer
48	Referenced Object ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Object Number—16-bit unsigned integer
16	Update Number	16-bit unsigned integer
8	Force ID	8-bit enumeration
8	Modifications	8-bit enumeration
32	Object Type	Domain—8-bit enumeration
		Object Kind—8-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
32	Specific Object Appearance	32-bit record
16	General Object Appearance	16-bit record
16	Number of Points (*N*)	16-bit unsigned integer
32	Requester Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
32	Receiving Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
192	Point #1	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point

Table 195—Areal Object State PDU *(continued)*

Field size (bits)	Areal Object State PDU fields	
• • •		
192	Point #*N*	*X*-component—64-bit floating point
		Y-component—64-bit floating point
		Z-component—64-bit floating point
Total Areal Object State PDU size = 384 + 192*N* bits where *N* is the number of points defining the areal object		

7.11 Simulation Management with Reliability protocol family

7.11.1 General

The PDUs of the Simulation Management with Reliability protocol family are described in 7.11.2 through 7.11.16.

7.11.2 Create Entity-R PDU

The creation of a new entity shall be communicated using a Create Entity-R PDU. See 5.12.4.2 for specific requirements on the use of the Create Entity-R PDU. The Create Entity-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

c) *Request ID.* This field shall identify the entity creation request being made by the SM. This field shall be represented by a 32-bit unsigned integer (see 6.2.75).

The format of the Create Entity-R PDU shall be as shown in Table 196.

Table 196—Create Entity-R PDU

Field size (bits)	Create Entity-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 51
		Protocol Family—8-bit enumeration 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Request ID	32-bit unsigned integer
Total Create Entity-R PDU size = 256 bits		

7.11.3 Remove Entity-R PDU

The removal of an entity from an exercise shall be communicated with a Remove Entity-R PDU. See 5.12.4.3 for specific requirements on the use of the Remove Entity-R PDU. The Remove Entity-R PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

c) *Request ID.* This field shall identify the specific and unique entity removal request being made by the SM. This field shall be represented by a 32-bit unsigned integer.

The format of the Remove Entity-R PDU shall be as shown in Table 197.

Table 197—Remove Entity-R PDU

Field size (bits)	Remove Entity-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 52
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Request ID	32-bit unsigned integer
Total Remove Entity-R PDU size = 256 bits		

7.11.4 Start/Resume-R PDU

The starting or resuming of an entity/exercise shall be communicated using a Start/Resume-R PDU. See 5.12.4.4 for specific requirements on the use of the Start/Resume-R PDU. The Start/Resume-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Real-World Time.* This field shall specify the real-world time at which the entity is to start/resume in the exercise. This information shall be used by the participating simulation applications to start/resume an exercise synchronously. This field shall be represented by a Clock Time record (see 6.2.14).

c) *Simulation Time.* This field shall specify the simulation time (time of day in the simulated world) at which the entity will start/resume in the exercise. This field shall be represented by a Clock Time record (see 6.2.14).

d) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

e) *Request ID.* This field shall identify the specific and unique start/resume request being made by the SM. This field shall be represented by a 32-bit unsigned integer.

The format of the Start/Resume-R PDU shall be as shown in Table 198.

Table 198—Start/Resume-R PDU

Field size (bits)	Start/Resume-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 53
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
64	Real-World Time	Hour—32-bit integer
		Time Past Hour—32-bit unsigned integer
64	Simulation Time	Hour—32-bit integer
		Time Past Hour—32-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Request ID	32-bit unsigned integer
Total Start/Resume-R PDU size = 384 bits		

7.11.5 Stop/Freeze-R PDU

The stopping or freezing of an entity/exercise shall be communicated using a Stop/Freeze-R PDU. See 5.12.4.5 for specific requirements on the use of the Stop/Freeze-R PDU. The Stop/Freeze-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Real-World Time.* This field shall specify the real-world time at which the entity is to stop/freeze in the exercise. This field shall be represented by a Clock Time record (see 6.2.14).

c) *Reason.* This field shall specify the reason that an entity or exercise was stopped/frozen and shall be represented by an 8-bit enumeration (see [UID 67]).

d) *Frozen Behavior.* This field shall specify the internal behavior of the simulation and its appearance while frozen to the other participants in the exercise and shall be represented by an 8-bit record (see [UID 68]).

e) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

f) *Request ID.* This field shall identify the specific and unique stop/freeze request being made by the SM. This field shall be represented by a 32-bit unsigned integer.

The format of the Stop/Freeze-R PDU shall be as shown in Table 199.

Table 199—Stop/Freeze-R PDU

Field size (bits)	Stop/Freeze-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 54
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
64	Real-World Time	Hour—32-bit integer
		Time Past Hour—32-bit unsigned integer
8	Reason	8-bit enumeration

Table 199—Stop/Freeze-R PDU *(continued)*

Field size (bits)	Stop/Freeze-R PDU fields	
8	Frozen Behavior	8-bit record
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
32	Request ID	32-bit unsigned integer
Total Stop/Freeze-R PDU size = 320 bits		

7.11.6 Acknowledge-R PDU

The acknowledgment of the receipt of a Start/Resume-R PDU, Stop/Freeze-R PDU, Create Entity-R PDU, or a Remove Entity-R PDU shall be communicated by issuing an Acknowledge-R PDU if the PDU being acknowledged indicates that the acknowledge reliability service shall be used for this transaction. See 5.12.4.6 for specific requirements on the use of the Acknowledge-R PDU. The Acknowledge-R PDU shall contain the following fields:

a) *Simulation Management PDU Header*. The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Acknowledge Flag*. This field shall indicate what type of message has been acknowledged. This field shall be represented by a 16-bit enumeration (see [UID 69]).

c) *Response Flag*. This field shall indicate whether the receiving entity was able to comply with the request and shall be represented by a 16-bit enumeration (see [UID 70]).

d) *Request ID*. This field shall identify the matching response to the specific Start/Resume-R, Stop/Freeze-R, Create Entity-R, or Remove Entity-R PDU sent by the SM. This field shall be represented by a 32-bit unsigned integer.

The format of the Acknowledge-R PDU shall be as shown in Table 200.

Table 200—Acknowledge-R PDU

Field size (bits)	Acknowledge-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 55
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused

Table 200—Acknowledge-R PDU *(continued)*

Field size (bits)	Acknowledge-R PDU fields	
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	Acknowledge Flag	16-bit enumeration
16	Response Flag	16-bit enumeration
32	Request ID	32-bit unsigned integer
Total Acknowledge-R PDU size = 256 bits		

7.11.7 Action Request-R PDU

A request from an SM to a managed entity to perform a specified action shall be communicated using an Action Request-R PDU. See 5.12.4.7 for specific requirements on the use of the Action Request-R PDU. The Action Request-R PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

c) *Request ID.* This field shall identify the request being made by the SM and shall be represented by a 32-bit unsigned integer.

d) *Action ID.* This field shall specify the particular action that is requested by the SM and shall be represented by a 32-bit enumeration (see [UID 71]).

e) *Datum Information.* This field shall specify the types of datum and the value of the datum to be communicated and shall be represented by a Datum Specification record (see 6.2.18).

The format of the Action Request-R PDU shall be as shown in Table 201.

Table 201—Action Request-R PDU

Field size (bits)	Action Request-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 56
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Request ID	32-bit unsigned integer
32	Action ID	32-bit enumeration
32	Padding	32 bits unused
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum #*N*	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits

Table 201—Action Request-R PDU *(continued)*

Field size (bits)	Action Request-R PDU fields	
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
⋮		
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

$$\text{Total Action Request-R PDU size} = 384 + 64N + \sum_{i=1}^{M} (64 + K_i + P_i) \text{ bits}$$

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.11.8 Action Response-R PDU

When an entity receives an Action Request-R PDU indicating that the acknowledge reliability service shall be used for this transaction, the entity shall acknowledge the receipt of the Action Request-R PDU. See 5.12.4.8 for specific requirements on the use of the Action Response-R PDU. The Action Response-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the matching response to a request made by the SM. This field shall be represented by a 32-bit unsigned integer.

c) *Request Status.* This field shall identify the status of the requested action and shall be represented by a 32-bit enumeration (see [UID 72]).

d) *Datum Information.* This field shall specify the types of datum and the value of the datum to be communicated and shall be represented by a Datum Specification record (see 6.2.18).

The format of the Action Response-R PDU shall be as shown in Table 202.

Table 202—Action Response-R PDU

Field size (bits)	Action Response-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 57
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
32	Request Status	32-bit enumeration
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum #*N*	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
	• • •	

Table 202—Action Response-R PDU *(continued)*

Field size (bits)	Action Response-R PDU fields	
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

$$\text{Total Action Response-R PDU size} = 320 + 64N + \sum_{i=1}^{M}(64 + K_i + P_i) \text{ bits}$$

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i/64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.11.9 Data Query-R PDU

A request for data from an entity shall be communicated by issuing a Data Query-R PDU. See 5.12.4.9 for specific requirements on the use of the Data Query-R PDU. The Data Query-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

c) *Request ID.* This field shall identify the data query request being made by the SM and shall be represented by a 32-bit unsigned integer.

d) *Time Interval.* This field shall specify the time interval between issues of Data-R PDUs (see 5.12.4.9.3). This field shall be represented by a Timestamp (see 6.2.88).

e) *Datum Information.* This record shall specify the types of datum for which information is required. Datum IDs, but not datum values, are listed in the Data Query PDU Datum Specification record (see 6.2.17).

The format of the Data Query-R PDU shall be as shown in Table 203.

Table 203—Data Query-R PDU

Field size (bits)	Data Query-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 58
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Request ID	32-bit unsigned integer
32	Time Interval	32-bit unsigned integer
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
32	Fixed Datum ID #1	32-bit enumeration
	• • •	
32	Fixed Datum ID #*N*	32-bit enumeration
32	Variable Datum ID #1	32-bit enumeration
	• • •	
32	Variable Datum ID #*M*	32-bit enumeration

Table 203—Data Query-R PDU *(continued)*

Field size (bits)	Data Query-R PDU fields
Total Data Query-R PDU size = 352 + 32*N* + 32*M* bits where *N* is the number of fixed datum records *M* is the number of variable datum records	

7.11.10 Set Data-R PDU

Initializing or changing internal state information shall be communicated using a Set Data-R PDU. See 5.12.4.10 for specific requirements on the use of the Set Data-R PDU. The Set Data-R PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

c) *Request ID.* This field shall identify the set data request being made by the SM and shall be represented by a 32-bit unsigned integer.

d) *Datum Information.* This field shall specify the types of datum and their value to be communicated. This field shall be represented by a Datum Specification record (see 6.2.18).

The format of the Set Data-R PDU shall be as shown in Table 204.

Table 204—Set Data-R PDU

Field size (bits)	Set Data-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 59
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer

Table 204—Set Data-R PDU *(continued)*

Field size (bits)	Set Data-R PDU fields	
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Request ID	32-bit unsigned integer
32	Number of Fixed Datum Records (N)	32-bit unsigned integer
32	Number of Variable Datum Records (M)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	• • •	
64	Fixed Datum #N	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_I bits
	• • •	
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

$$\text{Total Set Data-R PDU size} = 320 + 64N + \sum_{i=1}^{M} (64 + K_i + P_i) \text{ bits}$$

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.11.11 Data-R PDU

Information issued in response to a Data Query-R PDU or Set Data-R PDU shall be communicated using a Data-R PDU if the PDU being responded to indicates that the acknowledge reliability service shall be used for this transaction. See 5.12.4.11 for specific requirements on the use of the Data-R PDU. The Data-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the matching response to a Data Query-R PDU or Set Data-R PDU made by the SM and shall be represented by a 32-bit unsigned integer.

c) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

d) *Datum Information.* This field shall specify the types of datum and their value to be communicated. This field shall be represented by a Datum Specification record (see 6.2.18).

The format of the Data-R PDU shall be as shown in Table 205.

Table 205—Data-R PDU

Field size (bits)	Data-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 60
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer

Table 205—Data-R PDU *(continued)*

Field size (bits)	Data-R PDU fields	
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
⋮		
64	Fixed Datum #N	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
⋮		
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Data-R PDU size = $320 + 64N + \sum_{i=1}^{M}(64 + K_i + P_i)$ bits

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i/64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.11.12 Event Report-R PDU

A managed entity shall report the occurrence of a significant event to the SM using an Event Report-R PDU. See 5.12.4.12 for specific requirements on the use of the Event Report-R PDU. The Event Report-R PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Event Type.* This field shall specify the type of event that caused the issue of an Event PDU. This field shall be represented by a 32-bit enumeration (see [UID 73]).

c) *Datum Information.* This field shall specify the types of datum and their value be communicated. This field shall be represented by a Datum Specification record (see 6.2.18).

The format of the Event Report-R PDU shall be as shown in Table 206.

Table 206—Event Report-R PDU

Field size (bits)	Event Report-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 61
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Event Type	32-bit enumeration
32	Padding	32 bits unused
32	Number of Fixed Datum Records (*N*)	32-bit unsigned integer
32	Number of Variable Datum Records (*M*)	32-bit unsigned integer
64	Fixed Datum #1	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits
	⋮	
64	Fixed Datum #*N*	Fixed Datum ID—32-bit enumeration
		Fixed Datum Value—32 bits

Table 206—Event Report-R PDU *(continued)*

Field size (bits)	Event Report-R PDU fields	
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
⋮		
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Total Event Report-R PDU size = $320 + 64N + \sum_{i=1}^{M} (64 + K_i + P_i)$ bits

where

N is the number of fixed datum records
M is the number of variable datum records
K_i is the length of variable datum value i in bits
P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i / 64 \rceil 64 - K_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.11.13 Comment-R PDU

Arbitrary messages (character strings, for example) shall be entered into the data stream by using a Comment-R PDU. See 5.12.4.13 for specific requirements on the use of the Comment-R PDU. The Comment-R PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Number of Fixed Datum Records.* This field shall specify the number of Fixed Datum records in the Comment-R PDU and shall be represented by a 32-bit unsigned integer. The value of this field shall be set to zero.

c) *Number of Variable Datum Records.* This field shall specify the number of Variable Datum records required to supply database names or character fields that exceed 32 bits and shall be represented by a 32-bit unsigned integer.

d) *Variable Datum records.* These fields shall specify the types of Variable Datum, their length, and their value. This field shall be represented by a Variable Datum record (see 6.2.93).

The format of the Comment-R PDU shall be as shown in Table 207.

Table 207—Comment-R PDU

Field size (bits)	Comment-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration= 62
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Number of Fixed Datum Records	32-bit unsigned integer
32	Number of Variable Datum Records (M)	32-bit unsigned integer
$64 + K_1 + P_1$	Variable Datum #1	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_1)
		Variable Datum Value—K_1 bits
		Padding to 64-bit boundary—P_1 bits
⋮		
$64 + K_M + P_M$	Variable Datum #M	Variable Datum ID—32-bit enumeration
		Variable Datum Length—32-bit unsigned integer (K_M)
		Variable Datum Value—K_M bits
		Padding to 64-bit boundary—P_M bits

Table 207—Comment-R PDU *(continued)*

Field size (bits)	Comment-R PDU fields
Total Comment-R PDU size = $256 + \sum_{i=1}^{M}(64 + K_i + P_i)$ bits where M is the number of variable datum records K_i is the length of variable datum value i in bits P_i is the number of padding bits in Variable Datum record i, which is $\lceil K_i/64 \rceil 64 - K_i$ $\lceil x \rceil$ is the largest integer $< x + 1$.	

7.11.14 Record Query-R PDU

A request for one or more records of data from an entity shall be communicated by issuing a Record Query-R PDU. See 5.12.4.14 for specific requirements on the use of the Record Query-R PDU. The Record Query-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the record query request being made by the SM. This field shall be represented by a 32-bit unsigned integer.

c) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

d) *Event Type.* This field shall specify the type of event that caused the issue of a Record-R PDU (see 5.12.4.14.3). This field shall be represented by a 16-bit enumeration (see [UID 334]).

e) *Time.* This field shall specify the time interval between issues of Record-R PDUs (see 5.12.4.14.3). This field shall be represented by a Timestamp (see 6.2.88).

f) *Record Information.* This record shall specify the identification of the records for which information is requested and shall be represented by a Record Query Specification record (see 6.2.72).

The format of the Record Query-R PDU shall be as shown in Table 208.

Table 208—Record Query-R PDU

Field size (bits)	Record Query-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 65
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Event Type	16-bit enumeration
32	Time	32-bit unsigned integer
32	Number of Records (N)	32-bit unsigned integer
32	Record ID #1	32-bit enumeration
• • •		
32	Record ID #N	32-bit enumeration
Total Record Query-R PDU size = 320 + 32N bits where N is the number of records		

7.11.15 Set Record-R PDU

Initializing or changing internal parameter information shall be communicated using a Set Record-R PDU. See 5.12.4.15 for specific requirements on the use of the Set Record-R PDU. The Set Record-R PDU shall consist of the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the record set request being made by the SM. This field shall be represented by a 32-bit unsigned integer.

c) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

d) *Record Information.* This field shall specify the number of record sets, the types of records, and the parameter values to be communicated. This field shall be represented by a Record Specification record (see 6.2.73).

The format of the Set Record-R PDU shall be as shown in Table 209.

Table 209—Set Record-R PDU

Field size (bits)	Set Record-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type —8-bit enumeration = 64
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Padding	32 bits unused
32	Number of Record Sets (*R*)	32-bit unsigned integer

Table 209—Set Record-R PDU *(continued)*

Field size (bits)	Set Record-R PDU fields	
$128 + L_1Q_1 + P_1$	Record Set #1	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_1)
		Record Count—16-bit unsigned integer (Q_1)
		Record Values—(L_1Q_1) bits
		Padding to 64-bit boundary—P_1 bits
• • •		
$128 + L_RQ_R + P_R$	Record Set #*R*	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_R)
		Record Count—16-bit unsigned integer (Q_R)
		Record Values—(L_RQ_R) bits
		Padding to 64-bit boundary—P_R bits

Total Set Record-R PDU size = $320 + \sum_{i=1}^{R} (128 + L_iQ_i + P_i)$ bits

where

R is the number of record sets
L_i is the length of the record in Record Set i in bits
Q_i is the number of records included in Record Set i
P_i is the number of padding bits in Record Set i, which is $\lceil L_iQ_i/64 \rceil 64 - L_iQ_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.11.16 Record-R PDU

Information issued in response to a Record Query-R PDU or Set Record-R PDU shall be communicated using a Record-R PDU. See 5.12.4.16 for specific requirements on the use of the Record-R PDU. The Record-R PDU shall contain the following fields:

a) *Simulation Management PDU Header.* The Simulation Management PDU Header shall be represented by the Simulation Management PDU Header record. It consists of the PDU Header, the Originating ID, and the Receiving ID (see 6.2.82).

b) *Request ID.* This field shall identify the record query request being made by the SM. This field shall be represented by a 32-bit unsigned integer.

c) *Required Reliability Service.* This field shall identify the level of reliability service to be used for this transaction (see 5.12.3). This field shall be represented by an 8-bit enumeration (see [UID 74]).

d) *Event Type.* This field shall identify the type of event reported and shall be represented by a 16-bit enumeration (see [UID 333]).

e) *Response Serial Number.* This field shall be used to indicate the serial number of the Record-R PDU when more than one PDU is used to report record values. This field shall be represented by a 32-bit unsigned integer.

f) *Record Information.* This field shall specify the number of record sets, the types of records, and the parameter values to be communicated. This field shall be represented by a Record Specification record (see 6.2.73).

The format of the Record-R PDU shall be as shown in Table 210.

Table 210—Record-R PDU

Field size (bits)	Record-R PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type —8-bit enumeration = 63
		Protocol Family—8-bit enumeration = 10
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
48	Receiving ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
8	Required Reliability Service	8-bit enumeration
8	Padding	8 bits unused
16	Event Type	16-bit enumeration
32	Response Serial Number	32-bit unsigned integer
32	Number of Record Sets (*R*)	32-bit unsigned integer

Table 210—Record-R PDU *(continued)*

Field size (bits)	Record-R PDU fields	
$128 + L_1Q_1 + P_1$	Record Set #1	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_1)
		Record Count—16-bit unsigned integer (Q_1)
		Record Values—(L_1Q_1) bits
		Padding to 64-bit boundary—P_1 bits
• • •		
$128 + L_RQ_R + P_R$	Record Set #*R*	Record ID—32-bit enumeration
		Record Set Serial Number—32-bit unsigned integer
		Padding—32 bits unused
		Record Length—16-bit unsigned integer (L_R)
		Record Count—16-bit unsigned integer (Q_R)
		Record Values—(L_RQ_R) bits
		Padding to 64-bit boundary—P_R bits

Total Record-R PDU size = $320 + \sum_{i=1}^{R} (128 + L_iQ_i + P_i)$ bits

where

R is the number of Record Sets
L_i is the length of the record in Record Set i in bits
Q_i is the number of records included in Record Set i
P_i is the number of padding bits in Record Set i, which is $\lceil L_iQ_i/64 \rceil 64 - L_iQ_i$

$\lceil x \rceil$ is the largest integer $< x + 1$.

7.12 Information Operations protocol family

7.12.1 General

The PDUs of the Information Operations protocol family are described in 7.12.2 through 7.12.3.

7.12.2 Information Operations (IO) Action PDU

Actions initiated by an IO simulation to support interactions with other IO simulations shall be communicated using the IO Action PDU.

See 5.13.3 for specific requirements on the use of the IO Action PDU. The IO Action PDU shall contain the following fields:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Originating Simulation ID*. This field shall identify the simulation that is issuing the PDU and shall be represented by a Simulation Identifier record (see 6.2.81).

c) *Receiving Simulation ID*. This field shall identify the simulation to which this PDU is addressed, if applicable and shall be represented by a Simulation Identifier record (see 6.2.81).

d) *Request ID*. This field shall identify the request number for this IO Action PDU and shall be represented by a 32-bit unsigned integer.

e) *IO Warfare Type*. This field shall identify the type of IO warfare and shall be represented by a 16-bit enumeration (see [UID 285]).

f) *IO Simulation Source*. This field shall identify the name of the simulation model issuing this PDU and shall be represented by a 16-bit enumeration (see [UID 286]).

g) *IO Action Type*. This field shall identify the type of IO action and shall be represented by a 16-bit enumeration (see [UID 287]).

h) *IO Action Phase*. This field shall identify the phase of the IO action and shall be represented by a 16-bit enumeration (see [UID 288]).

i) *IO Attacker Entity ID*. This field shall identify the IO attacker entity and shall be represented by an Entity Identifier record (see 6.2.28).

j) *IO Primary Target Entity ID*. This field shall identify the IO primary target entity and shall be represented by an Entity Identifier record (see 6.2.28).

k) *Number of IO Records*. This field shall specify the number of IO records and shall be represented by a 16-bit unsigned integer.

l) *IO records*. One or more IO records (see 6.2.48) shall be included in the IO Action PDU. Other applicable Standard Variable records may also be included. These records shall conform to the Standard Variable Record format of the Standard Variable Specification record (see 6.2.83).

The format of the IO Action PDU shall be as shown on Table 211.

Table 211—IO Action PDU

Field size (bits)	IO Action PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 70
		Protocol Family—8-bit enumeration = 13
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer

Table 211—IO Action PDU *(continued)*

Field size (bits)	IO Action PDU fields	
48	Receiving Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
32	Request ID	32-bit unsigned integer
16	IO Warfare Type	16-bit enumeration
16	IO Simulation Source	16-bit enumeration
16	IO Action Type	16-bit enumeration
16	IO Action Phase	16-bit enumeration
32	Padding	32 bits unused
48	IO Attacker Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	IO Primary Target Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Padding	16 bits unused
16	Number of IO Records (N)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IO record #1	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer ($6 + K_1 + P_1$)
		Record-Specific fields—K_1 octets
		Padding to 64-bit boundary—P_1 octets
• • •		

Table 211—IO Action PDU *(continued)*

<table>
<tr><th>Field size (bits)</th><th colspan="2">IO Action PDU fields</th></tr>
<tr><td rowspan="4">$48 + 8K_N + 8P_N$</td><td rowspan="4">IO record #N</td><td>Record Type—32-bit enumeration</td></tr>
<tr><td>Record Length—16-bit unsigned integer
$(6 + K_N + P_N)$</td></tr>
<tr><td>Record-Specific fields—K_N octets</td></tr>
<tr><td>Padding to 64-bit boundary—P_N octets</td></tr>
<tr><td colspan="3">Total IO Action PDU size = $448 + 8\sum_{i=1}^{N}(6 + K_i + P_i)$ bits
where
N is the number of IO records
K_i is the length of the Record-Specific field in IO record #i in octets
P_i is the number of padding octets in IO record i, which is $\lceil (6 + K_i)/8 \rceil 8 - (6 + K_i)$
$\lceil x \rceil$ is the largest integer $< x + 1$.</td></tr>
</table>

7.12.3 Information Operations (IO) Report PDU

The information operations status of an entity shall be conveyed using the IO Report PDU. See 5.13.4 for specific requirements on the use of the IO Report PDU. This includes reporting the status of any entities and their associated equipments that are susceptible to an IO attack:

a) *PDU Header*. This field shall contain data common to all DIS PDUs. The PDU Header shall be represented by the PDU Header record (see 6.2.66).

b) *Originating Simulation ID*. This field shall identify the simulation that is issuing the PDU and shall be represented by a Simulation Identifier record (see 6.2.81).

c) *IO Simulation Source*. This field shall identify the name of the simulation model issuing this PDU and shall be represented by a 16-bit enumeration (see [UID 286]).

d) *IO Report Type*. This field shall identify the type of IO report and shall be represented by an 8-bit enumeration (see [UID 289]).

e) *IO Attacker Entity ID*. This field shall identify the IO attacker entity, if known and shall be represented by an Entity Identifier record (see 6.2.28).

f) *IO Primary Target Entity ID*. This field shall identify the IO primary target entity and shall be represented by an Entity Identifier record (see 6.2.28).

g) *Number of IO Records*. This field shall specify the number of IO records and shall be represented by a 16-bit unsigned integer.

h) *IO records*. One or more IO records (see 6.2.48) shall be included in the IO Report PDU. Other applicable Standard Variable records may also be included. These records shall conform to the Standard Variable Record format of the Standard Variable Specification record (see 6.2.83).

The format of the IO Report PDU shall be as shown in Table 212.

Table 212—IO Report PDU

Field size (bits)	IO Report PDU fields	
96	PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 71
		Protocol Family—8-bit enumeration = 13
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		PDU Status—8-bit record
		Padding—8 bits unused
48	Originating Simulation ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Reference Number—16-bit unsigned integer
16	IO Simulation Source	16-bit enumeration
8	IO Report Type	8-bit enumeration
8	Padding	8 bits unused
48	IO Attacker Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
48	Primary Target Entity ID	Site Number—16-bit unsigned integer
		Application Number—16-bit unsigned integer
		Entity Number—16-bit unsigned integer
16	Padding	16 bits unused
16	Padding	16 bits unused
16	Number of IO Records (N)	16-bit unsigned integer
$48 + 8K_1 + 8P_1$	IO record #1	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_1 + P_1)$
		Record-Specific fields—K_1 octets
		Padding to 64-bit boundary—P_1 octets
	• • •	

Table 212—IO Report PDU *(continued)*

Field size (bits)	IO Report PDU fields	
$48 + 8K_N + 8P_N$	IO record #N	Record Type—32-bit enumeration
		Record Length—16-bit unsigned integer $(6 + K_N + P_N)$
		Record-Specific fields—K_N octets
		Padding to 64-bit boundary—P_N octets

Total IO Report PDU size = $320 + 8 \sum_{i=1}^{N} (6 + K_i + P_i)$ bits

where

N is the number of IO records
K_i is the length of the Record-Specific field in IO record #i in octets
P_i is the number of padding octets in IO record i, which is $\lceil (6 + K_i) / 8 \rceil 8 - (6 + K_i)$

$\lceil x \rceil$ is the largest integer $< x + 1$.

8 Non-Real-Time protocol

8.1 Introduction

This clause contains the definition of the protocol for a set of six time management services for non-real-time applications within a DIS exercise. This application protocol is built on existing protocols defined in this document.

Time management deals with coordinating the synchronization of the simulation time with absolute timestamps or real-world time. The advancement of simulation time is coordinated by an SM capable of supporting the time management services specified by this protocol through the exchange of Simulation Management with Reliability PDUs (see 5.12) and is constrained by a rate in relation to real-world time. As an example, a rate of 2 means 2 s of simulation time will advance for each 1 s of real-world time. A particular case with a rate of 1 means that simulation time advances at the same rate as real-world time. All references to SM in Clause 8 refer to the specific SM in the exercise that supports the Non-Real-Time protocol time management services.

8.2 Definitions

8.2.1 advance report: Information sent periodically about the capability of an application to apply the time management parameters requested by the SM.

8.2.2 application rate: The maximum rate at which an entity's capabilities are computed by a simulation application.

8.2.3 application time: The actual simulation time in the application.

8.2.4 application timestep: A simulation time delta requested by the application, equal to the next simulation time at which this application needs to be synchronized minus the actual simulation time.

8.2.5 exercise synchronization: The situation when all applications start from a common simulation timestep.

8.2.6 feedback time: The absolute timestamp corresponding to the receipt of a Join Exercise Request.

8.2.7 simulation rate: A common time advancement rate for all participants in an exercise.

8.2.8 simulation timestamp: The timestamp used by all PDUs except the Simulation Management PDUs. The simulation timestamp contains the application time at which the data in the PDU were generated. Simulation Management PDUs shall use an absolute timestamp synchronized with the SM clock.

8.2.9 simulation timestep: The effective time delta applied by the SM for exercise synchronization.

8.2.10 time interval: A slice of simulation time between advance reports.

8.2.11 time latency: Averaged time delay generated by the network for PDU transmissions.

8.2.12 time scheme: A time management algorithm applied by the SM.

8.2.13 timestep: A slice of simulation time between two synchronization times.

8.3 Time management schemes

Different time management schemes can be selected at the creation of an exercise. The time management schemes associated with this protocol are as follows:

a) *Real-World Time.* This scheme is the default time management scheme for DIS exercises. In this case, no time management service is required. The simulation timestamp is the same as the absolute timestamp.

b) *Scaled Time.* Simulation time advances at a constant rate relative to real-world time. Non-Real Time protocol parameters are set before the start of the exercise by the SM. A simulation application may provide time advance reports for exercise synchronization control. This reporting includes an estimation of the maximum application rate capability sufficient to prevent processing delay. These application rates may be estimated by the applications themselves from the processor load since the previous time advance report. To avoid processor overload, the SM shall check that the simulation rate is less than the minimum of all application rates. In case of processor overload, the SM shall freeze the applications at a common simulation time, shall wait for the acknowledge of the slowest application, shall change the simulation rate, and shall resume the applications.

   Two modes of the Scaled Time scheme shall be managed by the SM:

   1) Fixed Rate. A constant time advancement rate that shall be set before starting the exercise and never changed even if the exercise is frozen.

   2) Variable Rate. The SM shall control the time advance reports and may change the time advancement rate during the exercise after a simulation time.

c) *Scaled and Stepped.* This time management scheme shall apply a constant rate for a simulation timestep and then shall wait for synchronization at the end of the timestep. The SM shall wait for the receipt of a time advance request from all applications at the same simulation time before sending a common time advance grant request. The SM shall control the simulation rate as in the Scaled Time scheme. This time management shall be used if, for one application, the processing delay for a slice of simulation time is unpredictable (e.g., garbage collecting and heavy computation). A simulation timestep delay shall be the maximum delay accepted for simulation time synchronization.

   Two modes of this time scheme shall be managed by the SM:

1) Fixed Step. The applications shall be frozen at the end of each simulation timestep and before a new time step starts.
2) Variable Step. As soon as the SM has received a time advance report from all applications, the SM shall send a new time step. The applications shall continue until the end of this new time step. The applications shall be frozen only if a new time step is not received.

8.4 Data representation

The following specifies the data representation for specific Non-Real-Time protocol data:

a) Application rate shall be represented by a 32-bit floating point number.
b) Application timestamp shall be represented by a timestamp (see 6.2.88).
c) Feedback time shall be represented by a timestamp (see 6.2.88).
d) Simulation rate shall be represented by a 32-bit floating point number.
e) Simulation time shall be represented by a Clock Time record (see 6.2.14).
f) Simulation timestep shall be represented by a Clock Time record (see 6.2.14).
g) Time interval shall be represented by a timestamp (see 6.2.88).
h) Time latency shall be represented by a timestamp (see 6.2.88).
i) Time scheme shall be represented by a 32-bit unsigned integer.

8.5 Reliability

A problem could occur if a simulation application participating in an exercise silently fails or if PDUs from a simulation application requesting a time advance are lost. The Simulation Management with Reliability protocol (see 5.12 and 7.11) shall be used in the Non-Real-Time protocol (i.e., the requests shall be issued again after an absolute time delay and the SM resigns the simulation application or freezes the exercise after a number of attempts).

A predefined minimum exercise rate shall provide to the SM a maximum absolute time delay to receive an advance report or a time advance grant for the current timestep. An extra time advance report may be requested by the SM at any time.

8.6 Time management services

8.6.1 General

The Non-Real-Time protocol contains the following six time management services:

a) Join Exercise Request
b) Set Time Parameters
c) Time Advance Report
d) Time Advance Request
e) Time Advance Grant
f) Resign Exercise Request

The Non-Real-Time protocol uses the following eight Simulation Management with Reliability PDUs to perform the time management services associated with this protocol:

— Action Request-R PDU
— Action Response-R PDU

— Set Data-R PDU
— Data Query-R PDU
— Data-R PDU
— Start/Resume-R PDU
— Stop/Freeze-R PDU
— Acknowledge-R PDU

When used in the Non-Real-Time protocol, the above eight Simulation Management with Reliability PDUs shall specify the Non-Real-Time protocol as the protocol family in the PDU Header record.

8.6.2 Join Exercise Request service

8.6.2.1 General

The Join Exercise Request time management service provides the means for applications desiring to operate in non-real time to request to join an exercise. The SM shall manage the simulation time and rate only for those applications that have requested to join the exercise.

8.6.2.2 Join Exercise Request before start of exercise

A simulation application shall request to join an exercise by issuing an Action Request-R PDU to the SM. Execution of the Join Exercise Request prior to the start of an exercise shall indicate the application is ready to start. The SM shall respond by issuing an Action Response-R PDU to the simulation application. The Start/Resume-R PDU shall be issued by the SM when all predefined applications have requested to join the exercise.

The Non-Real-Time protocol parameters for an application shall be initialized by the Action Response-R PDU, but these parameters could be changed after each new application has joined the exercise.

The Join Exercise Request actions shall be as shown in Figure 53.

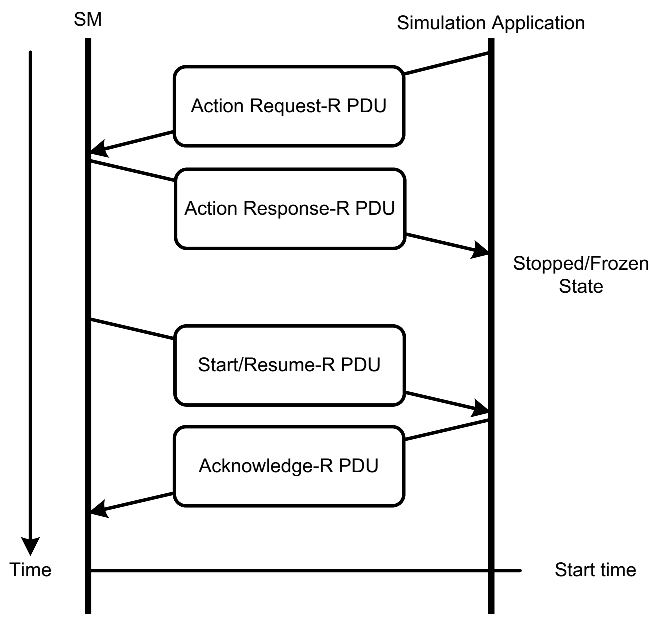


Figure 53—Join Exercise Request

8.6.2.3 Join Exercise Request after start of exercise

Execution of the Join Exercise Request following the start of an exercise shall indicate the application is ready to start immediately or at a specified simulation time. The SM shall respond by issuing an Action Response-R PDU to the simulation application and shall immediately issue a Start/Resume-R PDU for this application or shall issue a Start/Resume-R PDU when the specified simulation time occurs. These join exercise actions shall also be as shown in Figure 53.

Note that for the Scaled and Stepped time management scheme, the application may start at an intermediate time into a timestep. The initial timestep specified in the Action Response-R PDU shall be equal to the simulation time remaining before the next exercise synchronization time.

NOTE—For entities immediately joining an exercise (e.g., guided munition on a fired event), it is strongly recommended that the supporting application is already initialized when the entity creation occurs.

8.6.2.4 Absolute timestamp synchronization

An optional absolute time synchronization service is provided with the Join Exercise Request. This synchronization service is offered when the applications have no local mechanisms for clock synchronization. This synchronization is based on the absolute timestamp of the SM compensated by an estimation of the communication delay between the issuance of the join exercise Action Request-R PDU and the receipt of the associated Action Response-R PDU.

Before an application joins an exercise, the application may not have access to the absolute time of the SM. A means to allow the application to synchronize its real-world time with the SM real-world time would be for the application to issue its first Simulation Management with Reliability PDU using a relative timestamp. The SM shall return the equivalent absolute timestamp in the feedback time parameter of the join exercise Action Response-R PDU. After the receipt of the Action Response-R PDU, the application shall use an absolute timestamp.

8.6.2.5 Information contained in the Join Exercise Request

The Action Request-R PDU issued by a simulation application requesting to join an exercise shall contain the following information:

a) Identification of the SA issuing the Join Exercise Request
b) Identification of the SM
c) Request identification number for the action being requested by the SA
d) Identification of the specific action to be taken (Action ID = Join Exercise)
e) Level of reliability service to be used for the requested action
f) Number of datum, datum length, identification, and values pertinent to the Join Exercise Request:
    1) Exercise Name—the name of the exercise
    2) Application Time—the initial simulation time at which the SA wishes to join the exercise. The absence of this datum parameter in a join exercise Action Request-R PDU shall indicate the intention to immediately join the exercise
    3) Application Rate—the predefined computational rate capability of the simulation application
    4) Application Timestep—the maximum timestep authorized by the application

8.6.2.6 Issuance of the Join Exercise Request

The Action Request-R PDU issued by a simulation application requesting to join an exercise shall be issued using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

8.6.2.7 Receipt of the Join Exercise Request

An SM responding to a join exercise Action Request-R PDU shall issue an Action Response-R PDU to confirm receipt of the request. The Action Response-R PDU shall be issued using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group. The Action Response-R PDU shall contain the following information:

a) Identification of the SM
b) Identification of the simulation application that issued the Join Exercise Request
c) Identification number of the action being processed
d) Status of the action request
e) Number of datum, datum length, identification, and values pertinent to the join exercise response:
    1) Time Scheme—the time management scheme to be used in the exercise by the application
    2) Feedback Time—the timestamp of the Action Request-R PDU (the timestamp in the Action Response-R PDU Header shall contain the corresponding SM timestamp)
    3) Time Latency—an estimation of the communication time latency between the SM and the simulation application
    4) Simulation Time—the initial simulation time necessary to join the execution
    5) Simulation Rate—the initial time advancement rate
    6) (Required for the Scaled and Stepped scheme) Simulation Timestep—the initial timestep

8.6.3 Set Time Parameters service

8.6.3.1 General

This service shall be used by the SM to initialize or change the Non-Real-Time protocol parameters of an application. The initialization of the Non-Real-Time protocol parameters shall be done for each application joining the exercise before the start of an exercise. The SM shall set the simulation Non-Real-Time protocol parameters by issuing a Set Data-R PDU. The receiving application shall confirm the receipt of the Non-Real-Time protocol parameters by issuing a Data-R PDU.

The Set Time Parameter actions shall be as shown in Figure 54.

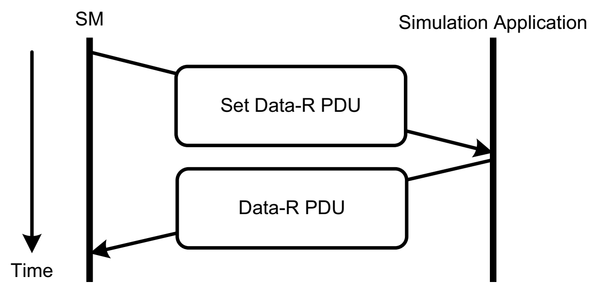


Figure 54—Set Time Parameters

8.6.3.2 Information contained in the Set Time Parameters direction

The Set Data-R PDU issued by an SM directing the application to set time parameters shall contain the following information:

a) Identification of the SM
b) Identification of the SA to receive the Set Data-R PDU
c) Identification number for the data issued by the SM
d) Level of reliability service to be used for the data issued by the SM
e) Number of datum, datum length, identification, and values pertinent to the Set Time Parameters action:
    1) Simulation Time—the initial simulation time at which the exercise shall start
    2) Simulation Rate—the initial time advance rate
    3) (Required for the Scaled and Stepped scheme) Simulation Timestep—the initial timestep

8.6.3.3 Issuance of the Set Time Parameters direction

The Set Data-R PDU issued by an SM directing the application to set time parameters shall be issued using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

8.6.3.4 Receipt of the Set Time Parameters direction

The simulation application that receives the set time parameters Set Data-R PDU shall issue a Data-R PDU confirming the information received in the Set Data-R PDU. The Data-R PDU shall be issued using a unicast of multicast communication service. If sent using the multicast communication service, then the

PDU shall be sent to all simulation applications that are members of the addressed multicast group. The Data-R PDU shall contain the following information:

a) Identification of the simulation application
b) Identification of the SM
c) Identification number for the data sent by the SM
d) Number of datum, datum length, identification, and values received in the set time parameters Set Data-R PDU:
    1) Simulation Time
    2) Simulation Rate
    3) (Required for the Scaled and Stepped scheme) Simulation Timestep

8.6.4 Time Advance Report service

8.6.4.1 General

The Time Advance Report service provides a means for the SM to request a time advance report from the simulation application. The SM shall issue a Data Query-R PDU to request the time advance report. The application shall issue the time advance report by transmitting immediately and then periodically a Data-R PDU.

The Time Advance Report actions shall be as shown in Figure 55.

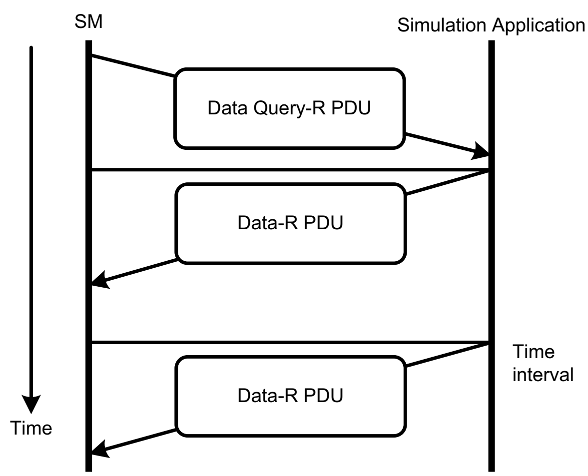


Figure 55—Time Advance Report

8.6.4.2 Information contained in the Time Advance Report

The time advance report Data Query-R PDU issued by the SM shall contain the following information:

a) Identification of the SM
b) Identification of the simulation application to which the query is addressed
c) Simulation time interval between the issuance of time advance reports
d) Identification number of the data being requested

e) Level of reliability service to be used for the data being requested

f) Number of datum, datum length, identification, and values pertinent to the Time Advance Report:

   1) Application Time—the actual simulation time in the application at the time of the Time Advance Report

   2) Application Rate—the actual maximum rate in the application at the time of the Time Advance Report

8.6.4.3 Issuance of the Time Advance Report

The time advance report Data Query-R PDU shall be issued by the SM using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

8.6.4.4 Receipt of the Time Advance Report

A simulation application responding to an SM with a Time Advance Report shall issue a Data-R PDU using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group. The Data-R PDU shall contain the following information for each report:

a) Identification of the simulation application issuing the time advance report Data-R PDU

b) Identification of the SM

c) Identification number for the data sent to the SM

d) Number of datum, datum length, identification, and values received in the time advance report Data Query-R PDU:

   1) Application Time

   2) Application Rate

8.6.5 Time Advance Request service

8.6.5.1 General

The Time Advance Request service provides the means for the SA to inform the SM that a simulation is ready to advance its internal state to a new simulation time. The SA shall issue an Action Request-R PDU and the SM shall return an Action Response-R PDU.

The Time Advance Request actions shall be as shown in Figure 56.

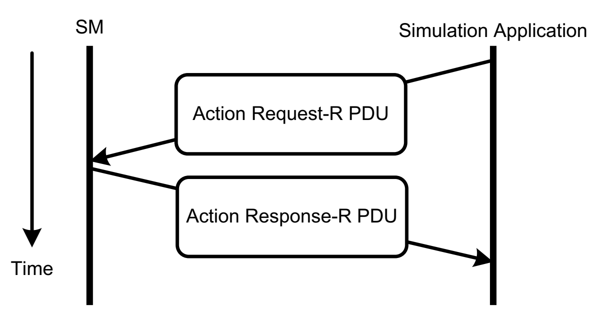


Figure 56—Time Advance Request

8.6.5.2 Information contained in the Time Advance Request

The time advance Action Request-R PDU shall contain the following information:

a) Identification of the simulation application issuing the Time Advance Request

b) Identification of the SM

c) Request identification number for the action being requested by the simulation application

d) Identification of the specific action to be taken (Action ID = Time Advance)

e) Level of reliability service to be used for the requested action

f) Number of datum, datum length, identification, and values pertinent to the Time Advance Request:

   1) Application Time—the actual simulation time in the application at the time of the Time Advance Request

   2) Application Rate—the actual maximum rate in the application at the time of the Time Advance Request

   3) Application Timestep—the maximum delta time requested by the application for the time advance

8.6.5.3 Issuance of the Time Advance Request

The time advance Action Request-R PDU shall be issued using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

8.6.5.4 Receipt of the Time Advance Request

An SM responding to a time advance Action Request-R PDU shall issue using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group. The Action Response-R PDU shall contain the following information:

a) Identification of the SM

b) Identification of the SA issuing the Action Request-R PDU

c) Identification number of the action being processed

d) Status of the action request

8.6.6 Time Advance Grant service

8.6.6.1 General

The Time Advance Grant time management service is the means by which the SM informs an application that it can advance a timestep from the current simulation time. The SM shall issue a Set Data-R PDU, and the simulation application shall immediately return a Data-R PDU.

The Time Advance Grant actions shall be as shown in Figure 57.

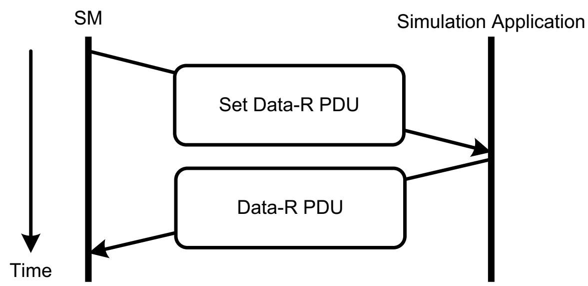


Figure 57—Time Advance Grant

8.6.6.2 Information contained in the Time Advance Grant

The time advance grant Set Data-R PDU issued by the SM shall contain the following information:

a) Identification of the SM
b) Identification of the simulation application to which the time advance grant Set Data-R PDU is addressed
c) Identification number for the data issued by the SM
d) Level of reliability service to be used for the data requested by the SM
e) Number of datum, datum length, identification, and values pertinent to the Time Advance Grant:
    1) Simulation Time—the next simulation time to which to advance time
    2) Simulation Rate—the effective simulation rate to apply until the next simulation time
    3) Simulation Timestep—the actual time delta used by the SM

8.6.6.3 Issuance of the Time Advance Grant

The time advance grant Set Data-R PDU shall be issued by the SM using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

8.6.6.4 Receipt of the Time Advance Grant

The simulation application responding to a Time Advance Grant shall issue a Data-R PDU confirming the information received in the Set Data-R PDU. The Data-R PDU shall be issued using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group. The Data-R PDU shall contain the following information:

a) Identification of the simulation application issuing the Data-R PDU
b) Identification of the SM
c) Identification number for the data being sent to the SM
d) Number of datum, datum length, identification, and values received in the Set Data-R PDU:
    1) Simulation Time
    2) Simulation Rate
    3) Simulation Timestep

8.6.7 Resign Exercise Request service

8.6.7.1 General

The Resign Exercise Request service provides the means for an application to indicate that it wishes to leave the exercise. A simulation application shall request to resign from the exercise by issuing an Action Request-R PDU to the SM. The SM shall respond by issuing an Action Response-R PDU to the simulation application. After receipt of the Action Response, the application shall remain in the exercise until it receives a Stop/Freeze-R PDU. The application shall issue a final Acknowledge-R PDU before resigning from the exercise.

The Resign Exercise Request actions shall be as shown in Figure 58.

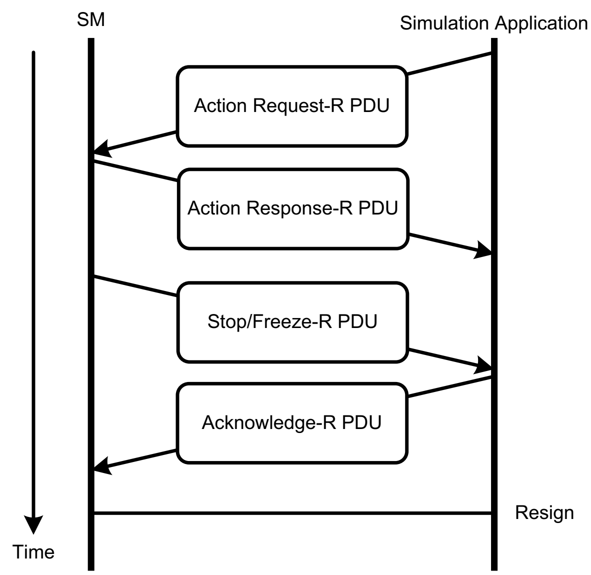


Figure 58—Resign Exercise Request

8.6.7.2 Information contained in the Resign Exercise Request

The simulation application wishing to resign from the exercise shall issue an Action Request-R PDU containing the following information:

a) Identification of the simulation application issuing the resign exercise Action Request-R PDU

b) Identification of the SM

c) Request identification number for the action being requested by the simulation application

d) Identification of the specific action to be taken (Action ID = Resign Exercise)

e) Level of reliability service to be used for the requested action

f) Number of datum, datum length, identification, and values pertinent to the Resign Exercise Request:

    1) Simulation Time—the simulation time at which the application wishes to resign from the exercise. The absence of this datum parameter shall indicate the intention to resign as soon as possible.

8.6.7.3 Issuance of the Resign Exercise Request

The simulation application wishing to resign from the exercise shall issue an Action Request-R PDU using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group.

8.6.7.4 Receipt of the Resign Exercise Request

An SM receiving a resign exercise Action Request-R PDU shall issue an Action Response-R PDU to confirm receipt of the request. The Action Response-R PDU shall be issued using a unicast or multicast communication service. If sent using the multicast communication service, then the PDU shall be sent to all simulation applications that are members of the addressed multicast group. The Action Response-R PDU shall contain the following information:

a) Identification of the SM

b) Identification of the SA requesting to resign from the exercise

c) Identification number for the action being processed

d) Status of the action request

9 Live Entity (LE) Information/Interaction protocol

9.1 General

This clause contains the specification of a protocol for use in DIS applications where conservation of network bandwidth is a prime concern. Although created primarily for use by systems involving live entities communicating over bandwidth-limited communication mechanisms, it is available for other similar uses. With the exception of references to record definitions found in Clause 6, this clause is self-contained consisting of an overview of the major architectural principles on which this protocol is based, the definition of protocol-specific data structures, and the specification of the PDUs that constitute the LE Information/Interaction protocol. In support of the reduction in communications channel bandwidth needed by the elements of this protocol, the PDUs of this protocol are not required to adhere to the record alignment requirements of 6.3.2.

9.2 Architectural principles

The LE Information/Interaction protocol relies on specific architectural principles that differ from those on which the previous clauses of this standard are based. The primary need for these protocol-specific principles is the limited communications bandwidth available to an LE participating in a DIS exercise. The following are the major architecture changes:

a) Increase in the maximum allowable transmission interval for the purposes of dead reckoning

b) Conversion of floating point fields to fixed binary fields

c) Decrease in the length of the fields constituting a PDU

d) Allowance for one or more subprotocol definitions that specify field lengths and format to be selected for use in an exercise prior to the start of an exercise

e) Allowance for some fields in some PDUs to be optional

In addition to the above, several architecture augmentations, summarized in Table 213, are required to support the special needs of live participants in a DIS exercise.

Table 213—Architecture augmentations for supporting LE

Augmentation	Rationale
Reduce length of frequently transmitted data	Break Entity State PDU into multiple PDUs so that they may be transmitted at different intervals and so that slowly changing information need not be reported as frequently as quickly changing information. Reduce the size of the PDU fields by eliminating implied precision that does not match the accuracy of the data measured.
Eliminate unused or non-changing fields in PDUs	Some information on some platforms cannot be supplied because it is not instrumented. A field containing a set of bit flags is included in some PDUs to signal the presence or absence of specific optional fields.
Use reference points for specification of LE location	By distributing, prior to an exercise, the 64-bit floating point world coordinate locations of a series of geographic reference points specific to the exercise domain, an LE's actual location can be communicated by means of a triplet of delta distances from a given reference point and the identification of the reference point.

To reduce the communication bandwidth needed during an exercise, certain unchanging exercise-specific data can be made available to the simulation applications prior to the start of an exercise in a standard format. Specifically, the LE Information/Interaction protocol uses a system of geographic reference points that can be defined and distributed prior to an exercise. These predefined reference points allow the actual location of an LE to be transmitted as delta offsets from one of these reference points. The suggested reference point file definition format is shown in Table 214, and the fields of this file format are as follows:

a) *Version*. This field shall specify the version identification of the data file and shall be represented by a 16-bit unsigned integer.

b) *Number of Reference Points*. This field shall specify the number of 64-bit triplets of world location data that follow in the file. This field shall be represented by a 16-bit unsigned integer.

c) *Reference Point*. This field shall specify the location of each reference point in the simulated world. This field shall be represented by a World Coordinates record (see 6.2.98).

Table 214—Reference Point file definition format

Field size (bits)	Field name	Data type
16	Version	16-bit unsigned integer
16	Number of reference points (*NR*)	16-bit unsigned integer

Table 214—Reference Point file definition format *(continued)*

Field size (bits)	Field name	Data type
192	Reference point (1 – *NR*)	World-*X*—64-bit floating point
		World-*Y*—64-bit floating point
		World-*Z*—64-bit floating point
Total Reference Point file format size = 224 bits		

Each reference point is logically assigned the next sequential number starting with zero.

The translation of a reference-point-based location (ref-*n*, Delta-*X*, Delta-*Y*, Delta-*Z*) into a standard WGS-84 *X*, *Y*, *Z* triplet shall be as shown:

X = World-X(ref-n) + Delta-X

Y = World-Y(ref-n) + Delta-Y

Z = World-Z(ref-n) + Delta-Z

The translation into a reference-based location is based on finding the reference point that minimizes the delta offsets from that reference point.

9.3 Basic records

9.3.1 General

This subclause specifies requirements for records specific to the LE Information/Interaction protocol.

9.3.2 LE PDU Header record

An LE PDU Header record shall be the first part of each LE Information/Interaction PDU. The fields of this record are as follows:

a) *Protocol Version.* This field shall specify the version of protocol used in this PDU and shall be specified by an 8-bit enumeration (see [UID 3]).
b) *Exercise ID.* This field shall specify the exercise to which the PDU pertains and shall be represented by an Exercise Identifier (see 6.2.34).
c) *PDU Type.* This field shall indicate the type of PDU that follows. This field shall be represented by an 8-bit enumeration (see [UID 4]).
d) *Protocol Family.* This field shall indicate the family of protocols to which the PDU belongs. This field shall be represented by an 8-bit enumeration (see [UID 5]).
e) *Timestamp.* This field shall specify the reference time corresponding to the simulation time at which the data in the PDU were generated and shall be represented by a Timestamp (see 6.2.88).
f) *Length.* This field shall specify the length of the PDU, including the PDU Header, in octets, and shall be represented by a 16-bit unsigned integer.
g) *Subprotocol Number.* This field shall specify the subprotocol to be used to decode the PDU. This field is represented by an 8-bit enumeration (see [UID 417]). No Subprotocol (0) is reserved for PDUs that are defined according to the requirements in Clause 5 and Clause 7 of this standard.

h) *Padding.* Although padding fields are not normally used in the LE Information/Interaction PDUs, the LE PDU Header record is padded so that it maintains the same length as the PDU Header in 6.2.66.

The format of the LE PDU Header record shall be as shown in Table 215.

Table 215—LE PDU Header record

Field size (bits)	Field name	Data type
8	Protocol Version	8-bit enumeration
8	Exercise ID	8-bit unsigned integer
8	PDU Type	8-bit enumeration
8	Protocol Family	8-bit enumeration
32	Timestamp	32-bit unsigned integer
16	Length	16-bit unsigned integer
8	Subprotocol Number	8-bit enumeration
8	Padding	8 bits unused
Total LE PDU Header record size = 96 bits		

9.3.3 Relative World Coordinates record

Location of the origin of the LE's coordinate system shall be specified by a set of four data values: reference point, Delta-*X*, Delta-*Y*, and Delta-*Z*. The shape of the Earth and the origin and orientation of this coordinate system shall be as defined in 1.6.3. The fields of this record are as follows:

a) *Reference Point.* This field shall identify the number of the reference point against which the Delta-*X*, Delta-*Y*, and Delta-*Z* distances are calculated. This field shall be represented by a 16-bit unsigned integer.

b) *Delta-X.* This field shall specify the difference between LE's *X*-direction coordinate and the *X*-direction coordinate of the reference point. This field shall be represented by a 16-bit fixed binary number. The binary point is defined by the subprotocol, but for the purposes of this standard, the binary point shall be assumed to be 3 (i.e., the least significant bit shall represent 0.125 m).

c) *Delta-Y.* This field shall specify the difference between LE's *Y*-direction coordinate and the *Y*-direction coordinate of the reference point. This field shall be represented by a 16-bit fixed binary number. The binary point is defined by the subprotocol, but for the purposes of this standard, the binary point shall be assumed to be 3 (i.e., the least significant bit shall represent 0.125 m).

d) *Delta-Z.* This field shall specify the difference between LE's *Z*-direction coordinate and the *Z*-direction coordinate of the reference point. This field shall be represented by a 16-bit fixed binary number. The binary point is defined by the subprotocol, but for the purposes of this standard, the binary point shall be assumed to be 3 (i.e., the least significant bit shall represent 0.125 m).

The format of the Relative World Coordinates record shall be as shown in Table 216.

Table 216—Relative World Coordinates record

Field size (bits)	Field name	Data type
16	Reference Point	16-bit unsigned integer
16	Delta-*X*	16-bit fixed binary
16	Delta-*Y*	16-bit fixed binary
16	Delta-*Z*	16-bit fixed binary
Total Relative World Coordinates record size = 64 bits		

9.3.4 Position Error record

This record defines the error components that are associated with the location measurement for an entity. These errors are measured in entity-centered tangent planes. Different error factors are associated with the horizontal and vertical error measurements. The binary point is defined by the subprotocol, but for the purposes of this standard, the binary point will be assumed to be 3 (i.e., the least significant bit shall represent 0.125 m). The format of the Position Error record shall be as shown in Table 217.

Table 217—Position Error record

Field size (bits)	Field name	Data type
16	Horizontal Error	16-bit fixed binary
16	Vertical Error	16-bit fixed binary
Total Position Error record size = 32 bits		

9.3.5 Orientation Error record

This record defines the orientation error components that are associated with the measurement of the entity's orientation. Units of measurement are in radians. The binary point is defined by the sub-protocol, but for the purposes of this standard the binary point shall be 8 (i.e., the least significant bit shall represent 0.003 906 2 radians). The format of the Orientation Error record shall be as shown in Table 218.

Table 218—Orientation Error record

Field size (bits)	Field name	Data type
16	Azimuth Error	16-bit fixed binary

Table 218—Orientation Error record *(continued)*

Field size (bits)	Field name	Data type
16	Elevation Error	16-bit fixed binary
16	Rotation Error	16-bit fixed binary
Total Orientation Error record size = 48 bits		

9.4 LE Information/Interaction protocol family

9.4.1 General

The LE Information/Interaction protocol consists of the following PDUs:

a) Time Space Position Information (TSPI) PDU
b) Appearance PDU
c) Articulated Parts PDU
d) LE Fire PDU
e) LE Detonation PDU

Each LE Information/Interaction PDU is defined using one or more standard fields. To promote bandwidth conservation, all parameters need not be transmitted in each issuance of a PDU, based on the ability of the entity to measure them or the fact that they are required for the exercise. Each PDU is constructed by placing fields from a set of standard fields into the proper order. For any individual PDU, some of the fields may be optional. If an optional data field is included in the PDU, then a bit field corresponding to the data field is set to one in an associated flag field and the data field shall be included in the PDU. If a data field is not present, then the bit field associated with the data field shall be set to zero in the flag field and no information relative to that data field shall be included in the PDU.

9.4.2 Time Space Position Information (TSPI) PDU

9.4.2.1 Purpose

The TSPI PDU shall communicate information about the LE's state vector. This PDU includes state information that is necessary for the receiving simulation applications to represent the issuing LE's location and movement in its own simulation.

9.4.2.2 Information contained in the TSPI PDU

The TSPI PDU shall contain the following information subject to the settings in the flag field:

a) Identification of the entity that issued the PDU.
b) Flags to specify the existence of optional data fields. The format of the TSPI Flag field shall be as shown in Table 219.
c) Information about the location of the LE in the real world and its real-world orientation, including:
    1) Location with respect to the world.
    2) Velocity.
    3) Orientation.
    4) Potential position and orientation reporting errors.

5) Dead reckoning parameters that should be employed when extrapolating the position of this entity. (Values in this field shall optionally include dead reckoning algorithm in use, linear acceleration, and angular velocity.)
6) Measured speed.

d) Length of system-specific data contained in the TSPI PDU.
e) System-specific data that may be shared between like systems.

Table 219—TSPI Flag field definition

Bit	State	Flag Octet 1 meaning
7	0	Reserved for flag continuation flag
6	0	System-Specific Data field is not included
	1	System-Specific Data field is included with the length defined by the System-Specific Data Length field
5	0	Measured Speed field is not included
	1	Measured Speed field is included
4	0	Dead Reckoning Parameter field is not included (use subprotocol default)
	1	Dead Reckoning Parameter field is included; algorithm requirements establish the need and existence of the Entity Linear Acceleration and Entity Angular Velocity fields
3	0	Orientation Error field is not included
	1	Orientation Error field is included
2	0	Position Error field is not included
	1	Position Error field is included
1	0	Entity Orientation field is not included
	1	Entity Orientation field is included
0	0	Entity Linear Velocity field is not included
	1	Entity Linear Velocity field is included
Total TSPI Flag field size = 8 bits		

9.4.2.3 Issuance of the TSPI PDU

The TSPI PDU shall be issued by an entity when any of the following occur:

a) The discrepancy between an entity's actual state (as determined by its own internal model or measurements) and its dead reckoned state (using specified dead reckoning algorithm) exceeds a predetermined threshold (see Annex E concerning dead reckoning and 5.3.2.3 concerning threshold values). This threshold includes changes in position/orientation information.
b) A predetermined length of real-world time has elapsed since the issuing of the last TSPI PDU. The TSPI PDU heartbeat parameter and tolerance shall be identified by the symbolic name HBT_PDU_TSPI. (See 6.1.8 for parameter details and default values.)
c) A change occurs in the dead reckoning algorithm used by the simulation application.

The TSPI PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

9.4.2.4 Receipt of the TSPI PDU

Upon receipt of the TSPI PDU, a simulation application shall determine whether the PDU contains more current information than that currently being used to model the transmitting entity. If so, the simulation application shall use the information contained therein to model the position and orientation of the entity that issued the PDU. Otherwise, the PDU shall be discarded.

If a TSPI entity's countdown timer expires, then all simulation applications shall remove that entity from the exercise. The TSPI entity timeout parameter shall be the value of HBT_PDU_TSPI multiplied by HBT_TIMEOUT_MPLIER.

9.4.2.5 Format of the TSPI PDU

Information about a particular entity shall be communicated by issuing a TSPI PDU. The TSPI PDU shall contain the following fields subject to the settings in the TSPI Flag field:

a) *LE PDU Header.* This field shall contain data common to all LE Information/Interaction PDUs. The LE PDU Header shall be represented by the LE PDU Header record (see 9.3.2).

b) *Live Entity ID.* This field shall identify the entity issuing the PDU. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall be only 8 bits long.

c) *TSPI Flags.* This field shall identify those optional data fields that are being transmitted with the current PDU. This field shall be defined as an 8-bit record as specified in item b) in 9.4.2.2.

d) *Entity Location.* This field shall specify an entity's physical location in the real world. This field shall be represented by a Relative World Coordinates record (see 9.3.3). Each delta distance component shall represent distance in decimeters.

e) *Entity Linear Velocity.* This optional field shall specify an entity's linear velocity. The coordinate system for an entity's linear velocity depends on the dead reckoning algorithm used. The structure of this field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96] subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol. Each vector component shall represent velocity in decimeters per second.

f) *Entity Orientation.* This optional field shall specify an entity's orientation. The structure of this field shall be represented by an Euler Angles record (see 6.2.32) subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol.

g) *Position Error.* This optional field shall specify the potential horizontal and vertical position error of the reporting entity. This field shall be represented by a Position Error record (see 9.3.4).

h) *Orientation Error.* This optional field shall specify the potential orientation error of the reporting entity. This field shall be represented by an Orientation Error record (see 9.3.5) that describes the error components in azimuth, elevation, and rotation measured against the entity's reported orientation.

i) *Dead Reckoning Parameters.* This optional field shall specify parameters for dead reckoning the position and orientation of the reporting entity. Dead reckoning algorithm in use, entity linear acceleration (optional based on dead reckoning algorithm in use), and entity angular velocity (optional based on dead reckoning algorithm in use) shall be included as part of the dead reckoning parameters. If the Dead Reckoning Parameters field is not included in the PDU, the recipient shall assume the use of a linear dead reckoning algorithm [Annex E algorithm DRM (FPW)] and shall assume all angular velocity and linear acceleration parameters to be zero:

   1) Dead Reckoning Algorithm. This field shall specify the dead reckoning algorithm in use by the issuing entity and shall be represented by an 8-bit enumeration (see [UID 44]).

2) Entity Linear Acceleration. This optional field shall specify the entity's linear acceleration. The structure of this field shall be represented by a Linear Acceleration Vector record [see item b) in 6.2.96] subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol. Each vector component shall represent acceleration in decimeters per second per second.

3) Entity Angular Velocity. This optional field shall specify the entity's angular velocity. The structure of this field shall be represented by an Angular Velocity Vector record (see 6.2.7) subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol. Each vector component shall represent angular velocity in radians per second.

j) *Measured Speed.* This optional field shall specify the entity's own measurement of speed (e.g., air speed for aircraft). This field shall be represented by a 16-bit fixed binary representing speed in decimeters per second.

k) *System-Specific Data Length.* This optional field shall specify the number of octets of system-specific data that immediately follow this field. If this field is present, the System-Specific Data field shall also be present and shall have a nonzero length. This field shall be represented by an 8-bit unsigned integer.

l) *System-Specific Data.* This optional field shall contain user-defined data and be specified in length by item k) above. The contents of this field are arbitrary and outside the bounds of this specification with the exception that no data shall be transmitted in this field that duplicates or can be derived from the data transmitted in this or other LE Information/Interaction PDUs.

The format of the TSPI PDU shall be as shown in Table 220.

Table 220—TSPI PDU

Field size (bits)	TSPI PDU fields	
96	LE PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 46
		Protocol Family—8-bit enumeration = 11
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		Subprotocol Number—8-bit enumeration
		Padding—8 bits unused
32	Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer

Table 220—TSPI PDU *(continued)*

Field size (bits)	TSPI PDU fields	
8	TSPI Flag	0 (Reserved)—1-bit Boolean
		System-Specific Data field included—1-bit Boolean
		Measured Speed field included—1-bit Boolean
		Dead Reckoning Parameter field included—1-bit Boolean
		Orientation Error field included—1-bit Boolean
		Position Error field included—1-bit Boolean
		Entity Orientation field included—1-bit Boolean
		Entity Linear Velocity field included—1-bit Boolean
64	Entity Location	Reference Point—16-bit unsigned integer
		Delta-*X*—16-bit fixed binary
		Delta-*Y*—16-bit fixed binary
		Delta-*Z*—16-bit fixed binary
48	Entity Linear Velocity	*x*-component—16-bit fixed binary
		y-component—16-bit fixed binary
		z-component—16-bit fixed binary
24	Entity Orientation	Psi (ψ)—8-bit fixed binary
		Theta (θ)—8-bit fixed binary
		Phi (ϕ)—8-bit fixed binary
32	Position Error	Horizontal Error—16-bit fixed binary
		Vertical Error—16-bit fixed binary
48	Orientation Error	Azimuth Error—16-bit fixed binary
		Elevation Error—16-bit fixed binary
		Rotation Error—16-bit fixed binary
56	Dead Reckoning Parameters	Dead Reckoning Algorithm—8-bit enumeration
		Entity Linear Acceleration—3 × 8-bit fixed binary
		Entity Angular Velocity—3 × 8-bit fixed binary
16	Measured Speed	16-bit fixed binary
8	System-Specific Data Length (*N*)	8-bit unsigned integer
8*N*	System-Specific Data	[see item l) in 9.4.2.5]
Total TSPI PDU size = 432 + 8*N* bits where *N* is the number of octets of system-specific data		

9.4.3 Appearance PDU

9.4.3.1 Purpose

The Appearance PDU shall communicate information about the appearance of an LE. This includes state information that is necessary for the receiving simulation applications to represent the issuing entity's appearance in the simulation application's own simulation.

9.4.3.2 Information contained in the Appearance PDU

The Appearance PDU shall contain the following information subject to the settings in the flag field:

a) Identification of the entity that issued the PDU.
b) Flags to specify the existence of optional data fields. The format of the Appearance Flag fields shall be as shown in Table 221 and Table 222.
c) Identification of the force to which the entity belongs.
d) Issuing entity's specific entity type.
e) Issuing entity's alternate entity type for use with the Guise function (see 5.3.2.6).
f) Capabilities of the entity, including:
    1) Resupply.
    2) Repair.
g) Information required for representation of the entity's appearance, including:
    1) Markings.
    2) Visual appearance of the entity (e.g., normal, smoking, on fire, and producing a dust cloud).
    3) Infrared (IR) appearance of the entity (e.g., heat sources on the outside).
    4) Electromagnetic (EM) appearance of the entity (e.g., electromagnetic masking system is on).
    5) Audio appearance of the entity (e.g., sound-deadening systems have to be activated).

Table 221—Appearance Flag-1 field definition

Bit	State	Flag Octet 1 meaning
7	0	Flag Octet 2 is not included
	1	Flag Octet 2 is included
6	0	Appearance-IR field is not included
	1	Appearance-IR field is included
5	0	Appearance-Visual field is not included
	1	Appearance-Visual field is included
4	0	Capabilities field is not included
	1	Capabilities field is included
3	0	Entity Marking field is not included
	1	Entity Marking field is included
2	0	Alternate Entity Type field is not included
	1	Alternate Entity Type field is included

Table 221—Appearance Flag-1 field definition *(continued)*

Bit	State	Flag Octet 1 meaning
1	0	Entity Type field is not included
	1	Entity Type field is included
0	0	Force ID field is not included
	1	Force ID field is included
Total Appearance Flag-1 field size = 8 bits		

Table 222—Appearance Flag-2 field definition

Bit	State	Flag Octet 2 meaning
7	0	Reserved for flag continuation flag
6	0	Unused
5	0	Unused
4	0	Unused
3	0	Unused
2	0	Unused
1	0	Appearance-Audio field is not included
	1	Appearance-Audio field is included
0	0	Appearance-EM field is not included
	1	Appearance-EM field is included
Total Appearance Flag-2 field size = 8 bits		

9.4.3.3 Issuance of the Appearance PDU

The Appearance PDU shall be issued by an entity when any of the following occur:

a) A change occurs in the entity's appearance.

b) A predetermined length of real-world time has elapsed since the issuing of the last Appearance PDU for that entity. The Appearance PDU heartbeat parameter and tolerance shall be identified by the symbolic name HBT_PDU_APPEARANCE. (See 6.1.8 for parameter details and default values.)

c) A change occurs in the entity's capabilities.

d) The entity ceases to exist in the synthetic environment. A final Appearance PDU shall be issued to indicate that the entity is in the Deactivated state.

The Appearance PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

9.4.3.4 Receipt of the Appearance PDU

Upon receipt of the Appearance PDU, a simulation application shall use the information contained therein to model the appearance of the entity that issued the PDU.

If the appearance of an entity indicates it is in the Deactivated state, then all simulation applications shall remove the entity from the exercise.

9.4.3.5 Format of the Appearance PDU

Information about the appearance of an LE shall be communicated by issuing an Appearance PDU. The Appearance PDU shall contain the following fields subject to the settings in the Appearance Flag field:

a) *LE PDU Header.* This field shall contain data common to all LE Information/Interaction PDUs. The LE PDU Header shall be represented by the LE PDU Header record (see 9.3.2).

b) *Live Entity ID.* This field shall identify the entity issuing the PDU. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall be only 8 bits long.

c) *Appearance Flags.* This field shall identify those optional data fields that are being transmitted with the current PDU. This field shall be defined as a 16-bit record as specified in item b) in 9.4.3.2.

d) *Force ID.* This optional field shall identify the force to which the issuing entity belongs and shall be represented by an 8-bit enumeration (see [UID 6]).

e) *Entity Type.* This optional field shall identify the entity type to be displayed by members of the same force as the issuing entity. This field shall be represented by an Entity Type record (see 6.2.30).

f) *Alternate Entity Type.* This optional field shall identify the entity type to be displayed by members of forces other than that of the issuing entity and shall be represented by an Entity Type record (see 6.2.30).

g) *Entity Marking.* This optional field shall identify any unique markings on an entity (e.g., a bumper number or a country symbol). This field shall be represented by an Entity Marking record (see 6.2.29).

h) *Capabilities.* This optional field shall specify the entity’s capabilities. This field shall be represented by an Entity Capabilities record (see 6.2.27).

i) *Appearance-Visual.* This optional field shall specify the dynamic changes to the entity’s visual appearance attributes. This field shall be represented by an Entity Appearance record (see 6.2.26).

j) *Appearance-IR.* This optional field shall specify the dynamic changes to the entity’s infrared appearance attributes. This field shall be represented by an Entity Appearance record (see 6.2.26).

k) *Appearance-EM.* This optional field shall specify the dynamic changes to the entity’s electromagnetic appearance attributes. This field shall be represented by an Entity Appearance record (see 6.2.26).

l) *Appearance-Audio.* This optional field shall specify the dynamic changes to the entity’s acoustic appearance attributes. This field shall be represented by an Entity Appearance record (see 6.2.26).

The format of the Appearance PDU shall be as shown in Table 223.

Table 223—Appearance PDU

Field size (bits)	Appearance PDU fields	
96	LE PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 47
		Protocol Family—8-bit enumeration = 11
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		Subprotocol Number—8-bit enumeration
		Padding—8 bits unused
32	Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Appearance Flag-1	Flag Octet 2 included—1-bit Boolean
		Appearance-IR field included—1-bit Boolean
		Appearance-Visual field included—1-bit Boolean
		Capabilities field included—1-bit Boolean
		Entity Marking field included—1-bit Boolean
		Alternate Entity Type field included—1-bit Boolean
		Entity Type field included—1-bit Boolean
		Force ID field included—1-bit Boolean
$8F_7$	Appearance Flag-2	0 (Reserved)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		Appearance-Audio field included—1-bit Boolean
		Appearance-EM field included—1-bit Boolean
$8F_0$	Force ID	8-bit enumeration

Table 223—Appearance PDU *(continued)*

Field size (bits)	Appearance PDU fields	
$64F_1$	Entity Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
$64F_2$	Alternate Entity Type	Entity Kind—8-bit enumeration
		Domain—8-bit enumeration
		Country—16-bit enumeration
		Category—8-bit enumeration
		Subcategory—8-bit enumeration
		Specific—8-bit enumeration
		Extra—8-bit enumeration
$96F_3$	Entity Marking	Character Set 8-bit enumeration
		11 8-bit unsigned integers
$32F_4$	Capabilities	32-bit record
$(F_5 + F_6 + G_0 + G_1)$	Appearance	Appearance-Visual—32-bit record
		Appearance-IR—32-bit record
		Appearance-EM—32-bit record
		Appearance-Audio—32-bit record

Total Appearance PDU size = $136 + 8F_7 + 8F_0 + 64F_1 + 64F_2 + 96F_3 + 32F_4 + 32(F_5 + F_6 + G_0 + G_1)$ bits

where

F_i is the value of bit i in the Appearance Flag-1 field
G_i is the value of bit i in the Appearance Flag-2 field

$$G_i = \begin{cases} 1 \text{ if } F_7 \text{ is 1 and } G_i \text{ is 1} \\ 0 \text{ if } F_7 \text{ is 0 or } G_i \text{ is 0} \end{cases}$$

9.4.4 Articulated Parts PDU

9.4.4.1 Purpose

The Articulated Parts PDU shall communicate information about an entity's articulated and attached parts. This PDU includes state information that is necessary for the receiving simulation applications to represent the issuing entity's articulated and attached parts appearance in its own simulation.

9.4.4.2 Information contained in the Articulated Parts PDU

The Articulated Parts PDU shall contain the following information:

a) Identification of the entity that issued the PDU

b) Information required for representation of the entity's visual appearance and position of its articulated parts, including:

   1) Number of articulation parameters and the parameter values to represent orientation of articulated parts

   2) Presence (including types and numbers) of attached parts or stores

9.4.4.3 Issuance of the Articulated Parts PDU

The Articulated Parts PDU shall be issued by an entity when the discrepancy between an entity's articulated part actual state (as determined by its own internal model or measurements) and the dead reckoned state (using specified dead reckoning algorithms) of the articulated part exceeds a predetermined threshold (see Annex E concerning dead reckoning and 5.3.2.3 concerning threshold values). This threshold includes changes in position/orientation and articulated part parameter information.

The Articulated Parts PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

9.4.4.4 Receipt of the Articulated Parts PDU

Upon receipt of the Articulated Parts PDU, a simulation application shall determine whether the PDU contains more current information than that currently being used to model the transmitting entity. If so, the simulation application shall use the information contained therein to model the position and orientation of articulated part of the entity that issued the PDU. Otherwise, the PDU shall be discarded.

9.4.4.5 Format of the Articulated Parts PDU

Information about the articulated and attached parts of a live entity shall be communicated by issuing an Articulated Parts PDU. The Articulated Parts PDU shall contain the following fields:

a) *LE PDU Header.* This field shall contain data common to all LE Information/Interaction PDUs. The LE PDU Header shall be represented by the LE PDU Header record (see 9.3.2).

b) *Live Entity ID.* This field shall identify the entity issuing the PDU. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall be only 8 bits long.

c) *Number of Variable Parameter Records.* This field shall specify the number of Variable Parameter records present. This field shall be represented by an 8-bit unsigned integer (see Annex I). The maximum number of Variable Parameter records in an Articulated Parts PDU is a function of MAX_PDU_SIZE_BITS.

d) *Variable Parameter records.* This field shall specify the parameter values for each Variable Parameter record that is included (see 6.2.94 and Annex I).

The format of the Articulated Parts PDU shall be as shown in Table 224.

Table 224—Articulated Parts PDU

Field size (bits)	Articulated Parts PDU fields	
96	LE PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration
		Protocol Family—8-bit enumeration = 48
		Timestamp—32-bit unsigned integer = 11
		Length—16-bit unsigned integer
		Subprotocol Number—8-bit enumeration
		Padding—8 bits unused
32	Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Number of Variable Parameter Records (*N*)	8-bit unsigned integer
128	Variable Parameter record #1	Record Type—8-bit enumeration
		Record-Specific fields—120 bits
⋮		
128	Variable Parameter record #*N*	Record Type—8-bit enumeration
		Record-Specific fields—120 bits
Total Articulated Parts PDU size = 136 + 128*N* bits where *N* is the number of Variable Parameter records		

9.4.5 Live Entity (LE) Fire PDU

9.4.5.1 General

Representation of weapons fire in a DIS exercise involving LEs shall consist of the following sequence of events:

a) *An LE Fires a Weapon.* The firing shall be communicated through the use of an LE Fire PDU.

b) *Munition Is Launched.* The launched munition shall be modeled by the simulation application controlling the munition. If tracking data are required for the munition, the munition shall be assigned a unique Entity Identifier by the firing entity's simulation application. In addition to issuing the LE Fire PDU, the simulation application modeling the munition's behavior shall issue TSPI PDUs for the munition according to the procedures for the use of the TSPI PDU (see 9.4.2). If tracking data are not required for the munition, the Munition ID field in the LE Fire PDU shall be NO_SPECIFIC_ENTITY, and no TSPI PDUs shall be issued for the munition. Tracking data should be provided for a munition if representing its travel between firing and impact would affect the outcome of the simulation, e.g., if simulation entities can detect and react to the munition during the munition's travel.

c) *Impact or Detonation Is Communicated.* The impact or detonation of a munition shall be represented through the use of an LE Detonation PDU. If the munition is not represented as an entity, detonation represents the end of its path. When a munition is represented as an entity, the termination of the existence of the munition entity is determined by the State bit in the Appearance-Visual field set to Deactivated (1) in the munition entity's Appearance PDU.

9.4.5.2 Information contained in the LE Fire PDU

The LE Fire PDU shall be used to communicate information associated with the firing of a weapon by an LE and shall contain the following information, effective at the instant the munition enters the simulated world, subject to the settings in the flag field:

a) Identification of the entity that issued the PDU.
b) Flags to specify the existence of optional data fields. The format of the LE Fire Flag field shall be as shown in Table 225.
c) Identification of the intended target entity if known to the simulation application. (Otherwise, this field shall not be included in the LE Fire PDU.)
d) Identification of tracked munitions. (Munitions not tracked are flagged as not having a munition identification field.)
e) Identification of the specific event marked by the firing of an entity's weapon.
f) Information required for representation of the path and impact of the munition, including:
    1) Location from which the munition was launched or fired. (If this information is flagged as not included in the PDU, then the dead reckoned location of the firing entity shall be used.)
    2) Type of munition fired.
    3) Warhead of the munition if applicable. (Otherwise, the field shall not be included in the PDU.)
    4) Fuse employed by the munition if applicable. (Otherwise, the field shall not be included in the PDU.)
    5) Quantity and rate at which the munition was fired. (If this field is flagged as not included in the PDU, a single round shall be assumed to have been fired.)
    6) Initial velocity of the munition when visible effects of the launch first become apparent.
    7) Range (three-dimension, straight-line distance) that the firing entity's fire control system has assumed for computing the fire control solution. (If this value is not available, the field will be flagged as not included in the PDU.)

Table 225—LE Fire Flag field definition

Bit	State	Flag Octet 1 meaning
7	0	Reserved for flag continuation field
6	0	Location field from which fire event occurs is not included
	1	Location field from which fire event occurs is included
5	0	Quantity and Rate fields of the Munition Descriptor record are not included
	1	Quantity and Rate fields of the Munition Descriptor record are included

Table 225—LE Fire Flag field definition *(continued)*

Bit	State	Flag Octet 1 meaning
4	0	Warhead and Fuse fields of the Munition Descriptor record are not included (use munition default)
	1	Warhead and Fuse fields of the Munition Descriptor record are present
3	0	Site Number and Application Number data are the same as the firing entity's and are not included in the Event ID
	1	Site Number and Application Number fields are included in the Event ID
2	0	Munition ID field is not included
	1	Munition ID field is included
1	0	Site Number and Application Number data are the same as the firing entity's and are not included in the Munition ID
	1	Site Number and Application Number fields are included in the Munition ID
0	0	Target Entity ID field is not included
	1	Target Entity ID field is included
Total LE Flag field size = 8 bits		

9.4.5.3 Issuance of the LE Fire PDU

The LE Fire PDU shall be issued by an entity at the moment it fires a weapon or as soon as practicable thereafter. If the number of rounds to be fired is not known when the first round clears the firing entity, then the number of committed rounds shall be sent. Subsequent LE Fire PDUs shall report the number of rounds in subsequent bursts or in half-second bursts in the case of continuous fire.

The LE Fire PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

9.4.5.4 Single rounds and bursts of fire

If the firing of the weapon represents a single round, the Quantity and Rate fields of the LE Fire PDU shall be marked as not included in the PDU, and a default of quantity of one and a rate of zero shall be understood. If the firing of a weapon or a group of weapons oriented in the same direction represents multiple rounds, the quantity and rate fields shall contain the quantity of munition fired and the rate at which it was fired, respectively. For LEs, the location for the group of weapons firing the burst of rounds shall be assumed to be the centroid of the aggregate unit firing the weapons.

9.4.5.5 Receipt of the LE Fire PDU

Upon receipt of a LE Fire PDU, a simulation application shall use the information therein to represent any necessary visual and aural effects produced by the firing of the weapon, whether it be a muzzle flash, noise, or smoke.

9.4.5.6 Format of the LE Fire PDU

The firing of a weapon shall be communicated by issuing an LE Fire PDU. The LE Fire PDU shall contain the following fields subject to the settings in the Fire Flag field:

a) *LE PDU Header.* This field shall contain data common to all LE Informative/Interactive PDUs. The LE PDU Header shall be represented by the LE PDU Header record (see 9.3.2).

b) *Live Firing Entity ID.* This field shall identify the firing entity. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall be only 8 bits long.

c) *Fire Flag.* This field shall identify those optional data fields that are being transmitted with the current PDU. This field shall be defined as an 8-bit record as specified in item b) in 9.4.5.2.

d) *Target Live Entity ID.* This optional field shall identify the intended target. If the intended target is unknown, this field shall not be included in the PDU; however, the Location field shall be present. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall be only 8 bits long.

e) *Munition Live Entity ID.* This optional field shall identify the fired munition if tracking data are required. If tracking data for the munition are not required, this field shall not be included in the PDU. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall be only 8 bits long. If the Site Number and Application Number fields are the same as those in the Firing Entity ID, they shall not be included; and the associated Fire Flag field bit shall be set to zero.

f) *Event ID.* This field shall contain an identification generated by the firing entity to associate related firing and detonation events. The structure of this field shall be represented by an Event Identifier record (see 6.2.33) except that the Site Number and Application Number fields shall be only 8 bits long. If the Site Number and Application Number are the same as those in the Firing Entity ID, they shall not be included, and the associated Fire Flag field bit shall be set to zero.

g) *Location.* This optional field shall specify the location, in relative coordinates, from which the munition was launched. This field shall be represented by a Relative World Coordinates record (see 9.3.3). This field is optional if the Target Entity ID is provided and mandatory if the Target Entity ID field is not included in the PDU. If the Location field is not included in the PDU, the firing location shall be the current dead reckoned location of the firing entity.

h) *Munition Descriptor.* This field shall describe the type of munition fired, warhead, fuse, quantity, and rate. This structure of this field shall be represented by a Munition Descriptor record (see 6.2.19.2). When used in the LE Information/Interaction protocol, the warhead, fuse, quantity, and rate fields of this record shall be optional.

i) *Velocity.* This field shall specify the velocity of the fired munition at the point when the issuing simulation application intends the externally visible effects of the launch (e.g., exhaust plume or muzzle blast) to first become apparent. The structure of this field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96] subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol. Each vector component shall represent velocity in decimeters per second.

j) *Range*. This field shall specify the range in meters that an entity's fire control system has assumed in computing the fire control solution. For systems where range is unknown or unavailable, this field shall contain the value zero. This field shall be represented by a 16-bit unsigned integer.

The format of the LE Fire PDU shall be as shown in Table 226.

Table 226—LE Fire PDU

Field size (bits)	LE Fire PDU fields	
96	LE PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 49
		Protocol Family—8-bit enumeration = 11
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		Subprotocol Number—8-bit enumeration
		Padding—8 bits unused
32	Firing Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Flags	0 (Reserved)—1-bit Boolean
		Location field from which fire event occurs is included—1-bit Boolean
		Quantity and Rate fields of the Munition Descriptor record are included—1-bit Boolean
		Warhead and Fuse fields of the Munition Descriptor record are included—1-bit Boolean
		Site Number and Application Number fields are included in the Event ID—1-bit Boolean
		Munition ID field is included—1-bit Boolean
		Site Number and Application Number fields are included in the Munition ID—1-bit Boolean
		Target Entity ID field is included—1-bit Boolean
$32F_0$	Target Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
$(16F_1 + 16)F_2$	Munition Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
$16F_3 + 16$	Event ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Event Number—16-bit unsigned integer

Table 226—LE Fire PDU *(continued)*

Field size (bits)	LE Fire PDU fields	
$64F_6$	Location	Reference Point—16-bit unsigned integer
		Delta-*X*—16-bit fixed binary
		Delta-*Y*—16-bit fixed binary
		Delta-*Z*—16-bit fixed binary
$64 + 32F_4 + 32F_5$	Munition Descriptor	Munition—64-bit Entity Type record
		Warhead—16-bit enumeration
		Fuse—16-bit enumeration
		Quantity—16-bit unsigned integer
		Rate—16-bit unsigned integer
48	Velocity	*X*-component—16-bit fixed binary
		Y-component—16-bit fixed binary
		Z-component—16-bit fixed binary
16	Range	16-bit unsigned integer
Total LE Fire PDU size = $280 + 32F_0 + (16F_1 + 16)F_2 + 16F_3 + 64F_6 + 32F_4 + 32F_5$ bits where F_i is the value of bit *i* in the LE Fire Flag field		

9.4.6 Live Entity (LE) Detonation PDU

9.4.6.1 Purpose

The LE Detonation PDU shall be used to communicate information associated with the impact or detonation of a munition.

9.4.6.2 Information contained in the LE Detonation PDU

The LE Detonation PDU shall contain the following information subject to the settings in the flag field:

a) Identification of the entity that issued the PDU.

b) Flags to specify the existence of optional data fields. The format of the Detonation Flag field shall be as shown in Table 227 and Table 228.

c) Identification of the target entity if an entity is impacted. (Otherwise, this data shall not be included in the PDU.)

d) Identification of tracked munitions. (Munitions not tracked shall not include this field in the PDU.)

e) Identification of the fire event responsible for the detonation. This information shall be the same as the Event ID from the corresponding LE Fire PDU. If the detonation is not preceded by a corresponding fire event, then the Event Number field of the Event Identifier record shall not be included in the PDU (e.g., land mine's detonation).

f) Information required for representation of the impact or detonation of the munition, including:

1) Location with respect to the world. (This field may be excluded if the targeted entity is specified at the location with respect to the world that can be obtained using the target entity's dead reckoned position.)

2) Velocity just before detonation/impact.

3) Orientation of the munition with respect to the target entity in the target entity's coordinate system at detonation.

4) Type of munition fired.

5) Warhead of the munition if it is not the default munition and fuse for the munition fired or otherwise not included in the PDU.

6) Fuse employed by the munition if it is not the default munition and fuse for the munition fired or otherwise not included in the PDU.

7) Quantity and rate at which it the munition was fired. If the quantity is one (thus, rate of zero), then this field shall not be included in the PDU.

8) Location of detonation with respect to the target entity in the target entity's coordinate system (present only if the Target Entity ID field is included in the PDU).

9) Detonation result.

Table 227—LE Detonation Flag-1 field definition

Bit	State	Flag Octet 1 meaning
7	0	Flag Octet 2 is not included
	1	Flag Octet 2 is included
6	0	Location in Entity Coordinates field is not included; Location in Relative World Coordinates field is included
	1	Location in Entity Coordinates field is included; Location in Relative World Coordinates field is not included
5	0	Quantity and Rate fields of the Munition Descriptor record are not included
	1	Quantity and Rate fields of the Munition Descriptor record are included
4	0	Warhead and Fuse fields of the Munition Descriptor record are not present (use munition default)
	1	Warhead and Fuse fields of Munition Descriptor record are included
3	0	Site Number and Application Number data are the same as the firing entity's and are not included in the Event ID
	1	Site Number and Application Number fields are included in Event ID
2	0	Munition ID field is not included
	1	Munition ID field is included
1	0	Site Number and Application Number fields are the same as the firing entity's and are not included in the Munition ID
	1	Site Number and Application Number fields are included in Munition ID

Table 227—LE Detonation Flag-1 field definition *(continued)*

0	0	Target Entity ID field is not included
	1	Target Entity ID field is included
Total LE Detonation Flag-1 field size = 8 bits		

Table 228—LE Detonation Flag-2 field definition

Bit	State	Flag Octet 2 meaning
7	0	Reserved for flag continuation flag
6	0	Unused
5	0	Unused
4	0	Unused
3	0	Unused
2	0	Unused
1	0	Event Number field is not included in Event ID
	1	Event Number field is included in Event ID
0	0	Munition Orientation field is not included
	1	Munition Orientation field is included
Total LE Detonation Flag-2 field size = 8 bits		

9.4.6.3 Issuance of the LE Detonation PDU

The LE Detonation PDU shall be issued by a simulation application at the moment that a munition being modeled by that simulation application impacts the terrain, a terrain feature, or another entity or at the moment the munition detonates. If the munition neither impacts nor detonates, the controlling simulation application shall issue a LE Detonation PDU with a detonation result of none when the controlling simulation application has ceased to model the munition.

The LE Detonation PDU shall be issued using a best effort multicast communication service. The PDU shall be sent to all simulation applications that are members of the addressed multicast group.

9.4.6.4 Inclusion of the Entity Identifier

If the impact or detonation is known to have affected only a specific entity that was physically contacted by the munition, a detonation result of *entity impact* shall be used. If the impact or detonation is known to have affected only a specific entity but that entity was not physically impacted by the munition prior to detonation, a detonation result of *entity proximate detonation* shall be used. In either case, the Entity ID of the specific entity affected shall be included in the Target Entity ID field of the LE Detonation PDU. The location of the impact or detonation in the affected entity's coordinates shall also be included. If the Target

Entity ID is not specified, then the location of the detonation in relative world coordinates (see 9.3.3) shall be used.

If the impact or detonation is known to have affected only the terrain and the terrain was physically contacted by the munition, a detonation result of *surface impact* shall be used. If the impact or detonation is known to have affected the terrain but the terrain was not physically contacted by the munition prior to detonation, a detonation result of *surface proximate detonation* shall be used. In either case, the location of the impact or detonation shall be communicated in relative world coordinates.

If neither a specific entity nor the terrain is affected by the munition, a detonation result of detonation shall be used if the munition detonates, and a detonation result of *none* shall be used if it does not. In either case, the Location in World Coordinates field shall hold the terminal location of the munition in relative world coordinates. The Location in Entity's Coordinates field and the Target Entity ID field shall not be included in the PDU.

9.4.6.5 Inclusion of articulated part parameters

No articulated part information of the target entity shall be included in the LE Detonation PDU.

Bandwidth constraints in DIS exercises involving LEs require that the articulated parts location not be transmitted in the LE Detonation PDU. The targeted entity shall determine whether the detonation impacted an articulated part or not.

9.4.6.6 Termination of the existence of munition entities

The termination of the existence of a munition entity shall be indicated by the issuance of an Appearance PDU with the State bit set to Deactivated (1) in the Appearance-Visual field.

9.4.6.7 Receipt of the LE Detonation PDU

Upon the receipt of an LE Detonation PDU, a simulation application shall use the information therein to represent the visual and aural effects that may be produced by the detonation or impact of the munition. The receiving simulation application shall also use the information to determine damage that may have been received as a result of the detonation. The simulation application shall issue a Appearance PDU to communicate any appearance changes resulting from the detonation.

9.4.6.8 Format of the LE Detonation

The detonation or impact of munitions shall be communicated by issuing a LE Detonation PDU. The LE Detonation PDU shall contain the following fields subject to the settings in the Detonation Flag field:

a) *LE PDU Header.* This field shall contain data common to all LE Information/Interaction PDUs. The LE PDU Header shall be represented by the LE PDU Header record (see 9.3.2).

b) *Firing Live Entity ID.* This field shall identify the firing entity. If the detonation is not preceded by an LE Fire PDU, then this field shall not be included in the current PDU. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall only be 8 bits long.

c) *Detonation Flags.* The flags field shall identify those optional data fields that are being transmitted in the current PDU. This field shall be defined as a 16-bit record as specified in item b) in 9.4.6.2.

d) *Target Live Entity ID.* This optional field shall identify the target entity. If the target identification is unknown, this field shall not be included in the PDU. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall only be 8 bits long.

e) *Munition Live Entity ID.* This optional field shall specify the entity identification of the fired munition if tracking data are required. This Munition ID shall correspond to the Munition ID specified in the LE Fire PDU that communicated the launch of the munition. A Munition ID field shall not be included in the current PDU if tracking data for the munition are not required. The structure of this field shall be represented by an Entity Identifier record (see 6.2.28) except that the Site Number and Application Number fields shall only be 8 bits long. If the Site Number and Application Number fields are the same as those in the Firing Entity ID, they shall not be included, and the associated Detonation Flag field bit shall be set to zero.

f) *Event ID.* This optional field shall contain the same data as in the Event ID field of the LE Fire PDU that communicated the launch of the munition. If the detonation is not preceded by a corresponding fire event, then the Event ID field shall not be included in the current PDU. The structure of this field shall be represented by an Event Identifier record (see 6.2.33) except that the Site Number and Application Number fields shall only be 8 bits long. If the Site Number and Application Number fields are the same as those in the Firing Entity ID, they shall not be included, and the associated Detonation Flag field bit shall be set to zero.

g) *Location.* This field shall specify the location of the detonation in a relative world coordinates. This field shall be represented by a Relative World Coordinates record (see 9.3.3) if the Target Entity ID field is not included in the PDU.

h) *Velocity.* This field shall specify the velocity of the munition immediately before detonation/impact. The velocity shall be represented in world coordinates. The structure of this field shall be represented by a Linear Velocity Vector record [see item c) in 6.2.96] subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol. Each vector component shall represent velocity in decimeters per second.

i) *Munition Orientation.* This optional field shall specify the orientation of the munition in entity-based coordinates at the time of detonation. The structure of this field shall be represented by an Euler Angles record (see 6.2.32) subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol.

j) *Munition Descriptor.* This field shall describe the type of munition fired, warhead, fuse, quantity, and rate. The structure of this field shall be represented by a Munition Descriptor record (see 6.2.19.2). When used in the LE Information/Interaction protocol, the warhead, fuse, quantity, and rate fields of this record shall be optional.

k) *Location in Entity's Coordinates.* This optional field shall specify the location of the detonation or impact in the target entity's coordinates. This information should be used for damage assessment. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.96] subject to the decreased field sizes and alternative data types used in the LE Information/Interaction protocol. If the Target Entity ID field is not included in the PDU, this field shall also not be included.

l) *Detonation Result.* This field shall specify the result of the detonation and shall be represented by an 8-bit enumeration (see [UID 62]).

The format of the LE Detonation PDU shall be as shown in Table 229.

Table 229—LE Detonation PDU

Field size (bits)	LE Detonation PDU fields	
96	LE PDU Header	Protocol Version—8-bit enumeration
		Exercise ID—8-bit unsigned integer
		PDU Type—8-bit enumeration = 50
		Protocol Family—8-bit enumeration = 11
		Timestamp—32-bit unsigned integer
		Length—16-bit unsigned integer
		Subprotocol Number—8-bit enumeration
		Padding—8 bits unused
32	Firing Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
8	Detonation Flag-1	1 (Flag Octet 2 included)—1-bit Boolean
		Location in Entity’s Coordinates field is included; Location in Relative World Coordinates field is not included—1-bit Boolean
		Quantity and Rate fields of the Munition Descriptor record are included—1-bit Boolean
		Warhead and Fuse fields of the Munition Descriptor record are included—1-bit Boolean
		Site Number and Application Number fields are included in Event ID—1-bit Boolean
		Munition ID field is included—1-bit Boolean
		Site Number and Application Number fields are included in Munition ID—1-bit Boolean
		Target Entity Number field is included—1-bit Boolean
$8F_7$	Detonation Flag-2	0 (Reserved)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		0 (Unused)—1-bit Boolean
		Event Number field is included—1-bit Boolean
		Munition Orientation field is included—1-bit Boolean

Table 229—LE Detonation PDU *(continued)*

Field size (bits)	LE Detonation PDU fields	
$32F_0$	Target Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
$(16F_1 + 16)F_2$	Munition Live Entity ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Entity Number—16-bit unsigned integer
$(16F_3 + 16)G_1$	Event ID	Site Number—8-bit unsigned integer
		Application Number—8-bit unsigned integer
		Event Number—16-bit unsigned integer
$64(1 - F_6)$	Location in World Coordinates	Reference Point—16-bit unsigned integer
		Delta-*X*—16-bit fixed binary
		Delta-*Y*—16-bit fixed binary
		Delta-*Z*—16-bit fixed binary
48	Velocity	*X*-component—16-bit fixed binary
		Y-component—16-bit fixed binary
		Z-component—16- bit fixed binary
$48G_0$	Munition Orientation	Psi (ψ)—16-bit fixed binary
		Theta (θ)—16-bit fixed binary
		Phi (ϕ)—16-bit fixed binary
$64 + 32F_4 + 32F_5$	Munition Descriptor	Munition—64-bit Entity Type record
		Warhead—16-bit enumeration
		Fuse—16-bit enumeration
		Quantity—16-bit unsigned integer
		Rate—16-bit unsigned integer

Table 229—LE Detonation PDU *(continued)*

Field size (bits)	LE Detonation PDU fields	
$48F_6$	Location in Entity Coordinates	x-component—16-bit fixed binary
		y-component—16-bit fixed binary
		z-component—16-bit fixed binary
8	Detonation Result	8-bit enumeration

Total LE Detonation PDU size =

$$256 + 8F_7 + 32F_0 + (16F_1 + 16)F_2 + (16F_3 + 16)G_1 + 64(1 - F_6) + 48G_0 + 32F_4 + 32F_5 + 48F_6 \text{ bits}$$

where

F_i is the value of bit i in the LE Detonation Flag-1 field
G_i is the value of bit i in the LE Detonation Flag-2 field

$$G_i = \begin{cases} 1 \text{ if } F_7 \text{ is 1 and } G_i \text{ is 1} \\ 0 \text{ if } F_7 \text{ is 0 or } G_i \text{ is 0} \end{cases}$$

Annex A

(normative)

Warfare

A.1 Scope

This annex defines the specific implementation requirements for various kinds of weapons, expendables, and explosions including the proper sequence of Protocol Data Units (PDUs) to represent various effects. This annex is normative in nature, and the requirements contained therein are treated similar to requirements found in the other clauses of the standard.

A.2 Weapons

A.2.1 Introduction

The requirements for specific types of weapons and specific conditions of their employment are defined in this annex.

A.2.2 Missiles

A.2.2.1 Introduction

This subclause addresses the requirements related to representing single-stage and multistage missiles. These include, but are not limited to, tactical ballistic and interceptor missiles. A ballistic missile shall fly a ballistic trajectory. Ballistic missiles with maneuvering capability shall represent such maneuvers. Nonballistic missiles shall fly an appropriate trajectory based on the type of missile.

A.2.2.2 Single-stage missiles and rockets

This subclause contains the requirements for single-stage ballistic and nonballistic missiles:

a) Single-stage missiles shall be represented by an entity for which Entity State PDUs shall be issued.
b) All single-stage missiles shall have a Fire PDU and a Detonation PDU issued for them.
c) A single-stage missile that reaches the end of its planned trajectory shall comply with the requirements specified in A.2.2.3.4.

A.2.2.3 Multistage missiles and rockets

A.2.2.3.1 General

This subclause contains the requirements for multistage ballistic and nonballistic missiles and rockets. Although these requirements are contained in the Warfare section, they also include simulating rockets such as multistage National Aeronautics and Space Administration (NASA) vehicles and commercial operations to launch satellites. The stages and any other parts that will normally separate during flight are referred to as components. They may be represented as Attached Part Variable Parameter (VP) records when there is a need to have precise spatial positioning information when components separate during flight. The final component is referred to as the final part. The center of the bounding volume of the final part represents the exact point for describing the spatial position for the missile or rocket entity. Following standard Distributed Interactive Simulation (DIS) spatial positioning rules, this point does not move in relation to the entity body coordinates. For example, the Saturn V rocket stages, the Service Module, and Command Module are components of the Apollo Rocket, and each would be represented as Attached Part VP records. Table A.1 indicates a typical sequence of events for a multistage missile flight.

Table A.1—Multistage missile flight

Step	Event	PDUs	VP records
1	Prelaunch	*Launcher* Entity State	Attached Part VP record if the missile is not represented as a separate entity while on the launcher.
2	Missile launched	*Launcher* Entity State	Attached Part VP record for the missile, if present, is set to detached and removed.
		Fire PDU	NA (The Fire PDU does not have VP records).
		Missile Entity State	— Attached Part VP records[a] representing the stages that will separate during flight. — An Attached Part VP record[a] representing the final part. — Entity Separation record if the station on the launcher needs to be identified.
3	Booster separates	Parent missile Entity State	Booster Attached Part VP records[a] removed, if present. Entity Type VP record included.
		Booster Entity State	Separation VP record.
4	2nd stage separates	*Parent missile* Entity State	Entity Type VP record updated and 2nd Stage Attached Part VP records[a] removed, if present.
		2nd stage Entity State	Separation VP record.
5	Last stage separates	*Parent missile* Entity State	Entity Type VP record updated and Warhead Attached Part VP records[a] removed, if present.
		Last stage Entity State	Separation VP record.
6	Warhead Impacts	Detonation PDU	No VP records are required.
		Warhead Entity State	Warhead entity is canceled, and fragment entities may be created.
7	Warhead Fragments	Entity State PDUs	Separation VP records for each fragment entity indicating the warhead entity as the parent entity.
8	Any component or the warhead reaches the Earth's surface	Entity State PDUs[b]	

[a]Attached Part VP records for components are only required if the simulation requires a precise spatial position to be known at all times as defined in item a3) in A.2.2.3.3.

[b]Either *final* Entity State PDUs are issued or the entities may remain.

A.2.2.3.2 Multistage missile entity—no component separations

If a multistage missile is modeled as a single entity for its entire flight with no separate entities created to represent the planned separation of components, then the following requirements shall apply:

a) The Entity Type contained in the Entity State PDU shall reflect the complete missile with all its components (e.g., all its stages).

b) A Fire PDU shall be issued prior to and coincident with the issuance of the *initial* Entity State PDU for the missile. If it was not launched from another entity, the Firing Entity Identification (ID) shall be set to NO_SPECIFIC_ENTITY.

c) If the parent entity had included the missile as an Attached Part VP record, the Detached Indicator in the record shall be set to Detached (1).

d) If the simulation knows the station that the missile was launched from, a Separation VP record with that information shall be included in the Entity State PDU update for the parent entity.

e) An Entity State PDU shall be issued for the parent entity if there is any changed data.

A.2.2.3.3 Missile entity—component separations

If a multistage missile or rocket vehicle is modeled with separate entities created to represent the separation of components, then the following requirements shall apply:

a) The *initial* Entity State PDU for the multistage vehicle (missile or rocket) shall contain the following information:

   1) The Entity Type field shall reflect the entire missile or rocket with all its components (e.g., all its stages) and shall not change as stages are jettisoned.

   2) If the missile or rocket is fired from a parent entity, and the simulation knows the station on that entity that the missile was launched from, a Separation VP record with that information may be included in the *initial* Entity State PDU.

   3) If the simulation determines that Attached Part VP records are required to determine vehicle configuration as stages are jettisoned for tracking purposes, the following requirements shall apply:

      i) An Attached Part VP record shall be included for each component that is visible and that would normally separate during flight, as well as for the component that is designated as the final part. The Attached Part VP record that represents the final part component shall remain with the entity for the entire flight.

      ii) The SISO-REF-010 document will include Entity Type enumerations for components that can separate, as well as an Entity Type for each configuration of remaining components. Other components that will not separate and fragments may also have Entity Type enumerations. The Entity Type representing the final part of a multistage missile or rocket will be annotated in SISO-REF-010 to indicate it is the part whose center of the bounding volume is used to represent the vehicle position.

b) A Fire PDU shall be issued prior to and coincident with the issuance of the *initial* Entity State PDU for the missile. If it was not launched from a parent entity, the Firing Entity ID shall be set to NO_SPECIFIC_ENTITY.

c) If the parent entity had included the missile as an Attached Part VP record, it shall be removed.

d) The separation of a component shall occur as follows:

   1) An *initial* Entity Type VP record shall be included in the Entity State PDU for the parent entity, if not already included, to reflect the remaining composition of the missile. It shall replace any previous Entity Type VP record. Once an Entity Type VP record is included, it shall remain for the life of the entity and may be updated to reflect a different Entity Type as required.

   2) If the parent entity included an Attached Part VP record for this component, the Attached Part VP record shall be removed.

3) The separated component shall be represented by an entity. A Separation VP record shall be included in the first Entity State PDU to indicate, as a minimum, the parent entity from which it separated.

e) Receiving simulations will need to process the Attached Part VP records representing components that may separate and the final part component if they require precise spatial position information. Other simulations that use the Entity Type VP record to determine the present missile or rocket composition need to check for such a record, and changes in the record, when they receive an Entity State or Entity State Update PDU for the missile or rocket.

A.2.2.3.4 End of trajectory

When a single or multistage missile reaches the end of its planned trajectory or that trajectory is interrupted, the following requirements shall apply:

a) If the missile reaches the end of its planned trajectory:
   1) A Detonation PDU shall be issued.
   2) A *final* Entity State PDU shall be issued for the missile.
   3) If missile fragmentation is modeled, Entity State PDUs shall be created to represent the fragments:
      i) No Fire PDU or Detonation PDU shall be associated with a fragment entity.
      ii) A Separation VP record shall be included in the *initial* Entity State PDU for a fragment entity.
b) If a missile does not reach the end of its planned trajectory due to being intercepted or for other reasons such as structural failure, the following requirements shall apply:
   1) A Detonation and *final* Entity State PDU shall be issued for the original missile entity.
   2) Fragment entities may be created as specified in item c) below.
c) Each fragment shall be represented by an entity as follows:
   1) The *initial* Entity State PDU for each fragment shall contain a Separation VP record.
   2) Nothing in this standard shall preclude a fragment entity from breaking into more fragment entities.
   3) No Detonation PDU shall be issued for a fragment unless it explodes.

A.2.3 Submunitions

A.2.3.1 Introduction

Submunitions are defined by the DOD and the North Atlantic Treaty Organization (NATO) as "any munition that, to perform its task, separates from a parent munition." The representation of submunitions in DIS takes into account the need for accountability of the submunition parts, their relationship to each other, visual appearance considerations, and minimizing the number of PDUs involved from the time the initial submunition dispenser is launched until the time that all the individual bomblets either explode or become duds. Figure A.1 illustrates a typical submunition deployment, and Table A.2 indicates the PDUs that are required for each phase of a submunition engagement.

A.2.3.2 General requirements

The following requirements shall apply to a submunition:

a) The terrain or water surface should be modeled for the area in which the submunitions will be dispensed.

b) A component of a submunition that is required to be tracked by a sensor or seen visually shall be represented by an entity when separated from other components.

c) A Fire PDU shall be issued for the initial submunition weapon entity when it separates from the parent entity that is the launch platform:

   1) Subsequent separations of weapon container components (e.g., canisters or dispensers containing submunitions that will be released) shall be indicated by the inclusion of a Separation VP record in the *initial* Entity State PDU for that component.

   2) If there is a need to indicate a launch flame, the Launch Flash Appearance field shall be set to Launch Flash Present (1) in the submunition component entity.

d) A submunition component that represents the final projectile or bomblets that is intended to explode or penetrate a target as a kinetic energy weapon shall have a Fire PDU and a Detonation PDU issued for it:

   1) This requirement shall apply whether or not the final projectile or bomblet is represented as an entity.

   2) A Fire PDU shall be issued for each projectile or bomblet in order to allow visualization of the smoke that is associated with it at the simulation time it is fired. The Fire PDU fields shall be set as follows:

      i) The Firing Entity ID field shall be the entity that it was directly fired from if an entity existed. If not, the Firing Entity ID shall be the Entity ID of the last entity containing the fired projectile or bomblet.

      ii) The Location field shall be the location of the projectile or bomblet when it separated from its component. For example, in Figure A.1, the skeet is the component that contains the individual projectiles. The skeet may or may not have been modeled as an entity.

      iii) The Target Entity ID shall contain the Entity ID of the target, if known. Otherwise this field shall be set to TARGET_ID_UNKNOWN.

      iv) The Munition ID shall contain the Entity ID of the projectile or bomblet if it is modeled as an entity. Otherwise this field shall be set to NO_SPECIFIC_ENTITY.

      v) The Event ID shall be set to the Event ID contained in the first Fire PDU that launched the initial submunition entity from a parent entity.

      vi) Other fields shall be set to appropriate values.

   3) The Detonation PDU shall be populated as follows:

      i) The Detonation Result shall be set to an appropriate value in the range of 1 to 6. If it is determined that the projectile or bomblet is a dud, a Detonation Result of Dud shall be set regardless of whether there was an entity for the projectile or bomblet.

      ii) Other fields shall be set as appropriate.

   4) Representation of dud projectiles and dud bomblets:

      i) If the final projectile or bomblet was represented as an entity, and the Detonation PDU indicates that it is a dud, the simulation may either cancel the entity or it may remain active in the exercise.

      ii) If there was no entity for the projectile or bomblet, and the Detonation PDU indicates that it is a dud, an entity may be created and an Entity State PDU issued for it. If this occurs, the Entity State PDU shall be issued simultaneously with the issuance of the Detonation PDU. A Separation VP record shall be included in the *initial* Entity State PDU for the dud entity with the Parent Entity field set to the last entity that was portrayed that contained this projectile.

A.2.3.3 Sequence of events

Figure A.1 shows a typical submunition sequence of events. Table A.2 indicates the PDUs and records associated with each event in the sequence shown in Figure A.1.

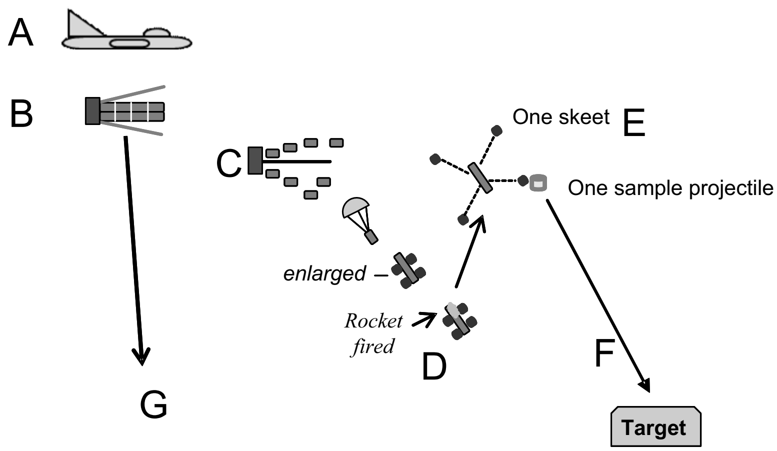


Figure A.1—Submunition simulation

Table A.2—Submunition simulation DIS PDUs for Figure A.1

Seq.	Entity	Event	PDUs	Records/actions
1	A	Fighter aircraft with smart weapon	*Fighter aircraft* Entity State	(Optional) Attached Part VP record—Tactical Munitions Dispenser.
2	B	Launch Smart Weapon *Tactical Munitions Dispenser (TMD)*	Fire	Indicates launched weapon (TMD).
			Fighter aircraft Entity State	Attached Part VP record—Indicates detached in Entity State PDU.
			Initial TMD Entity State	Separation VP record included if station the TMD launched from is known.
3	B	TMD Opens to release canisters	*TMD* Entity State	None required.
4	C × 10	Submunition Canisters (SC) released	*Initial SC* Entity State × 10	Separation VP record.
5	D × 10	SC Opens 4 skeets—rocket fires	*SC* Entity State × 10	Entity Appearance record—Power on, burst flames present.
6	E	Skeets ejected	*Skeet* Entity State PDU × 4	Separation VP record.
			No Skeet Entity State PDUs	None required.

Table A.2—Submunition simulation DIS PDUs for Figure A.1 *(continued)*

Seq.	Entity	Event	PDUs	Records/actions
7	F	Projectiles fired	*Projectile entities (optional)*— Entity State PDU × 4 Fire PDU × 4	None required.
			No Projectile entity created Fire PDU × 4	None required.
8	G	Projectiles detonate	*final* Entity State PDU for each projectile entity, if present Detonation PDU—each projectile	None required.
			No Projectile entity created Detonation PDU	None required.
9	B	TMD falls to Earth	*TMD* Entity State	None required.
10	F	TMD crashes on land or water.	*TMD* Entity State	TMD entity either remains as debris or is canceled.

A.2.4 Directed energy

A.2.4.1 Types of directed energy weapons

There are many types of directed energy weapons. The DE Fire PDU and Entity Damage Status PDU support all types. Additional Aimpoint record types may be developed, as needed, to provide information unique to a specific type of Directed Energy (DE) engagement. Table A.3 provides a list.

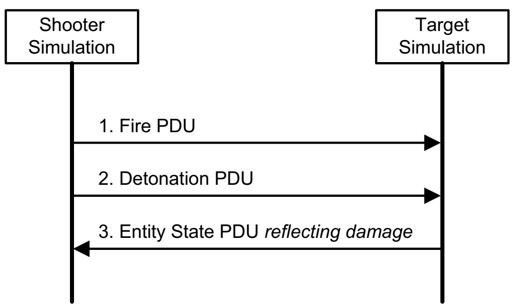
Table A.3—Directed energy weapons

DE weapon type	Examples
High-Energy Laser (HEL)	Airborne Laser (ABL) Advanced Tactical Laser (ATL) Mobile Tactical High-Energy Laser (MTHEL)
Low-Energy Laser (LEL)	Personnel Halting and Stimulation Response (PHASR) rifle Laser dazzlers
High-Power Microwave (HPM)	Active Denial System (ADS)
Directional Acoustic	Acoustic beam weapons Sequential Arc Discharge Acoustic Generator

A.2.4.2 Multiresolution considerations

Although the DE Fire PDU and the Entity Damage Status PDU are the preferred PDUs to communicate information associated with the firing of a DE weapon and its impact on a target entity or other object, not all simulations involved in a DE engagement may have implemented the high-fidelity solution as lower fidelity directed energy implementations existed prior to the introduction of these two PDUs. This means that a multiresolution synthetic environment is likely to exist as regards directed energy engagements with some shooters and target simulations having implemented the high-fidelity DE simulation and some the basic DE simulation. Figure A.2 defines the sequence of events for a basic DE engagement and Figure A.3 for a high-fidelity engagement. The high-fidelity DE engagement supports realistic portrayal of the effects of a DE weapon on a target during the engagement, as well as after the firing ceases. During a DE engagement, the results of a DE weapon being fired will usually be immediately noticeable when the firing begins. The operator of a high-fidelity shooter simulation requires immediate and continuous feedback of changes to a target while a shot is active as does personnel associated with the target entity if it is, for example, a virtual simulator.

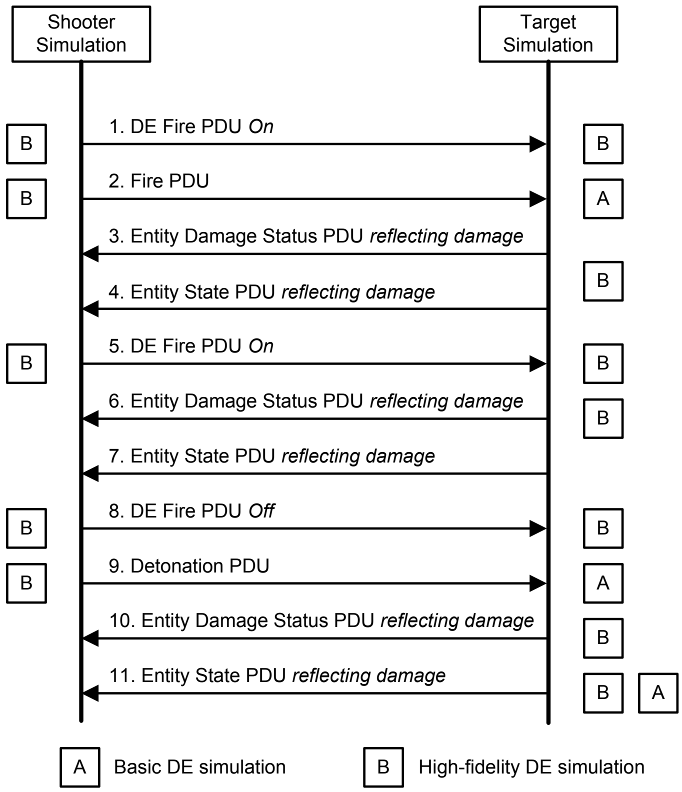
Feedback of detailed visual damage caused by a DE weapon is provided by the Entity Damage Status PDU. Other damage results are conveyed by changes to applicable Appearance fields and by Variable Parameter records contained in the Entity State PDU.



NOTES

1. The Fire PDU is sent once when the DE weapon is initially fired.
2. The Detonation PDU is sent once when the DE shot is terminated.
3. The target simulation outputs an Entity State PDU to reflect the Entity State PDU to reflect the damage assessment based on receiving the Detonation PDU.

Figure A.2—Basic directed energy shot sequence



NOTES

1 and 2. The DE Fire PDU and Fire PDU are sent once when the DE weapon is initially fired. A high-fidelity DE Simulation shall process the DE Fire PDU and a basic DE simulation may process the Fire PDU. Thereafter a DE Fire PDU is sent during the engagement based on the DE Fire PDU issuance rules. (5)

3. A high-fidelity DE target simulation shall reflect any damage during the engagement by issuing Entity Damage Status PDUs, as well as, issuing an Entity State PDU if applicable. (3, 4, 6, 7). There is no requirement to send the Entity Damage Status PDU or an Entity State PDU as a result of receiving a DE Fire PDU unless damage, or a change in damage, is determined.

8 and 9. A DE Fire PDU with the Flags record Weapon on/off state field (bit 0) set to Weapon Off (0) and a Detonation PDU are sent simultaneously at the end of the engagement.

Figure A.3—High-fidelity directed energy shot sequence

Table A.4 specifies the requirements for a simulation to transmit and receive PDUs associated with a directed energy engagement based on whether they are a high-fidelity DE simulation or have only a basic DE capability.

Table A.4—Directed energy PDU requirements

PDU	DE engagement				Notes
	Shooter		Target		
	Hi-Fi	Basic	Hi-Fi	Basic	
DE Fire	T	NT	R	NR	
Fire	T	T	RO	RO	See item a) in A.2.4.2
Detonation	T	T	R	R	See item b) in A.2.4.2
Entity Damage Status	R	NR	T	NT	

Legend
T—Transmit required NT—Not Transmitted
R—Receipt required NR—Not Received RO—Receipt Optional

The following additional requirements shall apply to DE simulations:

a) A simulation that transmits the DE Fire PDU shall output a single Fire PDU immediately following the initial DE Fire PDU that is sent for a DE engagement.

b) The Detonation PDU shall be output once per DE engagement when the engagement is terminated.

A.2.4.3 DE Precision Aimpoint record

Implementation of targeting and damage assessment algorithms for some precision DE weapons will include a three-dimensional geometric model of the target. In such cases, dead reckoning and network latencies will introduce positioning errors that will require "clamping" of the dead-reckoned target spot onto the geometric model representing the target entity. A similar problem will manifest itself if the firing and target simulations use dissimilar geometric models for the target. Subclause 5.4.5.4 describes the rules governing the clamping of the target spot onto the target model.

The DE spot at the target is characterized by the beam spot size and irradiance profile shape in the plane normal to the beam path. Typically, the irradiance profile is a Gaussian shape whose peak is given in the Peak Irradiance field. Its width is characterized by the distance from the center at which the irradiance level is the product of $1/e^2$ and the peak irradiance. This distance is at $\sqrt{2}\sigma$, where σ is the standard deviation of the Gaussian function.

The beam spot can be circular or elliptical. An elliptical shape can result from beam spreading phenomena driven by the transverse velocity of the target or by dithering the beam in one direction. The target velocity-induced spread is oriented in the direction of the transverse velocity, but the dither spread might be in another direction. The resultant elliptical shape will have some orientation with a major and minor axis width; thus, two sizes shall be specified as the Beam Spot Cross-Section Semi-Major Axis and Beam Spot Cross-Section Semi-Minor Axis.

The orientation of the ellipse is specified by the Beam Spot Cross-Section Orientation Angle field. The beam coordinate system is a right-handed Cartesian coordinate system with the center of the beam along the *x*-axis of the system (Figure A.4). The beam coordinate system is constructed as:

— The X-axis is the vector from the firing entity aperture to the target spot.

— The Y-axis is the local "down" vector at the target spot crossed into the X-axis.

— The Z-axis is the X-axis crossed into the Y-axis.

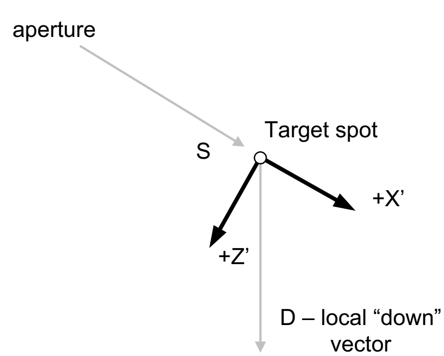


Figure A.4—The beam coordinate frame (X'Y'Z')

In purely mathematical terms, this can be expressed as three unit vectors of the form:

$$X' = \frac{S-A}{|S-A|} \tag{A.1}$$

where S is the target spot location and A is the aperture location.

$$Y' = \frac{D' \times X'}{|D|} \quad \text{if} \quad \frac{D' \times X'}{|D|} \neq \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{; otherwise} \quad Y' = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \tag{A.2}$$

$$Z' = X' \times Y' \tag{A.3}$$

A special provision is required in Equation (A.2) so cases where the beam is parallel to the local down vector still generate a valid solution.

The Beam Spot Cross-Section Orientation Angle, ϕ, reflects the angle in the Y'Z' plane from the Z-axis to the semi-major axis of the cross-section ellipse (see Figure A.5).

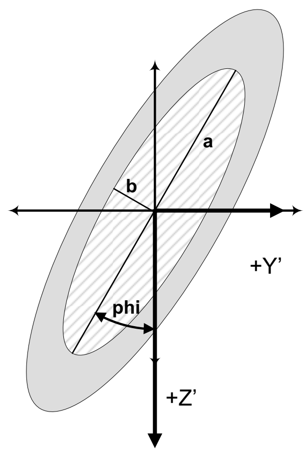


Figure A.5—Beam spot cross-section orientation angle

The irradiance at a point on the Y'Z' plane is then given by:

$$I(y,z) = I_{Peak} e^{-\left(\frac{y_c^2}{r_y^2} + \frac{z_c^2}{r_z^2}\right)} \tag{A.4}$$

where

$$z_c = z\cos(-\phi) - y\sin(-\phi)$$

$$y_c = z\sin(-\phi) + y\cos(-\phi)$$

where y, z is the point of interest in the y–z plane (the plane normal to the line-of-sight) of the beam coordinate frame, and r_y and r_z are the Azimuth Beamwidth and Elevation Beamwidth, respectively.

Annex B

(normative)

Specific transponder and interrogator systems

B.1 Scope

B.1.1 General

This annex defines the information, issuance, and receipt rules for specific transponder and interrogator systems. A normative annex represents requirements that shall be adhered to as if they were in the body of the standard. They are placed here because of the amount of information being presented. The term "Identification Friend or Foe (IFF)" is used throughout this annex when referring to requirements that are not unique to a specific transponder or interrogator system. The transponder and interrogator systems supported by the IFF Protocol Data Unit (PDU) are those used to perform surveillance, tracking, and identification of civilian and military aircraft, ships, and vehicles. This includes the Air Traffic Control Radar Beacon System (ATCRBS), Identification Friend or Foe (IFF), and Mark X/XII/XIIA systems, collectively known by the acronym AIMS, as well as other systems such as Soviet systems and unique systems in limited use. ATCRBS includes civilian modes such as Mode A, C and Mode Select (Mode S). The term "system" and "system type" are used interchangeably in this annex unless otherwise indicated.

The implementation of information exchange requirements and their associated issuance and receipt rules for interrogator and transponder systems contained in this standard also requires a knowledge of such systems as are contained in civilian and military standards, specifications, and other publications. Those publications that have to be understood and used to implement this standard are referenced in this annex and listed in Clause 2.

B.1.2 Overview

The term "transponder," as used in conjunction with the IFF PDU, is defined as a device used for the surveillance, tracking, and identification of aircraft, ships, and vehicles, which has the following characteristics:

a) A device that transmits a signal on a radio frequency without being interrogated.
b) A device that transmits a predetermined message in response to a predefined received signal.
c) A receiver-transmitter that generates a reply signal upon proper electronic interrogation.

The term "interrogator," as used in conjunction with the IFF PDU, is defined as a device that sends an electronic signal to which a transponder will reply.

The types of transponders and interrogators currently implemented using the IFF PDU include:

— Aircraft, ship, and vehicle military transponders
— Aircraft and ship civilian transponders

Although the IFF PDU is currently used to support transponders and interrogators associated with military and civilian aircraft and naval ships, its design supports the inclusion of additional transponders and interrogators such as the Emergency Locator Transmitter (ELT) and the Automated Identification System (AIS) transponders used to track civilian vessels.

Each system type and system name that uses the IFF PDU and for which there is a corresponding enumeration shall require the inclusion of a subclause in this annex. The subclause will identify the purpose of the system, associated information content implementation requirements, and issuance and receipt rules. Where a system type has the same requirements as another system type, only the differences need be specified. Exemptions from any of the transponder and interrogator requirements specified in 5.7.6 and 7.6.5 shall also be noted. System names shall be identified with the system types to which they apply. System names include both generic system names and names that identify specific models of an IFF system made by different manufacturers. There may be a system name that identifies a model number as well as a version number for the model.

Table B.1 lists all the transponder and interrogator system types and system names associated with this edition of the standard. New system types and system names introduced between revisions to this standard will be processed by the working group responsible for maintaining this standard and reviewed to maintain proper integration with existing IFF systems.

Table B.1—System types and names

System type		System name		Ref.
Enum value	Title	Enum value	Title	
1	Mark X/XII/ATCRBS Transponder			B.5.2
	Mark X	1 17 18	Generic Mark X Generic Mark X (A) Generic Mark X (SIF)	B.5.3.2
		6 7	Generic Mark X/XII/ATCRBS Generic Mark X/XII/ATCRBS/Mode S	B.5.2
	Mark XII	2	Generic Mark XII	B.5.2.3
		6 7	Generic Mark X/XII/ATCRBS Generic Mark X/XII/ATCRBS/Mode S	B.5.2
	ATCRBS	3	Generic ATCRBS	B.5.2.4
		6 7	Generic Mark X/XII/ATCRBS Generic Mark X/XII/ATCRBS/Mode S	B.5.2
2	Mark X/XII/ATCRBS Interrogator			B.5.3
	Mark X	1 17 18	Generic Mark X Generic Mark X (A) Generic Mark X (SIF)	B.5.3.2
		6 7	Generic Mark X/XII/ATCRBS Generic Mark X/XII/ATCRBS/Mode S	B.5.3
	Mark XII	2	Generic Mark XII	B.5.3.3
		6 7	Generic Mark X/XII/ATCRBS Generic Mark X/XII/ATCRBS/Mode S	B.5.3
	ATCRBS	3	Generic ATCRBS	B.5.3.4
		6 7	Generic Mark X/XII/ATCRBS Generic Mark X/XII/ATCRBS/Mode S	B.5.3
3	Soviet Transponder	4	Generic Soviet	B.5.4

Table B.1—System types and names *(continued)*

System type		System name		Ref.
Enum value	Title	Enum value	Title	
4	Soviet Interrogator	4	Generic Soviet	B.5.5
5	Reply Receiver "B" (RRB) Transponder	10 8 9	Generic RRB Aircraft Radio Installation (ARI) 5954 ARI 5983	B.5.6
6	Mark XIIA Interrogator	11	Generic Mark XIIA	B.5.7
7	Mode 5 Interrogator	12	Generic Mode 5	B.5.8
8	Mode S Interrogator	5	Generic Mode S	B.5.9
9	Mark XIIA Transponder	11	Generic Mark XIIA	B.5.10
10	Mode 5 Transponder	12	Generic Mode 5	B.5.11
11	Mode S Transponder	5	Generic Mode S	B.5.12
12	Mark XIIA Combined Interrogator/Transponder (CIT)	13	Generic Mark XIIA Combined Interrogator/Transponder (CIT)	B.5.13
13	Mark XII Combined Interrogator/Transponder (CIT)	14	Generic Mark XII Combined Interrogator/Transponder (CIT)	B.5.14
14	Traffic Collision Avoidance System (TCAS)/Airborne Collision Avoidance System (ACAS) Transceiver	15	Generic TCAS I/ACAS I Transceiver	B.5.15
		16	Generic TCAS II/ACAS II Transceiver	

B.1.3 System operation

B.1.3.1 General

A brief overview of the system types and associated modes is provided here. Technical details regarding the implementation and use of specific transponders and interrogators in actual operations may be obtained from appropriate civilian and military technical publications. The International Civil Aviation Organization (ICAO) is the primary source of information for civilian systems. Various DoD, North Atlantic Treaty Organization (NATO), and other country military organizations are the sources for military systems. Some of the information may be restricted or otherwise classified.

Additional transmit and receive requirements, and specific Mode 5 tactical data message formats, are contained in the AIMS 03-1000A Technical Standard (see Clause 2). Only military Mode 5/S information releasable to the public is included in this standard. This standard supports interoperability between simulated military Mode 5/S transponders and interrogators capable of interrogating either Mode 5 or Mode S. The "(U) Mark XIIA Mode 5 and Mode Select (Mode S) Joint Concept of Operations (Joint CONOPS)" [B6] publication provides guidance for the employment of military Mode 5/S systems in joint operations (see Annex J).

This standard also supports other types of IFF systems. The IFF PDU is designed to allow for the inclusion of additional IFF data records by governments and other organizations to support new transponder and interrogator requirements.

The civilian Mode S technical specifications are contained in a series of International Civil Aviation Organization publications. A list of basic Mode S-related documents is grouped in Clause 2 under the heading International Civil Aviation Organization (ICAO) basic Mode S-related publications.

The following is a brief overview of common systems and modes associated with ATCRBS, IFF, and Mark X/XII/XIIA (AIMS) and other transponders and interrogators.

NOTE—The U.S. military and coalition countries identify IFF modes using a number system, while the international civilian world uses a letter system. Some modes are only applicable to the military and, therefore, do not have equivalent civilian aviation letter designations. These include Mode 1, 2, 4, and 5. Mode 5 was originally the military designation for the civilian Mode C (encoded altitude), but that number was reused for the replacement mode for Mode 4. Military aircraft flying in controlled civilian airspace are required to comply with civilian transponder capabilities for the type of aircraft and airspace. Most military aircraft, therefore, have Mode 3/A and Mode C capability in addition to their military modes. (A given military aircraft may not have all the possible military mode capabilities.) Military aircraft that will be flying in civilian airspace where Mode S is required are being upgraded with that additional civilian mode.

B.1.3.2 Systems

The following is a brief overview of common transponder systems present in a distributed simulation environment. There is not always a direct correlation between interrogators and transponders of the same system type. For example, a Mark XII interrogator is able to interrogate some or all of the modes associated with Mark X, Mark XII, and Mark XIIA transponders although some of the code replies may not be processed correctly. A Mark XII interrogator may have the capability to interrogate Mode S although a Mark XII transponder will not have a Mode S capability. Likewise, a Mark XIIA interrogator is not required to be able to interrogate Mode S, although many models will, while a Mark XIIA transponder is required to transmit Mode S. The only way to know the full capabilities of a given transponder or interrogator is by reference to a manufacturer's specifications for it. The IFF PDU contains fields and records to enable all the capabilities of a given transponder or interrogator system for a specific model to be defined. This may require the addition of new IFF data records:

a) *Mark X System*. The Mark X System is the oldest military system. Mark X transponders only reply with an indication of having Mode 1, 2, or 3 but without numeric codes, and Ident (I/P) and Military Emergency. There are no known U.S. or coalition military aircraft that currently use the Mark X system.

b) *Mark X (SIF) System*. This system consists of the following capabilities: Mode 1 (32 codes), Mode 2 (4096 codes), and Mode 3/A (64 codes), along with Ident (I/P) and Military Emergency. There are few, if any, U.S. or coalition military aircraft that currently use the Mark X (SIF) system.

c) *Mark X (A) System*. This system consists of the following capabilities: Mode 1 (32 codes), Mode 2 (4096 codes), Mode 3/A (4096 codes), and Mode C (encoded altitude), along with Ident (I/P) and Military Emergency. Mode 3/A went from 64 codes to 4096 codes.

d) *Mark XII System*. This system consists of the following capabilities: Mode 1 (32 codes), Mode 2 (4096 codes), Mode 3/A (4096 codes), Mode C (encoded altitude), and Mode 4, along with Ident (I/P), Military Emergency, and Unmanned Aircraft (X) indicators. Mode 4 is a military-only Positive Identification (PID) system. Certain models of the Mark XII system may also have a Mode S capability.

e) *Mark XIIA System*. This system consists of the following capabilities: Mode 1 (4096 codes), Mode 2 (4096 codes), Mode 3/A (4096 codes), Mode C (encoded altitude), Mode 4, Mode 5, and Mode S (selected levels), along with Ident (I/P), Military Emergency, and Unmanned aircraft (X) indicators. Mode 5 is a replacement PID system for the Mode 4 system. However, if an aircraft already has a Mode 4 capability, it is retained in addition to receiving the new Mode 5 capability and the Mode 4 data will be included in a Mode 5 reply.

f) *ATCRBS*. The Air Traffic Control Radar Beacon System (ATCRBS) is an all-encompassing term used to identify civilian transponder/interrogator systems used by commercial and general aviation. It consists of Mode A (4096 codes), Mode C (encoded altitude), Mode S, and the TCAS/ACAS systems. It also includes some Special Reply indicators. All military aircraft also have some

ATCRBS capabilities in addition to other systems that are military-only as they have to adhere to civilian air traffic control regulations when flying in such airspace.

g) *Mode S*. Mode S is a civilian ATCRB system that has a different interrogation-reply communications interface than Mode 3/A and C systems. It provides additional information on aircraft that is organized into Mode S levels. Only certain levels are mandated, and they are being phased in over a number of years in different geographic areas of the world. European airspace is now phasing in Mode S requirements, while the U.S. Federal Aviation Administration (FAA) authorities have no present plans to require Mode S in its airspace. U.S. and coalition military aircraft are also required to implement some Mode S levels if they will be flying in civilian airspace that requires Mode S.

h) *TCAS/ACAS I/II Systems*. There are two traffic collision and avoidance systems in wide use in civilian aviation. These are the TCAS and ACAS systems. These are essentially the same system with the TCAS system used in the United States and ACAS in Europe. Other countries of the world have adopted either one or the other system. A TCAS system has its own interrogator (transceiver) and interrogates Mode A/C and Mode S transponders. (TCAS does not know the Mode 3/A code and just uses the reply to determine range and bearing. However, if the transponder has a Mode C capability, a TCAS/ACAS II system can also obtain altitude.) A TCAS/ACAS I system provides a traffic advisory of a possible conflict. TCAS/ACAS II provides both a traffic advisory to a pilot as well as a resolution advisory directing the pilot of each aircraft involved in a potential collision to make a specific evasive maneuver.

i) *Soviet Systems*. The Soviets developed their own transponder and interrogator systems beginning with World War II. The SRO-2 transponder was a popular model installed on Soviet military aircraft. The SO-69M IFF transponder and other models also exist. Although an IFF PDU for a Soviet transponder or interrogator currently has minimal field and record requirements, the IFF PDU Layer 1, Layer 2, and Layer 5 field formats and IFF Data records provide the capability to describe any Soviet transponder or interrogator at any level of detail required. The United States and other countries have had a SIGINT capability to interrogate Soviet transponders since the 1970s, and that is why IFF PDUs for hostile aircraft are common in training exercises where the enemy includes countries that have bought Soviet fighters and other military aircraft for their air forces.

j) *RRB Systems*. The Reply Receiver "B" (RRB) transponder is in operational use by the British Royal Navy and other countries. The RRB transponder is interrogated by primary radar and gives an edge of band response. Hence, there are no interrogators. An RRB transponder is fitted to a number of aircraft such as the Lynx HAS 3 helicopter and the Sea Harrier. An RRB receiver aboard a platform such as a ship receives the RRB transmission and displays an indication on a radar screen. The response contains 1 of 16 reply codes, which are normally interpreted as a decimal instead of as an octal number. It also has a radar enhancement signal that works somewhat similar to an I/P reply.

B.1.3.3 Modes

This subclause provides a brief overview of common military and civilian modes and codes:

a) *Mode 1*. This is a military-only mode that consists of 32 octal codes with a unique format. It is normally used to define a mission (range 00 to 73). (See Mode 5 for a description of Enhanced Mode 1.)

b) *Mode 2*. This is a military-only mode that consists of 4096 codes. It is used to identify a specific aircraft or ship (range 0000 octal to 7777 octal).

c) *Mode 3/A*. Civilian Mode A is designated by the military as Mode 3 and consists of 4096 codes. (The military identifies modes by numbers, and the civilian organizations use letters.) The civilian air traffic control agencies use Mode 3/A codes to indicate the type of rule an aircraft is operating under, certain special conditions such as an emergency or hijacking, or it may be assigned as a unique identification number for an aircraft flying in controlled airspace. All military transponder and interrogator systems implement Mode 3/A (range 0000 octal to 7777 octal).

d) *Mode 4*. This is a military-only mode used to positively identify an aircraft as a friend.

e) *Mode C*. This is a civilian mode and indicates the altitude of an aircraft based on standard pressure altitude. All Mark XII and Mark XIIA military transponder and interrogator systems implement Mode C.

f) *Mode 5*. This is a military-only mode used to positively identify an aircraft as a friend. It replaces Mode 4. In addition to Mode 5 Positive Identification (PID) data, a Mark XIIA system also includes Mode 1, E1, 2, 3/A, C, Military S, as well as Ident, Emergency, and some other special responses. E1 refers to Enhanced Mode 1, which consists of 4096 octal codes instead of the normal Mode 1 set of 32 octal codes. Military S is a limited implementation of the civilian Mode S system that has been introduced primarily in Europe and is a requirement of certain classes of civilian and military aircraft.

g) *Mode S*. Mode S is a civilian mode, which like Mode A and C is required to also be implemented for military aircraft operating in worldwide airspace that requires this mode. Europe is a primary area that requires Mode S.

h) *Special Replies*. Some transponders have one or more Special Reply Indicators, which are described below:

   1) Military Emergency. This reply indicates a military emergency and is separate from the use of the Mode 3/A.

   2) Identification of Position (I/P). A pilot transmits the I/P special code by pressing a button in response to a request from an Air Traffic Control (ATC) or military controller to "Squawk Flash" or "Squawk Ident." It is transitory in nature and is used by controllers to locate an aircraft on their display. The I/P code is sent in response to just a few interrogations and then is automatically terminated.

   3) Unmanned Aircraft (X). This reply indicates that the aircraft is unmanned.

i) *Squitter*. Some transponders have a squitter capability. This allows them to transmit digital messages (reports) without being interrogated. Squitter reports are output at rates that may vary from several times a second to a second or more. Both military Mode 5 and civilian Mode S transponders have squitter capabilities as listed below:

   1) Civilian Mode S Squitter. Civilian Mode S squitter requirements are specified in ICAO publications (see Clause 2). There are two types of Mode S squitter: acquisition (short) and extended. Mode S short squitter reports contain aircraft ID data and are used for air-to-air surveillance such as between TCAS/ACAS-equipped aircraft. Mode S extended squitter reports contain aircraft ID, position data, and other parameters. Mode S extended squitter supports Automatic Dependent Surveillance—Broadcast (ADS-B).

   2) Military Mode 5 Squitter. The military Mode 5 squitter also sends reports without being interrogated. Mode 5 squitter reports only contain military Mode 5 message formats defined in AIMS 03-1000A. Also, they do not have the acquisition and extended squitter types. Some military systems (e.g., Mark XIIA) have both military Mode 5 squitter as well as Mode S squitter.

B.2 Identification Friend or Foe (IFF) PDU records

B.2.1 General

B.2.1.1 Record list

Table B.2 lists the IFF PDU records associated with one or more system types and system names. For Clause B.5, in which unique requirements for each system type are described, which of these records is required or is optional to be implemented for a given system is indicated. Non-self-identifying records that are applicable to some, but not all, systems are listed in this annex. If that type of record is applicable to any system type, it is listed in 6.2 in the body of the standard. The non-self-identifying records in 6.2 are listed here:

a) Beam Data record (see 6.2.11) *Layer 2*
b) Entity Coordinate Vector record (6.2.96) *Layer 1*
c) Entity Identification (ID) record (6.2.28) *Layer 1*
d) Event ID record (6.2.33) *Layer 1*
e) Fundamental Operational Data record (6.2.39) *Layer 1*
f) IFF Fundamental Parameter Data record (6.2.44) *Layer 2*
g) Layer Header record (6.2.51) *Layer 2 and higher*
h) PDU Header record (6.2.66) *Layer 1*
i) PDU Status record (6.2.67) *Layer 1*
j) Secondary Operational Data record (6.2.76) *Layer 2*
k) System ID record (6.2.87) *Layer 1*

Table B.2—IFF PDU records

Record name	Subcl.	Record name	Subcl.
Antenna Location IFF Data record	B.2.2	Mode 5 Transponder Basic Data record	B.2.29
Basic Interactive IFF Data record	B.2.3	Mode 5 Transponder Location IFF Data record	B.2.30
Change/Options record	B.2.4	Mode 5 Transponder SD record	B.2.31
Crypto Control IFF Data record	B.2.5	Mode 5 Transponder Status record	B.2.32
DAP Source record	B.2.6	Mode C Altitude record	B.2.33
Delta Barometric Altitude record	B.2.7	Mode C Interrogator Status record	B.2.34
Delta Mode 5 Altitude record	B.2.8	Mode S Aircraft Identification record	B.2.35
Enhanced Mode 1 Code record	B.2.9	Mode S Altitude record	B.2.36
Ground Initiated Comm-B (GICB) IFF Data record	B.2.10	Mode S Interrogator Basic Data record	B.2.37
Interactive Basic Mode 5 IFF Data record	B.2.11	Mode S Interrogator Identifier record	B.2.38
Interactive Basic Mode S IFF Data record	B.2.12	Mode S Interrogator Status record	B.2.39
Interactive Mode 4 Reply IFF Data record	B.2.13	Mode S Levels Present record	B.2.40
Interactive Mode 5 Reply IFF Data record	B.2.14	Mode S Transponder Basic Data record	B.2.41
Interrogated Modes record	B.2.15	Mode S Transponder Status record	B.2.42
ISLS record	B.2.16	Modifier record	B.2.43
Mode 1 Code record	B.2.17	RRB Code record	B.2.44
Mode 1 Interrogator Status record	B.2.18	Soviet Interrogator Status record	B.2.45
Mode 2 Code record	B.2.19	Soviet Transponder Status record	B.2.46
Mode 2 Interrogator Status record	B.2.20	Squitter Airborne Position Report IFF Data record	B.2.47

Table B.2—IFF PDU records *(continued)*

Record name	Subcl.
Mode 3/A Code record	B.2.21
Mode 3/A Interrogator Status record	B.2.22
Mode 4 Code record	B.2.23
Mode 4 Interrogator Status record	B.2.24
Mode 5 Altitude record	B.2.25
Mode 5 Interrogator Basic Data record	B.2.26
Mode 5 Interrogator Status record	B.2.27
Mode 5 Message Formats record	B.2.28

Record name	Subcl.
Squitter Airborne Velocity Report IFF Data record	B.2.48
Squitter Event-Driven Report IFF Data record	B.2.49
Squitter Identification Report IFF Data record	B.2.50
Squitter Surface Position Report IFF Data record	B.2.51
System Status record	B.2.52
TCAS/ACAS Status record	B.2.53
Transponder Location Error IFF Data record	B.2.54

B.2.1.2 Types of records

There are two types of records used in the IFF PDU: self-identifying records and non-self-identifying records:

a) *Self-identifying Record*. A self-identifying record has a unique numeric identifier that is contained in the Record Type field of the record. This type of record is referred to as an IFF Data record, and those words are always part of the record title (e.g., Crypto Control IFF Data record). IFF Data records shall only be contained in the IFF Data record section of an IFF layer (see 6.2.43). The IFF Data record description will indicate in which layer(s) it may be included. An IFF Data record may include one or more non-self-identifying records, but it shall not include another IFF Data record. IFF Data records may be included in Layer 5 to add additional fields or records to extend Layer 1 or Layer 2 information because those layers do not have an IFF Data record section.

b) *Non-self-identifying Record*. A non-self-identifying record is one that does not have a Record Type field to allow it to be uniquely identified. This type of record is found in the fixed field section of a layer of an IFF PDU and may also be present in an IFF Data record. A non-self-identifying record may contain other non-self-identifying records but shall not include any IFF Data records.

B.2.1.3 Special fields

B.2.1.3.1 General

Layers 1 and 2 contain special fields called System-Specific Data fields and Data Field 1 and Data Field 2. These fields may contain a single field value or a non-self-identifying record whose content is based on the system type. These fields shall only be used for data that is applicable to one or more system types, but not to all system types. If it is necessary to add an additional field or record to Layer 1 or 2 that is applicable to all system types, present or future, then an IFF Data record shall be used for that purpose and included in Layer 5. The use of any of these special fields by a system type are defined in the Information Content section of B.5 for that system type.

Special fields also include any fields of IFF PDU records that have special relationships to other fields in an IFF PDU.

NOTE—When a new system type is implemented, and it is desired to use a special field for data associated with that system type, the present uses of the special field needs to be reviewed to determine whether one of the uses is applicable to the new system type. If this is the case, that special field cannot be used for new data, and another special field, an IFF Data record, or the use of one of the Parameter fields (1 to 6) of the Fundamental Operational data record can be used instead.

B.2.1.3.2 Special field specifications

The specifications for special fields are listed here:

a) *Alternate Mode 4 Challenge/Reply field*. This is an 8-bit enumeration field that is contained in Data Field 1 of the Fundamental Operational Data record (6.2.39) for those system types that are required to implement this field as specified in B.5. This field is implemented for transponders that have a Mode 4 capability. It shall indicate the reply that would be received from a Mode 4 transponder by a Mode 4 interrogator. It consists of the following enumerations: No Statement (0), Valid (1), Invalid (2) No Response (3), and Unable to Verify (4). (See B.5.1.2.2 for the special issuance and receipt rules associated with this field.) This field shall be set to No Statement (0) when it does not apply to the system type.

b) *System Mode*. This is an 8-bit enumeration field that is contained in the System ID record (see 6.2.87). It provides additional values associated with the master switch for a transponder. The enumerations for this field and their relationship to the System On/Off Status field of the System Status record shall be as follows:

    1) No Statement (0). This status shall indicate that the transponder does not implement a nonzero value for this field. There is no additional system mode information. When set to this mode, the System On/Off Status field may be set to either Off (0) or On (1).

    2) Off (1). This status shall indicate that the transponder is powered off. In this case, it is not emitting any signal or able to reply to an interrogation. When set to this mode, the System On/Off Status field shall be set to Off (0).

    3) Standby (2). This status shall indicate that the transponder is powered on but is not emitting any signal or able to reply to an interrogation. When set to this mode, the System On/Off Status field shall be set to Off (0).

    4) Normal/On (3). When set to either Normal or On, it shall indicate that the transponder is powered on and is operating at full sensitivity as defined by applicable specifications for this system type and system name. It would emit a signal either in response to an interrogation or if the squitter function is on. When set to this mode, the System On/Off Status field shall be set to On (1).

    5) Emergency (4). When set to Emergency, it shall indicate that the transponder is powered on and is operating at full sensitivity as defined by applicable specifications for this system type and system name. It would emit a signal in response to an interrogation or when the squitter function is on. When set to this mode, the System On/Off Status field shall be set to On (1).

    6) Low or Low Sensitivity (5). When set to either Low or Low Sensitivity, it shall indicate that the transponder is powered on, but the transponder is operating at reduced sensitivity to interrogations. The sensitivity to all interrogations is reduced as specified in the applicable specifications for this system type and system name. This means that a transponder will be less likely to respond to an interrogation from an interrogator that is further away. However, if this is a military transponder with a Mode 4 capability, it still has normal sensitivity for Mode 4 interrogations. When set to this mode, the System On/Off Status field shall be set to On (1).

    7) A simulation may have the capability to set the System On/Off Status field of the System Status record directly instead of having it set based on the present System Mode field value as described herein. In that case, the System Mode field value will be automatically changed to be comparable as specified in item l1) in 5.7.6.3.

B.2.1.3.3 Layer 1 special fields

The special fields associated with Layer 1 are as follows:

a) System-Specific Data field. No system type currently implements this field.
b) Fundamental Operational Data record
   1) Data Field 1
      i) Alternate Mode 4 Challenge/Reply field (B.2.1.3.2)
   2) Data Field 2
      i) Modifier record (B.2.43)
c) Change/Options record
   1) System-Specific Field 1
      i) Alternate Mode 4 Indicator [item b) in B.2.4]
   2) System-Specific Field 2
      i) Alternate Mode C Indicator [item c) in B.2.4]
d) System ID record (6.2.87)
   1) System Mode (B.2.1.3.2)

B.2.1.3.4 Layer 2 special fields

The special field associated with Layer 2 are as follows:

a) System-Specific Data
   1) ISLS record (B.2.16)
2) Secondary Operational Data
   1) Operational Parameter 1 (no values have been assigned to this data field)
   2) Operational Parameter 2 (no values have been assigned to this data field)

B.2.2 Antenna Location IFF Data record

The specification of the location of each auxiliary IFF antenna for a transponder or interrogator associated with a specific IFF System ID for an entity shall be communicated by the Antenna Location IFF Data record. An auxiliary IFF antenna is one that is in addition to the primary antenna whose location is contained in Layer 1 of the IFF PDU. This is an optional record for issuance and receipt. However, if this record is implemented, it shall adhere to the requirements specified herein. This record shall be contained in the appropriate IFF Data record section of a layer of the IFF PDU based on the system type as follows: Layer 3 for a Mode 5 or Mode 5/S system, Layer 4 for a Mode S only system, and Layer 5 for all other system types. The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5009.
b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)
c) *System Name*. This field shall specify the name of the system with which this auxiliary IFF antenna is associated. It shall be represented by the same system name as contained in the System ID record (see [UID 83]).
d) *Antenna Reference Number*. This field shall specify the reference number associated with each auxiliary antenna. The first auxiliary antenna shall be given the number 1 and then incremented by one for each additional antenna associated with the same System ID for an entity.

e) *Antenna Status*. This field shall specify the status of the auxiliary antenna. It shall be represented as an 8-bit enumeration (see [UID 371]) with the following values: No Statement (0), Not Able to Emit (1), and Able to Emit (2). The Able to Emit (2) status indicates that this auxiliary IFF antenna is turned on and would transmit emissions if interrogated or otherwise would send an emission.

f) *Relative Antenna Location*. This field shall specify the location of the auxiliary antenna relative to the associated emitting entity's coordinate system. This field shall be represented by an Entity Coordinate Vector record [see item a) in 6.2.98].

The format of the Antenna Location IFF Data record shall be as shown in Table B.3.

Table B.3—Antenna Location IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5009	32-bit enumeration
16	Record Length = 24	16-bit unsigned integer
16	System Name	16-bit enumeration
8	Antenna Reference Number	8-bit unsigned integer
8	Antenna Status	8-bit enumeration
16	Padding	16 bits unused
96	Relative Antenna Location	Entity Coordinate Vector record
Total Antenna Location IFF Data record size = 192 bits		

B.2.3 Basic Interactive IFF Data record

The specification of basic information for an interactive interrogation, reply, or squitter transmission between two specific entities having an IFF system contained in an *interactive* IFF PDU shall be communicated by the Basic Interactive IFF Data record. This record shall be the first IFF Data record in the IFF Data record section of Layer 5 of an *interactive* IFF PDU. The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5010.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer (see 6.2.43).

c) *Interrogating Entity ID*. This field shall specify the Entity ID of the entity that is interrogating another entity. It shall be represented by an Entity ID record (see 6.2.28).

d) *Interrogated Entity ID*. This field shall specify the Entity ID of the entity to which the interrogation is being directed. It shall be represented by an Entity ID record (see 6.2.28).

e) *Interactive Event ID*. This field shall contain the Interactive Event ID. It shall be represented by an Event ID record (see 6.2.33).

f) *Interrogated Modes*. This field shall indicate the modes being interrogated. It shall be represented by an Interrogated Modes record (see B.2.15).

g) *Transmission Indicator*. This field shall specify whether this is an interrogation, a reply to an interrogation, or a squitter transmission. It shall be represented by an 8-bit enumeration (see [UID 372]).

h) *Reply Amplification*. This field shall provide amplifying information about the reply. It shall be represented by an 8-bit enumeration (see [UID 373]).

The format of the Basic Interactive IFF Data record shall be as shown in Table B.4.

Table B.4—Basic Interactive IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5010	32-bit enumeration
16	Record Length = 32	16-bit unsigned integer
48	Interrogating Entity ID	Entity Identifier record
48	Interrogated Entity ID	Entity Identifier record
48	Interactive Event ID	Event Identifier record
16	Interrogated Modes	Interrogated Modes record
8	Transmission Indicator	8-bit enumeration
8	Reply Amplification	8-bit enumeration
32	Padding	32 bits unused
Total Basic Interactive IFF Data record size = 256 bits		

B.2.4 Change/Options record

The Change/Options record shall be used to communicate the reasons for the issuance of an IFF PDU and to provide the status of various options that may be associated with an interrogator or transponder. Some of the fields are common to all system types and regardless of whether it is an interrogator or transponder. Other fields are only applicable to specific system types or depend on whether it is a transponder or an interrogator.

The fields of this record, except for fields required to be implemented by all system types, depend on the system type or subsystem type (e.g., for System Type 1, the subsystem types are Mark X, Mark XII, and ATCRBS as a given transponder designated as a System Type 1 transponder may be one of the three types). References to system type in this subclause shall include subsystem types unless otherwise indicated.

This is an 8-bit record as shown in Table B.5. There are five bit positions that have a single, permanently assigned field type for each position. There are two bit positions designated as System-Specific Fields 1 and 2. These two System-Specific fields are permanently assigned to the indicated bit positions. Each System-Specific field may have one or more field types assigned to it that are applicable to one or more system types. A given field type may only be assigned to one System-Specific field. Table B.5 indicates the field types associated with each System-Specific field and which system types, if any, that are required to implement them.

The fields of this record shall be as follows:

a) *Change Indicator*. This 1-bit enumeration field shall indicate whether this is the initial issuance of an IFF PDU for an entity or whether data has changed from a previously issued IFF PDU. All system types shall implement this field:

   1) Not Initial Report/No Change (0). The Change Indicator shall be set to this value if both of the following conditions are met:

      i) This is not the initial issuance of an IFF PDU.

      ii) There has been no change in data since the last IFF PDU was issued.

   2) Initial Report/Change (1). The Change Indicator shall be set to this value if any of the following conditions are met:

      i) This is the initial issuance of an IFF PDU for an entity.

      ii) There has been a change in data since the last IFF PDU was issued.

      iii) This is the first *heartbeat* IFF PDU following a non-*heartbeat* IFF PDU where the non-*heartbeat* IFF PDU had the Change Indicator set to Initial Report/Change (1).

b) *System-Specific Field 1*. This System-Specific field may have one or more field types assigned to it. If a system type is not required to implement this field, the field shall be set to zero. The field type(s) assigned to System-Specific Field 1 are as follows:

   1) Alternate Mode 4 Indicator. This field shall indicate whether a system type has a Mode 4 capability. This field shall be set to Not Present (0) or Present (1). If set to Present (1), the Alternate Mode 4 Challenge/Reply field will be to an appropriate Mode 4 reply value. This field shall be implemented by any system type that has a Mode 4 capability.

   2) Additional Fields. A system type that does not have a Mode 4 capability may use this field for other data such as a state or status value unique to that system type. In this case, the name of the system-specific field and its purpose will be added to this paragraph along with the system type(s) to which it applies.

c) *System-Specific Field 2*. This is a System-Specific field that may have one or more field types assigned to it. The field types assigned to System-Specific Field 2 are as follows:

   1) Alternate Mode C Indicator. This field indicates whether this system type has a Mode C capability. This field shall be set to Not Present (0) or Present (1). This field shall be implemented by any system type that has a Mode C capability.

   2) Additional Fields. A system type that does not have a Mode 4 capability may use this field for other data such as a state or status value unique to that system type. In this case, the name of the system-specific field and its purpose will be added to this paragraph along with the system type(s) to which it applies.

d) *Heartbeat Indicator*. This 1-bit enumeration field shall indicate whether this IFF PDU is a Heartbeat (1) or Not a Heartbeat (0). This field shall be optional to implement for System Types 1 through 5 and is required to be implemented for all other system types.

e) *Transponder/Interrogator Indicator*. This 1-bit enumeration field (see [UID 337]) shall indicate whether the IFF system is a Transponder (0) or an Interrogator (1). This field shall be optional to implement for System Types 1 through 5 and is required to be implemented for all other system types.

   NOTE—A system type can be a combined transponder and interrogator so the system type and system name is not sufficient to identify whether it is a transponder or an interrogator.

f) *Simulation Mode*. This 1-bit enumeration field (see [UID 338]) shall indicate whether this IFF system is in the Regeneration (0) or Interactive (1) Simulation Mode. This field shall be implemented by all system types.

g) *Interactive Capable*. This 1-bit enumeration field shall indicate whether or not the IFF system type for this entity is currently operating in the Interactive Mode as follows: Capable (1) or Not Capable (0).

h) *Test Mode*. This 1-bit enumeration field shall indicate whether the IFF system has been placed in the equipment test mode. The test mode shall be set to either Off (0) or On (1). (See B.5.1.2.7.)

The format of the Change/Options record shall be as shown in Table B.5.

Table B.5—Change/Options record

Field name	Bits	Value
Change Indicator	0	Enumeration
System-Specific Field 1	1	Enumeration
System-Specific Field 2	2	Enumeration
Heartbeat Indicator	3	Enumeration
Transponder/Interrogator Indicator	4	Enumeration
Simulation Mode	5	Enumeration
Interactive Capable	6	Enumeration
Test Mode	7	Enumeration
Total Change/Options record size = 8 bits		

The specific requirements to implement a given Change/Options record field are based on the system type as shown in Table B.6.

Table B.6—Change/Options implementation by system type

System type/title	Change/Options field names						
	Chg Ind.	System Specific Field 1*	System Specific Field 2*	Heartbeat Indicator	T/I Ind.	Sim Mode	Test Mode
1 Mark X/XII/ATCRBS Transponder							
If Mark X subtype	R	NF	NF	O	O	R	O
If Mark XII subtype	R	R1	R2	O	O	R	O
If ATCRBS subtype	R	NF	R3	O	O	R	O
2 Mark X/XII/ATCRBS Interrogator							
If Mark X subtype	R	NF	NF	O	O	R	O
If Mark XII subtype	R	NF	NF	O	O	R	O
If ATCRBS subtype	R	NF	NF	O	O	R	O
3 Soviet Transponder	R	NF	NF	O	O	R	O
4 Soviet Interrogator	R	NF	NF	O	O	R	O

Table B.6—Change/Options implementation by system type *(continued)*

System type/title	Change/Options field names						
	Chg Ind.	**System Specific Field 1***	**System Specific Field 2***	**Heartbeat Indicator**	**T/I Ind.**	**Sim Mode**	**Test Mode**
5 RRB Transponder	R	NF	NF	O	O	R	O
6 Mark XIIA (5/S) Interrogator	R	NF	NF	R	R	R	O
7 Mode 5 Interrogator	R	NF	NF	R	R	R	O
8 Mode S Interrogator	R	NF	NF	R	R	R	O
9 Mark XIIA (5/S) Transponder	R	R1	R2	R	R	R	O
10 Mode 5 Transponder	R	R1	R2	R	R	R	O
11 Mode S Transponder	R	R1	R2	R	R	R	O
12 Combined Mark XIIA Interrogator/Transponder							
If Interrogator IFF PDU	R	NF	NF	R	R	R	O
If Transponder IFF PDU	R	R1	R2	R	R	R	O
13 Combined Mark XII Interrogator/Transponder							
If Interrogator IFF PDU	R	NF	NF	R	R	R	O
If Transponder IFF PDU	R	R1	R2	R	R	R	O
14 TCAS/ACAS Transceiver	R	NF	NF	R	NF	R	O
Additional IFF System Types	R	N1	N1	R	N1	R	N2

NOTE—System-Specific fields are specific to a given system type. Each field can only be used for one purpose.

Legend

Ind. = Indicator
N1 = Additional IFF system types may use this field if required to implement a new field definition. A System-Specific field can only contain one field definition per system type. If not used, the field shall be set to zero.
N2 = Future IFF systems will determine whether the Test Mode is to be optional or required for a given system type.
NF = No fields defined. No System-Specific fields are defined for this system type or subtype.

R = Required
R1 = Required. Contains the Alternate Mode 4 Indicator (see B.5.1.2.2)
R2 = Required. Contains the Alternate Mode C Indicator (see B.5.1.2.3)
R3 = Required if this ATCRBS transponder has a Mode C capability (see B.5.1.2.3)

O = Optional

B.2.5 Crypto Control IFF Data record

The Crypto Control IFF Data record shall be used to communicate Mode 5 interrogator and transponder pseudo crypto control information. This is to allow the simulation of encryption related to Mode 5 IFF interrogators and transponders. This is not related to the encryption of real-world communications between simulations and other participants in an exercise.

This is an optional IFF Data record for a Mode 5 interrogator or transponder that (when used) is included in the IFF Data record section of Layer 3. However, if this record is implemented, it shall adhere to the requirements specified here and elsewhere in this standard. See B.5.1.2.4 for additional requirements related to associating Mode 4 and 5 pseudo crypto codes with related fields in other records.

Some Mode 5 interrogators and transponders include the legacy Mode 4 capability. This record includes Mode 4 pseudo crypto Code A and B information that may be included if desired. If Mode 4 pseudo crypto information is included, at a minimum, the Mode 4 Pseudo Crypto Code A field shall be set accordingly and the associated Active Indicator shall be set to Active (1). If both the Mode 4 Pseudo Crypto Code A and Code B are included, then one of the associated Active Indicator fields shall be set to Active (1) and the other to Not Active (0); i.e., exactly one shall be active at a time. If this Mode 5 interrogator or transponder does not have the legacy Mode 4 capability, then the Mode 4 Pseudo Crypto Code A and B fields shall be set to zero and the associated Active Indicator fields shall be set to Not Active (0).

NOTE 1—The Mode 5 pseudo crypto information contained in this record needs to be correlated with the Mode 5 Reply field of the Mode 5 Transponder Status record as specified in B.2.27. If Mode 4 Pseudo Crypto Code A or Code B is included, the currently active Mode 4 Pseudo Crypto Code (A or B) needs to match the Layer 1 Mode 4 Code.

When the Crypto Control IFF Data record is included in an IFF PDU, the Crypto Control field of the Mode 5 Transponder Status record shall be set to Crypto Control IFF Data Record Present (1).

NOTE 2—Care needs to be taken when discussing details of Mode 5 as certain information related to this capability are classified.

The fields and subfields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. The record type value shall be set to 5000.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer (see 6.2.43).

c) *Mode 5 Pseudo Crypto Code*. This field shall contain the pseudo crypto code for a Mode 5 transponder or interrogator. This field is a 16-bit record that uses 12 bits for the pseudo crypto code with 4 bits of padding. The pseudo crypto code shall contain either a 12-bit pseudo crypto code with a value 0 to 4094 (decimal) or an indication to use the Mode 5 Reply value in the Mode 5 Transponder Status record when set to 4095 (decimal).

d) *Mode 4 Pseudo Crypto Code A*. This field shall contain the Mode 4 pseudo crypto value for Code A and the associated Active Indicator. This field is a 16-bit record. The subfields shall be as follows:

   1) Pseudo Crypto Code. This subfield shall contain either a 12-bit pseudo crypto code with a value 0 to 4094 (decimal) or an indication to use the Mode 4 Reply value in the Layer 1 Mode 4 Code record when set to 4095 (decimal).

   2) Active Indicator. This 1-bit enumeration subfield shall indicate whether this pseudo crypto code is Active (1) or Not Active (0).

e) *Mode 4 Pseudo Crypto Code B*. This field shall contain the Mode 4 pseudo crypto value for Code B and the associated Active Indicator. This field is a 16-bit record and uses the same subfields as shown in item d) above.

The format of the Crypto Control IFF Data record shall be as shown in Table B.7.

Table B.7—Crypto Control IFF Data record

Field size (bits)	Field name	Value
32	Record Type = 5000	32-bit enumeration
16	Record Length = 12	16-bit unsigned integer
16	Mode 5 Pseudo Crypto Code	Pseudo Crypto Code (0 to 11)
		Padding (12 to 15)
16	Mode 4 Pseudo Crypto Code A	Pseudo Crypto Code (0 to 11)
		Active Indicator (12)
		Padding (13 to 15)
16	Mode 4 Pseudo Crypto Code B	Pseudo Crypto Code (0 to 11)
		Active Indicator (12)
		Padding (13 to 15)
Total Crypto Control IFF Data record size = 96 bits		

B.2.6 DAP Source record

The source of Downlink of Aircraft Parameters (DAP) shall be specified by the DAP Source record. This record is contained in the DAP Source field of the Mode S Transponder Basic Data record. See B.5.1.2.6 for special issuance and receipt rules related to DAP data.

DAP information is transmitted by a Mode S transponder system when requested and if it has implemented the appropriate Mode S Level. When a simulation is operating in the Regeneration Mode and sends an IFF PDU for a Mode S transponder, all field values of this record shall be set to Compute Locally (0). It is up to a receiving simulation that requires DAP data to compute the DAP values based on truth data and other local data along with any necessary algorithms. If the sending simulation is operating in the Interactive Mode, it may set one or more DAP values to IFF Data Record Available (1) if it has included an IFF Data record for that DAP value. Access to appropriate ICAO documentation and specific transponder model specifications are required to properly set and process DAP values (see Clause 2).

NOTE—IFF Data records have not been defined for specific DAP data. When such records are developed, it is recommended that the DAP field format from the appropriate ICAO publication be used.

The fields of the DAP Source record shall be as follows:

a) *Indicated Air Speed (IAS)*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).
b) *Mach Number*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).
c) *Ground Speed*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).
d) *Magnetic Heading*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).

e) *Track Angle Rate*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).

f) *True Track Angle*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).

g) *True Airspeed*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).

h) *Vertical Rate*. This 1-bit enumeration field shall be set to Compute Locally (0) or IFF Data Record Available (1).

The format of this record shall be as shown in Table B.8.

Table B.8—DAP Source record

Bit	Field name	Value
0	Indicated Air Speed (IAS)	Enumeration
1	Mach Number	Enumeration
2	Ground Speed	Enumeration
3	Magnetic Heading	Enumeration
4	Track Angle Rate	Enumeration
5	True Track Angle	Enumeration
6	True Airspeed	Enumeration
7	Vertical Rate	Enumeration
Total DAP Source record size = 8 bits		

B.2.7 Delta Barometric Altitude record

The Delta Barometric Altitude record shall be used to specify delta barometric altitude values associated with a Transponder Location Error IFF Data record. If there is no error in barometric altitude, this record shall be set to zero. This is a 16-bit record.

The fields of this record shall be as follows:

a) *Delta Barometric Altitude*. This field shall indicate the delta barometric altitude in the resolution specified by the Altitude Resolution field. For example, if the delta barometric altitude is 1000 feet and the Altitude Resolution field is set to 100-foot increments, the Delta Barometric Altitude field would be set to 10. This 11-bit field shall be represented by an 11-bit unsigned integer.

b) *Positive/Negative Indicator*. This 1-bit enumeration field (see [UID 362]) shall indicate whether the delta altitude value is Positive (0) or Negative (1). The Positive indicator means the offset is above the entity's present truth altitude, and a Negative indicator means the offset is below the entity's present truth altitude.

c) *Altitude Resolution*. This 1-bit enumeration field (see [UID 361]) shall indicate the resolution of the altitude as either 100-foot (0) or 25-foot (1) resolution.

The format of the Delta Barometric Altitude record shall be as shown in Table B.9.

Table B.9—Delta Barometric Altitude record

Field name	Bits	Value
Delta Barometric Altitude	0 to 10	11-bit unsigned integer
Positive/Negative Indicator	11	Enumeration
Altitude Resolution	12	Enumeration
Padding	13 to 15	3 bits unused
Total Delta Barometric Altitude record size = 16 bits		

B.2.8 Delta Mode 5 Altitude record

The Delta Mode 5 Altitude record shall be used to specify delta Mode 5 altitude values associated with a Transponder Location Error IFF Data record. If there is no error in Mode 5 altitude, this record shall be set to zero. This is a 16-bit record.

The fields of this record shall be as follows:

a) *Delta Mode 5 Altitude*. This 11-bit field shall indicate the delta Mode 5 altitude in the resolution specified by the Altitude Resolution field. For example, if the delta altitude is 1000 feet and the Altitude Resolution field is set to 100-foot increments, the Delta Mode 5 altitude field would be set to 10. This field shall be represented by an 11-bit unsigned integer.

b) *Positive/Negative Indicator*. This 1-bit enumeration field (see [UID 362]) shall indicate whether the delta altitude value is Positive (0) or Negative (1). The Positive indicator means the offset is above the entity's present truth altitude and a Negative indicator means the offset is below the entity's present truth altitude.

c) *Altitude Resolution*. This 1-bit enumeration field (see [UID 361]) shall indicate the resolution of the altitude as either 100-foot (0) or 25-foot (1) resolution.

The format of the Delta Mode 5 Altitude record shall be as shown in Table B.10.

Table B.10—Delta Mode 5 Altitude record

Field name	Bits	Value
Delta Mode 5 Altitude	0 to 10	11-bit unsigned integer
Positive/Negative Indicator	11	Enumeration
Altitude Resolution	12	Enumeration
Padding	13 to 15	3 bits unused
Total Delta Mode 5 Altitude record size = 16 bits		

B.2.9 Enhanced Mode 1 Code record

The Enhanced Mode 1 Code record shall provide Mode 5 Enhanced Mode 1 information. When used, it shall be included in the Layer 3 Mode 5 Transponder Basic Data record (B.2.29). This is a 16-bit record. The range is 0000 octal to 7777 octal. For example, given an Enhanced Mode 1 code of 3457 octal, the first digit (3) becomes Code Element 4 (A) and occupies bits 9 to 11, digit (4) is Code Element 3 (B), digit (5) is Code Element 2 (C), and digit (7) is Code Element 1 (D). They occupy the bit positions as shown in Table B.11.

In actual operations, a Mode 5 transponder will reply to an interrogation request for its Mode 1 code received from a non-Mode 5 interrogator that is capable of interrogating the 2-digit, Mode 1 code. However, a Mode 5 transponder will reply with the Enhanced Mode 1, 4-digit code value instead of a normal Mode 1, 2-digit code value. The non-Mode 5 interrogator will process only two of the four digits. The normal Mode 1, 2-digit code has a special numeric range. In order that a non-Mode 5 interrogator model processes an Enhanced Mode 1 Code record (B.2.9) as they would in actual operations, a Mode 5 transponder shall also include and set the Layer 1, Mode 1 Code record (B.2.17) to appropriate values. As in real operations, this translation may result in certain numbers being interpreted incorrectly at the receiving interrogator. The required translation so that a non-Mode 5 interrogator receives the same incorrect values that they would in actual operations is specified in Table B.11.

Table B.11—Enhanced Mode 1 translation

Enhanced Mode 1 Code record		Mode 1 Code record	
Field name	**Value (octal)**	**Field name**	**Value (octal)**
Code Element 1 (D)	0 to 7	Not applicable	—
Code Element 2 (C)	0 to 7	Not applicable	—
Code Element 3 (B)	0 to 7	Code Element 1 (B) [No change in values 0 to 3]	Enhanced / Mode 1 0 to 3 / 0 to 3 4 becomes 0 5 becomes 1 6 becomes 2 7 becomes 3
Code Element 4 (A)	0 to 7	Code Element 2 (A) [No change in values 0 to 7]	Enhanced / Mode 1 0 to 7 / 0 to 7

NOTE—It is expected that, until all aircraft with a Mode 1 capability transition to the Mode 5 System and its Enhanced Mode 1 capability, Mode 5 transponders will set their 4-digit Enhanced Mode 1 code to match the Mark XII 2-digit Mode 1 Code limitation. This will be accomplished by placing zeros in Enhanced Mode 1 Code Elements 1 and 2 and ensuring that Elements 3 and 4 contain values in the special 2-digit, Mode 1 Code range. However, in the event that this procedure is not followed, the requirements specified herein will result in the resulting Enhanced Mode 1 Code data being processed incorrectly by a non-Mode 5 capable interrogator just as would occur in the real world.

The fields of this record are as follows:

a) *Code Element 1*. This field shall identify Element 1 of the Enhanced Mode 1 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element D and is the fourth digit reading left to right in an operational display of this mode.)

b) *Code Element 2*. This field shall identify Element 2 of the Enhanced Mode 1 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element C and is the third digit reading left to right in an operational display of this mode.)

c) *Code Element 3*. This field shall identify Element 3 of the Enhanced Mode 1 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element B and is the second digit reading left to right in an operational display of this mode.)

d) *Code Element 4*. This field shall identify Element 4 of the Enhanced Mode 1 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element A and is the first digit reading left to right in an operational display of this mode.)

e) *On/Off Status*. This 1-bit enumeration field shall indicate whether Enhanced Mode 1 is On (1) or Off (0). The other fields of this record are independent of the On/Off Status field and shall remain unaffected when the status changes.

f) *Damage Status*. This 1-bit enumeration field shall indicate the damage status of the Enhanced Mode 1 equipment and be set to either No Damage (0) or Damaged (1).

g) *Malfunction Status*. This 1-bit enumeration field shall indicate the malfunction status of the Enhanced Mode 1 equipment and be set to either No Malfunction (0) or Malfunction (1).

The format of the Enhanced Mode 1 Code record shall be as shown in Table B.12. The letters in parentheses are the AIMS letter designation for each element.

Table B.12—Enhanced Mode 1 Code record

Field name	Bits	Value (octal)
Code Element 1 (D)	0 to 2	0 to 7
Code Element 2 (C)	3 to 5	0 to 7
Code Element 3 (B)	6 to 8	0 to 7
Code Element 4 (A)	9 to 11	0 to 7
Padding	12	0
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Enhanced Mode 1 record size = 16 bits		

B.2.10 GICB IFF Data record

The Ground-Initiated Comm-B (GICB) IFF Data record shall be used to communicate a Transponder/Interrogator (T/I) data link message that uses the GICB format. Where reference is made to using data values from an appropriate International Civil Aviation Organization (ICAO) technical publication for a specific data field in this record, the specific data and associated numerical assignments from the ICAO publication shall be used (see Clause 2). When required, this record shall be included in Layer 5 of an IFF PDU.

The GICB IFF Data record shall contain the following fields:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5007.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *T/I Link Type*. This field shall contain the T/I link type. This field shall be represented by a 16-bit enumeration (see [UID 370]).

d) *GICB Message Designator*. This field shall contain the GICB message designator as defined in appropriate ICAO technical publications (see Clause 2). This field shall be represented by a 16-bit enumeration.

e) *GICB Register/Message Number*. This field shall contain the applicable register and message number for this message. It shall be represented by an 8-bit unsigned integer.

f) *GICB Status*. This field shall contain two subfields—Process Status and Register Data Validity. Each field shall be represented by an 8-bit enumeration.

g) *GICB Register*. This field shall contain the GICB register data. It shall be represented by a 32-bit enumeration.

h) *GICB Message*. This field shall contain the GICB message data. It shall consist of two subfields—MSB and LSB. Each subfield shall be represented by a 32-bit unsigned integer.

The format of the GICB IFF Data record shall be as shown in Table B.13.

Table B.13—GICB IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5007	32-bit enumeration
16	Record Length = 28	16-bit unsigned integer
16	Padding	16 bits unused
16	T/I Link Type	16-bit enumeration
16	GICB Message Designator	16-bit enumeration
8	GICB Register/Message Number	8-bit unsigned integer
8	Padding	8 bits unused
16	GICB Status	Process Status—8-bit enumeration
		Register Data Validity—8-bit enumeration
32	GICB Register	32-bit enumeration
64	GICB Message	MSB—32-bit unsigned integer
		LSB—32-bit unsigned integer
Total GICB IFF Data record size = 224 bits		

B.2.11 Interactive Basic Mode 5 IFF Data record

The specification of basic Interactive Mode 5 data for a Mode 5 interrogation contained in an *interactive* IFF PDU and the Mode 5 transponder data in the *interactive* IFF PDU reply to an Interactive Mode 5 interrogation shall be communicated by the Interactive Basic Mode 5 IFF Data record. The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5013.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Mode 5 Message Formats*. This field shall specify the Mode 5 Message Formats as follows:

   1) For a Mode 5 interrogator, this field contains the specific Message Formats being requested. These message formats may be different from those contained in the *regeneration* IFF PDU for this interrogator.

   2) For a Mode 5 transponder, this field contains the specific Message Formats that it actually responded to for this interrogation. This may be different from the Message Formats contained in the interrogation request, as well as the present Message Formats contained in the *regeneration* IFF PDU for this transponder.

   It shall be represented by a Mode 5 Message Formats record (see B.2.28).

The format of the Interactive Basic Mode 5 IFF Data record shall be as shown in Table B.14.

Table B.14—Interactive Basic Mode 5 IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5013	32-bit enumeration
16	Record Length = 12	16-bit unsigned integer
16	Padding	16 bits unused
32	Mode 5 Message Formats	Mode 5 Message Formats record
Total Interactive Basic Mode 5 IFF Data record size = 96 bits		

B.2.12 Interactive Basic Mode S IFF Data record

The specification of basic Interactive Mode S data that needs to be included in an *interactive* IFF PDU from a Mode S interrogator or transponder shall be communicated by the Interactive Basic Mode S IFF Data record. The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5014.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Mode S Levels*. This field shall specify the Mode S Levels that are contained in an Interactive Mode interrogation or in a transponder reply to the interrogation. It shall be represented by a Mode S Levels Present record (see B.2.40). The value of this field may or may not be the same as the present value for this record as contained in the Layer 4, Mode S Levels Present field for a Mode S interrogator or transponder:

   1) Mode S Interrogator. This field indicates the Mode S levels it is interrogating.

   2) Mode S Transponder. This field indicates the actual levels contained in this reply.

d) *Mode S Transmit State*. This field shall specify the Mode S Transmit State that would be contained in this specific Mode S interactive interrogation. This field shall be represented by an 8-bit enumeration (see [UID 347]). It shall use the same enumerations as specified for the Transmit State field of the Mode S Interrogator Status record (see B.2.39). The value of this field may or may not be the same as the present value for this record as contained in Layer 4 for a Mode S interrogator:

1) Mode S Interrogator. Indicates the Transmit State of the interactive interrogation.
2) Mode S Transponder. Not applicable and set to zero.

e) *Mode S Interrogator Identifier*. This field shall specify the Mode S Interrogator Identifier that would be contained in this specific Mode S interactive interrogation. It shall use the Mode S Interrogator Identifier record (see B.2.38). The values of this record may or may not be the same as the present values for this record as contained in Parameter 4 of the Fundamental Operational Data record of Layer 4 for a Mode S interrogator. The Mode S Interrogator Identifier record as used for this field shall contain, as a minimum, a valid Primary IC Code (IC Type and IC Code fields contain valid data). The Secondary IC Code field is optional:

1) Mode S Interrogator. Indicates the interrogator identifier for this Mode S interrogator.
2) Mode S Transponder. Not applicable and set to zero.

The format of the Interactive Basic Mode S IFF Data record shall be as shown in Table B.15.

Table B.15—Interactive Basic Mode S IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5014	32-bit enumeration
16	Record Length = 12	16-bit unsigned integer
8	Mode S Levels	Mode S Levels Presents records
8	Mode S Transmit State	8-bit enumeration
16	Mode S Interrogator Identifier	Mode S Interrogator Identifier record
16	Padding	16 bits unused
Total Interactive Basic Mode S IFF Data record size = 96 bits		

B.2.13 Interactive Mode 4 Reply IFF Data record

The specification of an interactive Mode 4 reply to an Interactive Mode 4 interrogation shall be communicated by the Interactive Mode 4 Reply IFF Data record. It is included in Layer 5 for an interactive reply. The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5011.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Interactive Mode 4 Reply*. This field shall specify the Mode 4 reply that would be sent by this transponder to this Interactive Mode 4 interrogation. It shall use the same enumerations as specified for the Alternate Mode 4 Challenge/Reply field. This may not be the same as the regeneration value. It shall be represented by an 8-bit enumeration (see [UID 96]).

The format of the Interactive Mode 4 Reply IFF Data record shall be as shown in Table B.16.

Table B.16—Interactive Mode 4 Reply IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5011	32-bit enumeration
16	Record Length = 12	16-bit unsigned integer
8	Interactive Mode 4 Reply	8-bit enumeration
8	Padding	8 bits unused
32	Padding	32 bits unused
Total Interactive Mode 4 Reply IFF Data record size = 96 bits		

B.2.14 Interactive Mode 5 Reply IFF Data record

The specification of an Interactive Mode 5 reply to an Interactive Mode 5 interrogation shall be communicated by the Interactive Mode 5 Reply IFF Data record. It is included in Layer 5 for an interactive reply. The fields of this record are as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5012.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Interactive Mode 5 Reply*. This field shall specify the Mode 5 reply that would be sent by this IFF system to this Interactive Mode 5 interrogation. It shall use the same enumeration values as specified for the Mode 5 Reply field of the Mode 5 Transponder Status record (see B.2.32). It shall be represented by an 8-bit enumeration (see [UID 350]).

The format of the Interactive Mode 5 Reply IFF Data record shall be as shown in Table B.17.

Table B.17—Interactive Mode 5 Reply IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5012	32-bit enumeration
16	Record Length = 12	16-bit unsigned integer
8	Interactive Mode 5 Reply	8-bit enumeration
8	Padding	8 bits unused
32	Padding	32 bits unused
Total Interactive Mode 5 Reply IFF Data record size = 96 bits		

B.2.15 Interrogated Modes record

The specific modes and other basic data types related to an interactive interrogation, reply, or squitter transmission shall be communicated using the Interrogated Modes record. This is a non-self-identifying record that is contained in the Basic Interactive IFF Data record. The fields of this record are as follows:

a) *Mode 1*. This field shall specify if Mode 1 is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

b) *Mode 2*. This field shall specify if Mode 2 is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

c) *Mode 3/A*. This field shall specify if Mode 3/A is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

d) *Mode 4*. This field shall specify if Mode 4 is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

e) *Mode 5*. This field shall specify if Mode 5 is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

f) *Mode S*. This field shall specify if Mode S is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

g) *Mode C Altitude*. This field shall specify if Mode C Altitude is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

h) *TCAS/ACAS*. This field shall specify if TCAS/ACAS is being interrogated or contained in a transponder reply. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

i) *Special IP*. This field shall specify if this is a Special Identification of Position (IP) transmission. It shall be represented by a 1-bit enumeration set to Not Included (0) or Included (1).

j) *Squitter*. This field shall specify if this is an interactive squitter transmission. It shall be represented by a 1-bit enumeration value set to Not Included (0) or Included (1).

The format of the Interrogated Modes record shall be as shown in Table B.18.

Table B.18—Interrogated Modes record

Field name	Bits	Value
Not used	0	Not used
Mode 1	1	Enumeration
Mode 2	2	Enumeration
Mode 3/A	3	Enumeration
Mode 4	4	Enumeration
Mode 5	5	Enumeration
Mode S	6	Enumeration
Mode C Altitude	7	Enumeration
TCAS/ACAS	8	Enumeration
Special IP	9	Enumeration

Table B.18—Interrogated Modes record *(continued)*

Field name	Bits	Value
Squitter	10	Enumeration
Padding	11 to 31	Not used
Total Interrogated Modes record size = 32 bits		

B.2.16 Interrogator Side Lobe Suppression (ISLS) record

The Interrogator Side Lobe Suppression (ISLS) record is contained in the 24-bit System-Specific data field of an IFF Fundamental Parameter Data record, which may be included in Layer 2 of the IFF PDU. ISLS inhibits transponder replies to all challenges not radiated in the main beam of the interrogating antenna.

The fields of this record are as follows:

a) *Antenna Type*. This field shall indicate the type of antenna used by the system. This field shall be represented by an 8-bit enumeration (see [UID 349]).

b) *Pulse Rise Time*. This field shall contain the real-world time required for the leading edge of pulses P1, P2, and P3 to rise from 10 to 90 percent of its maximum voltage amplitude. Pulse Rise Time shall shall be expressed in one-hundredths of a microsecond where 1 is 0.01 µs and 99 is 0.99 µs. This field shall be represented by an 8-bit unsigned integer.

c) *Pulse Fall Time*. This field shall contain the real-world time required for the leading edge of pulses P1, P2, and P3 to fall from 10 to 90 percent of its maximum voltage amplitude. Pulse Fall Time shall be expressed in one-hundredths of a microsecond where 1 is 0.01 µs and 99 is 0.99 µs. This field shall be represented by an 8-bit unsigned integer.

The format of the ISLS record shall be as shown in Table B.19.

Table B.19—ISLS record

Field size (bits)	Field name	Data type
8	Antenna Type	8-bit enumeration
8	Pulse Rise Time	8-bit unsigned integer
8	Pulse Fall Time	8-bit unsigned integer
Total ISLS record size = 24 bits		

B.2.17 Mode 1 Code record

The Mode 1 Code record shall provide IFF Mode 1 information. When used, it shall be included as Parameter 1 of the Fundamental Operational Data record. The range is 00 to 73 with the first digit in the range 0 to 7 and the second digit in the range 0 to 3 reading left to right. For example, given a Mode 1 code of 73, the first digit (7) becomes Code Element 2 (A) and occupies bits 3 to 5 and digit (3) is Code Element 1 (B) and occupies bits 0 to 2 as shown in Table B.20. The fields of this record are as follows:

a) *Code Element 1*. This field shall identify Element 1 of the 2-digit Mode 1 code. The range shall be 0 octal to 3 octal. (This is also referred to as Code Element B and is the second digit reading left to right in an operational display of this mode.)
b) *Code Element 2*. This field shall identify Element 2 of the 2-digit Mode 1 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element A and is the first digit reading left to right in an operational display of this mode.)
c) *On/Off Status*. This 1-bit enumeration field shall indicate whether Mode 1 is On (1) or Off (0). The other fields of this record are independent of the On/Off Status field and shall remain unaffected when the status changes.
d) *Damage Status*. This 1-bit enumeration field shall indicate the damage status of the Mode 1 equipment and be set to either No Damage (0) or Damaged (1).
e) *Malfunction Status*. This 1-bit enumeration field shall indicate the malfunction status of the Mode 1 equipment and be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 1 Code record shall be as shown in Table B.20. The letters in parentheses in the Field Name column are the AIMS letter designation for each element. The valid data by System Name are shown in the Value column.

Table B.20—Mode 1 Code record

Field name	Bits	System name	Value
Code Element 1 (B)	0 to 2	Mark X	0; not used
		Mark X (SIF)	0 octal to 3 octal
		Mark X (A)	
		Mark XII	
		Mark XIIA	
Code Element 2 (A)	3 to 5	Mark X	0; not used
		Mark X (SIF)	0 octal to 7 octal
		Mark X (A)	
		Mark XII	
		Mark XIIA	
Padding	6 to 12	Mark X Mark X (SIF) Mark X (A) Mark XII Mark XIIA	0
On/Off Status	13		Enumeration
Damage	14		Enumeration
Malfunction	15		Enumeration
Total Mode 1 Code record size = 16 bits			

B.2.18 Mode 1 Interrogator Status record

The Mode 1 Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Mode 1.

When used, this record shall be included as Parameter 1 of the Fundamental Operational Data record for an interrogator capable of interrogating Mode 1. This is a 16-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode 1 interrogation capability is On (1) or Off (0).

b) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode 1 interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

c) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode 1 interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 1 Interrogator Status record shall be as shown in Table B.21.

Table B.21—Mode 1 Interrogator Status record

Field name	Bits	Value
Padding	0 to 12	0
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode 1 Interrogator Status record size = 16 bits		

B.2.19 Mode 2 Code record

The Mode 2 Code record shall provide Mode 2 code information. When used, this record shall be included as Parameter 2 of the Fundamental Operational Data record. This is a 16-bit record. The Mode 2 code range is 0000 to 7777. For example, given a Mode 2 code of 2467, the first digit (2) becomes Code Element 4 (A) and occupies bits 9 to 11, digit (4) is Code Element 3 (B), digit (6) is Code Element 2 (C) and digit (7) is Code Element 1 (D). They occupy the bit positions as shown in Table B.22. The fields of this record are as follows:

a) *Code Element 1*. This field shall identify Element 1 of the Mode 2 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element D and is the fourth digit reading left to right in an operational display of this mode.)

b) *Code Element 2*. This field shall identify Element 2 of the Mode 2 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element C and is the third digit reading left to right in an operational display of this mode.)

c) *Code Element 3*. This field shall identify Element 3 of the Mode 2 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element B and is the second digit reading left to right in an operational display of this mode.)

d) *Code Element 4*. This field shall identify Element 4 of the Mode 2 code. The range shall be 0 octal to 7 octal. (This is also referred to as Code Element A and is the first digit reading left to right in an operational display of this mode.)

e) *On/Off Status*. This 1-bit enumeration field shall indicate whether Mode 2 is On (1) or Off (0). The other fields of this record are independent of the On/Off Status field and shall remain unaffected when the status changes.

f) *Damage Status*. This 1-bit enumeration field shall indicate the damage status of the Mode 2 equipment and be set to either No Damage (0) or Damaged (1).

g) *Malfunction Status*. This 1-bit enumeration field shall indicate the malfunction status of the Mode 2 equipment and be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 2 Code record shall be as shown in Table B.22. The letters in parentheses in the Field Name column are the AIMS letter designation for each element. The valid data by System Name are shown in the Value column.

Table B.22—Mode 2 Code record

Field name	Bits	System name	Value
Code Element 1 (D)	0 to 2	Mark X	0; not used
		Mark X (SIF) Mark X (A) Mark XII Mark XIIA	0 octal to 7 octal
Code Element 2 (C)	3 to 5	Mark X	0; not used
		Mark X (SIF) Mark X (A) Mark XII Mark XIIA	0 octal to 7 octal
Code Element 3 (B)	6 to 8	Mark X	0; not used
		Mark X (SIF) Mark X (A) Mark XII Mark XIIA	0 octal to 7 octal
Code Element 4 (A)	9 to 11	Mark X	0; not used
		Mark X (SIF) Mark X (A) Mark XII Mark XIIA	0 octal to 7 octal
Padding	12	Mark X Mark X (SIF) Mark X (A) Mark XII Mark XIIA	0
On/Off Status	13		Enumeration
Damage Status	14		Enumeration
Malfunction Status	15		Enumeration
Total Mode 2 Code record size = 16 bits			

B.2.20 Mode 2 Interrogator Status record

The Mode 2 Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Mode 2.

When used, this record shall be included as Parameter 2 of the Fundamental Operational Data record for an interrogator capable of interrogating Mode 2. This is a 16-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode 2 interrogation capability is On (1) or Off (0).

b) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode 2 interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

c) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode 2 interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 2 Interrogator Status record shall be as shown in Table B.23.

Table B.23—Mode 2 Interrogator Status record

Field name	Bits	Value
Padding	0 to 12	0
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode 2 Interrogator Status record size = 16 bits		

B.2.21 Mode 3/A Code record

The Mode 3/A Code record shall provide Mode 3/A code information. When used, this record shall be included as Parameter 3 of the Fundamental Operational Data record. This is a 16-bit record. The range is 0000 to 7777. For example, given a Mode 3/A code of 7461, the first digit (7) becomes Code Element 4 (A) and occupies bits 9 to 11, digit (4) is Code Element 3 (B), digit (6) is Code Element 2 (C), and digit (1) is Code Element 1 (D). They occupy the bit positions as shown in Table B.24. The fields of this record are as follows:

a) *Code Element 1*. This 3-bit field shall identify Element 1 of the Mode 3/A code. The range shall be as shown in Table B.24 for each system and subsystem type. (This is also referred to as Code Element D and is the fourth digit reading left to right in an operational display of this mode.)

b) *Code Element 2*. This 3-bit field shall identify Element 2 of the Mode 3/A code. The range shall be as shown in Table B.24 for each system and subsystem type. (This is also referred to as Code Element C and is the third digit reading left to right in an operational display of this mode.)

c) *Code Element 3*. This 3-bit field shall identify Element 3 of the Mode 3/A code. The range shall be as shown in Table B.24 for each system and subsystem type. (This is also referred to as Code Element B and is the second digit reading left to right in an operational display of this mode.)

d) *Code Element 4*. This 3-bit field shall identify Element 4 of the Mode 3/A code. The range shall be as shown in Table B.24 for each system and subsystem type. (This is also referred to as Code Element A and is the first digit reading left to right in an operational display of this mode.)

e) *On/Off Status*. This 1-bit enumeration field shall indicate whether Mode 3/A is On (1) or Off (0). The other fields of this record are independent of the On/Off Status field and shall remain unaffected when the status changes.

f) *Damage Status*. This 1-bit enumeration field shall indicate the damage status of the Mode 3/A equipment and be set to either No Damage (0) or Damaged (1).

g) *Malfunction Status*. This 1-bit enumeration field shall indicate the malfunction status of the Mode 3/A equipment and be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 3/A Code record shall be as shown in Table B.24. The letters in parentheses in the field column are the AIMS letter designation for each element. The valid data by System Name are shown in the Value column.

Table B.24—Mode 3/A Code record

<table>
<tr><th>Field name</th><th>Bits</th><th>System name</th><th>Value</th></tr>
<tr><td rowspan="2">Code Element 1 (D)</td><td rowspan="2">0 to 2</td><td>Mark X
Mark X (SIF)</td><td>0; not used</td></tr>
<tr><td>Mark X (A)</td><td>0 octal to 7 octal</td></tr>
<tr><td rowspan="2">Code Element 2 (C)</td><td rowspan="2">3 to 5</td><td>Mark X
Mark X (SIF)</td><td>0; not used</td></tr>
<tr><td>Mark X (A)</td><td>0 octal to 7 octal</td></tr>
<tr><td rowspan="2">Code Element 3 (B)</td><td rowspan="2">6 to 8</td><td>Mark X
Mark X (SIF)</td><td>0; not used</td></tr>
<tr><td>Mark X (SIF)
Mark X (A)
Mark XII
Mark XIIA</td><td>0 octal to 7 octal</td></tr>
<tr><td rowspan="2">Code Element 4 (A)</td><td rowspan="2">9 to 11</td><td>Mark X
Mark X (SIF)</td><td>0; not used</td></tr>
<tr><td>Mark X (SIF)
Mark X (A)
Mark XII
Mark XIIA</td><td>0 octal to 7 octal</td></tr>
<tr><td>Padding</td><td>12</td><td>—</td><td>0</td></tr>
<tr><td>On/Off Status</td><td>13</td><td rowspan="3">Mark X
Mark X (SIF)
Mark X (A)</td><td>Enumeration</td></tr>
<tr><td>Damage Status</td><td>14</td><td>Enumeration</td></tr>
<tr><td>Malfunction Status</td><td>15</td><td>Enumeration</td></tr>
<tr><td colspan="4">Total Mode 3/A Code record size = 16 bits</td></tr>
</table>

B.2.22 Mode 3/A Interrogator Status record

The Mode 3/A Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Mode 3/A.

When used, this record shall be included as Parameter 3 of the Fundamental Operational Data record for an interrogator capable of interrogating Mode 3/A. This is a 16-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode 3/A interrogation capability is On (1) or Off (0).

b) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode 3/A interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

c) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode 3/A interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 3/A Interrogator Status record shall be as shown in Table B.25.

Table B.25—Mode 3/A Interrogator Status record

Field name	Bits	Value
Padding	0 to 12	0
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode 3/A Interrogator Status record size = 16 bits		

B.2.23 Mode 4 Code record

The Mode 4 Code record shall contain a Mode 4 pseudo crypto code or an indication to use the Alternate Mode 4 Challenge/Reply (B.2.19). When used, this record shall be included as Parameter 4 of the Fundamental Operational Data record. This is a 16-bit record. See B.5.1.2.2 for the special issuance and receipt rules associated with this record. The fields of this record are as follows:

a) *Mode 4 Code*. This field shall contain either a pseudo crypto code or an indication to use the Alternate Mode 4 Challenge/Reply. The valid pseudo crypto code range shall be 0 to 4094 decimal with 4095 reserved to indicate there is no pseudo crypto code and to use the Alternate Mode 4 Challenge/Reply instead. It shall be represented by a 11-bit unsigned integer.

b) *On/Off Status*. This 1-bit enumeration field shall indicate whether Mode 4 is On (1) or Off (0). The other fields of this record are independent of the On/Off Status field and shall remain unaffected when the status changes.

c) *Damage Status*. This 1-bit enumeration field shall indicate the damage status of the Mode 4 equipment and be set to either No Damage (0) or Damaged (1).

d) *Malfunction Status*. This 1-bit enumeration field shall indicate the malfunction status of the Mode 4 equipment and be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 4 Code record shall be as shown in Table B.26.

Table B.26—Mode 4 Code record

Field name	Bits	Value
Mode 4 Code	0 to 11	Unsigned integer
Padding	12	0
On/Off Status	13	Enumeration

Table B.26—Mode 4 Code record *(continued)*

Field name	Bits	Value
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode 4 Code record size = 16 bits		

B.2.24 Mode 4 Interrogator Status record

The Mode 4 Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Mode 4.

When used, this record shall be included as Parameter 4 of the Fundamental Operational Data record for an interrogator capable of interrogating Mode 4. This is a 16-bit record. The fields of this record are as follows:

a) *Mode 4 Code*. This field shall either contain a pseudo crypto code or an indication that only the Alternate Mode 4 Challenge/Reply field value is provided [see item a) in B.2.1.3.2]. The valid pseudo crypto code range shall be 0 to 4094 decimal with 4095 reserved to indicate there is no pseudo crypto code and to use the Alternate Mode 4 Challenge/Reply field value instead. It shall be represented by a 12-bit unsigned integer.

   NOTE—The Alternate Mode 4 Challenge/Reply field is a required field for any interrogator or transponder that has a Mode 4 capability regardless of whether a pseudo crypto code is included. There are special Mode 4 issuance and receipt rules (see B.5.1.2.2).

b) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode 4 interrogation capability is On (1) or Off (0).

c) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode 4 interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

d) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode 4 interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 4 Interrogator Status record shall be as shown in Table B.27.

Table B.27—Mode 4 Interrogator Status record

Field name	Bits	Value
Mode 4 Code	0 to 11	12-bit unsigned integer
Padding	12	1 bit unused
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode 4 Interrogator Status record size = 16 bits		

The Mode 4 Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Mode 4.

When used, this record shall be included as Parameter 4 of the Fundamental Operational Data record for an interrogator capable of interrogating Mode 4. This is a 16-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode 4 interrogation capability is On (1) or Off (0).

b) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode 4 interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

c) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode 4 interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 4 Interrogator Status record shall be as shown in Table B.28.

Table B.28—Mode 4 Interrogator Status record

Field name	Bits	Value
Padding	0 to 12	0
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode 4 Interrogator Status record size = 16 bits		

B.2.25 Mode 5 Altitude record

The Mode 5 Altitude record shall be used to specify Mode 5 altitude values associated with a Mode 5 Transponder Location IFF Data record. Not all fields of that record are required to contain realistically simulated latitude, longitude, and altitude data. If realistically simulated Mode 5 altitude data is being reported, this record shall be updated at a rate and in a manner that reflects ICAO requirements related to the transmission of this type of information (see Clause 2). If realistically simulated Mode 5 altitude data is not being reported, the Mode 5 Altitude field of this record shall indicate to use entity truth data to derive Mode 5 altitude instead. This is a 16-bit record.

The fields of this record shall be as follows:

a) *Mode 5 Altitude*. This 11-bit field shall indicate the Mode 5 altitude in the resolution specified by the Altitude Resolution field. If this field is not being set to a realistically simulated altitude, the numeric value of 4095 shall be set to indicate to use entity truth data. For example, if the altitude is 1000 feet and the Altitude Resolution field is set to 100-foot increments, the Mode 5 altitude value in this field would be set to 10.

b) *Altitude Resolution*. This 1-bit enumeration field (see [UID 361]) shall indicate the resolution of the altitude as either 100-foot (0) or 25-foot (1) resolution.

The format of the Mode 5 Altitude record shall be as shown in Table B.29.

Table B.29—Mode 5 Altitude record

Field	Bits
Mode 5 Altitude	0 to 10
Padding	11
Altitude Resolution	12
Padding	13 to 15
Total Mode 5 Altitude record size = 16 bits	

B.2.26 Mode 5 Interrogator Basic Data record

The Mode 5 Interrogator Basic Data record is included in the Mode 5 Interrogator format for Layer 3 of the IFF PDU. This record shall consist of fields and other records as follows:

a) *Mode 5 Interrogator Status*. This field shall indicate the status of a Mode 5 interrogator. It shall be represented by a Mode 5 Interrogator Status record (see B.2.27).

b) *Mode 5 Message Formats Present*. This field shall indicate the Mode 5 Message Formats supported by this Mode 5 interrogator. See B.2.28 for requirements related to setting this field for a Mode 5 interrogator. This field shall be represented by a Mode 5 Message Formats record (see B.2.28).

c) *Padding*. The padding fields shown in Table B.30 are intended to maintain the same structure as the Mode 5 Transponder Basic Data field. This padding can be used for future fixed fields applicable to Mode 5 interrogators.

d) *Interrogated Entity ID*. This field shall specify the Entity ID of the entity to which an active interrogation is being directed (see B.2.27). If there is no active interrogation, this field shall be set to NO_SPECIFIC_ENTITY. It shall be represented by an Entity ID record (see 6.2.28).

The format of the Mode 5 Interrogator Basic Data record shall be as shown in Table B.30.

Table B.30—Mode 5 Interrogator Basic Data record

Field size (bits)	Field name	Data type
8	Mode 5 Interrogator Status	Mode 5 Interrogator Status record
8	Padding	8 bits unused
16	Padding	16 bits unused
32	Mode 5 Message Formats Present	Mode 5 Message Formats record
48	Interrogated Entity ID	Entity ID record
16	Padding	16 bits unused
Total Mode 5 Interrogator Basic Data record size = 128 bits		

B.2.27 Mode 5 Interrogator Status record

The Mode 5 Interrogator Status record shall provide Mode 5 interrogator information. This record is included in the Mode 5 Interrogator Basic Data record (see B.2.26) for an interrogator capable of interrogating Mode 5. Knowledge of AIMS 03-1000A, and specific interrogator model specifications are required to properly implement these fields. This is an 8-bit record. The fields of this record are as follows:

a) *IFF Mission*. This 3-bit enumeration field (see [UID 346]) shall identify the IFF mission of this Mode 5 interrogator. Refer to AIMS 03-1000A to determine the applicable IFF mission. (Some of the IFF Mission enumerations are not applicable to an interrogator.)

b) *Message Formats Status*. This 1-bit enumeration field (see [UID 380]) shall indicate whether the Mode 5 Message Formats record indicates a Capability (0) or an Active Interrogation (1):

   1) When set to Capability (0), it indicates that the Mode 5 Message Formats record includes all message formats the interrogator is able to interrogate.

   2) When set to Active Interrogation (1), it indicates that this is an active interrogation and the Mode 5 Message Formats record only contains the specific message format(s) currently being requested from a Mode 5 transponder.

c) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode 5 interrogation capability is On (1) or Off (0).

d) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode 5 interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

e) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode 5 interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 5 Interrogator Status record shall be as shown in Table B.31.

Table B.31—Mode 5 Interrogator Status record

Field name	Bits	Value
IFF Mission	0 to 2	Enumeration
Message Formats Status	3	Enumeration
Padding	4	Unused
On/Off Status	5	Enumeration
Damage Status	6	Enumeration
Malfunction Status	7	Enumeration
Total Mode 5 Interrogator Status record size = 8 bits		

B.2.28 Mode 5 Message Formats record

The Mode 5 Message Formats record shall be used to indicate the message formats associated with a Mode 5 transponder or interrogator as follows:

a) *Mode 5 Transponder*. When a Mode 5 transponder is in the Regeneration Mode, the included message formats shall be those that this Mode 5 transponder is capable of supporting and could be in a reply to a Mode 5 interrogation. The Message Formats Status field of the Mode 5 Transponder Status record is set to Capability (0) in this case.

When a Mode 5 transponder is in the Interactive Mode, the requirements specified in B.2.11 (Interactive Basic Mode 5 IFF Data record) are applicable.

b) *Mode 5 Interrogator*. When a Mode 5 interrogator is in the Regeneration Mode, the included message formats shall be either:

   1) Those that this Mode 5 interrogator is capable of supporting and could be sent in a Mode 5 interrogation as indicated by the Message Formats Status field being set to Capability (0).

   2) Only the specific message formats associated with this Mode 5 interrogator's current active interrogation as indicated by the Message Formats Status field being set to Active Interrogation (1).

   When a Mode 5 interrogator is in the Interactive Mode, the requirements specified in B.2.11 (Interactive Basic Mode 5 IFF Data record) are applicable.

This record shall be included in the Mode 5 Message Formats Present field of the Mode 5 Transponder Basic Data record and also the Mode 5 Interrogator Basic Data record contained in Layer 3. This is a 32-bit record. The fields of this record are as follows:

— *Message Formats 0 to 31*. Each bit shall represent a specific message format as shown in Table B.32. It shall be represented by an enumeration as follows: Not Present (0) and Present (1).

The format of the Mode 5 Message Formats record shall be as shown in Table B.32.

Table B.32—Mode 5 Message Formats record

Field name	Bits	Value
Message Format 0	0	Enumeration
• • •		
Message Format 31	31	Enumeration
Total Mode 5 Message Formats record size = 32 bits		

B.2.29 Mode 5 Transponder Basic Data record

The Mode 5 Transponder Basic Data record is included in the Mode 5 Transponder format for Layer 3 of the IFF PDU. This record consists of fields and other records as follows:

a) *Mode 5 Status*. This field shall specify the status of the Mode 5 transponder and shall be represented by a Mode 5 Transponder Status record (see B.2.32).

b) *Personal Identification Number*. This field shall contain the Personal Identification Number (PIN) that uniquely identifies the platform carrying this Mode 5 transponder. The PIN shall be a number in the range 00 000 octal to 37 777 octal and shall be represented as a 16-bit unsigned integer.

c) *Mode 5 Message Formats Present*. This field shall indicate the Mode 5 Message Formats supported by the data contained in Layer 3. See B.2.28 for requirements related to setting this field for a Mode 5 transponder. This field shall be represented by the Mode 5 Message Formats record (see B.2.28).

d) *Enhanced Mode 1*. This field shall be represented by the Enhanced Mode 1 Code record (see B.2.9).

e) *National Origin*. This field shall contain the national origin. The national origin is the country that owns the platform carrying a Mode 5 transponder. This field shall be represented by a 16-bit enumeration whose values are available through appropriate government agencies to authorized personnel.

NOTE—The National Origin is not the same as the Country field enumerations contained in Entity Type records. For simulations that have modeled Mode 5 (e.g., Mark XIIA) transponders based on the original AIMS 03-1000, the AIMS 03-1000A standard contains instructions on how to set separate National Origin and Mission Code values to equate to the present National Origin field.

f) *Supplemental Data*. This field shall contain supplemental data for a Mode 5 transponder. This field shall be represented by the Mode 5 Transponder SD record (see B.2.31).

g) *Navigation Source*. This field shall indicate the Navigation source. This field shall be represented by an 8-bit enumeration (see [UID 359]).

h) *Figure of Merit*. This field shall contain the Figure of Merit. The Figure of Merit shall be a number in the range of 0 to 31 decimal. This field shall be represented by an 8-bit unsigned integer.

The format of the Mode 5 Transponder Basic Data record is shown in Table B.33.

Table B.33—Mode 5 Transponder Basic Data record

Field size (bits)	Field name	Data type
16	Mode 5 Status	Mode 5 Transponder Status record
16	Personal Identification Number (PIN)	16-bit unsigned integer
32	Mode 5 Message Formats Present	Mode 5 Message Formats record
16	Enhanced Mode 1	Enhanced Mode 1 Code record
16	National Origin	16-bit enumeration
8	Supplemental Data	Mode 5 Transponder SD record
8	Navigation Source	8-bit enumeration
8	Figure of Merit	8-bit unsigned integer
8	Padding	8 bits unused
Total Mode 5 Transponder Basic Data record size = 128 bits		

B.2.30 Mode 5 Transponder Location IFF Data record

The specification of the real-time location and altitude information transmitted by a Mode 5 transponder shall be communicated using the Mode 5 Transponder Location IFF Data record. While some Mode 5 message formats use different field sizes for latitude and longitude (see AIMS 03-1000A), the Latitude and Longitude fields in this record use a standard field size. When this record is included in an IFF PDU, the IFF PDU shall be sent at a rate commensurate with the rate specified in AIMS 03-1000A for the associated platform type. If location or altitude errors are modeled, the information contained in this record shall reflect the inclusion of those errors. This is an optional record. However, if implemented, the requirements specified herein shall apply. The fields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5001.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Latitude*. This field shall specify the latitude of the entity. The latitude shall be represented as two's complement coded latitude with the Most Significant Bit (MSB) equal to –90 degrees.

d) *Longitude*. This field shall specify the longitude of the entity. The longitude shall be represented as two's complement coded longitude with the Most Significant Bit (MSB) equal to –180 degrees.

e) *Mode 5 Altitude*. This field shall specify the Mode 5 Altitude of the entity:

   1) Mode 5 Altitude. This subfield shall contain the Mode 5 altitude. It shall be represented as a binary coded altitude referenced to –1000 feet. All altitudes greater than 50 175 feet shall be reported with 100-foot resolution. If this field does not contain the reported Mode 5 altitude, it shall be set to 2047 (decimal) to indicate to use entity truth altitude to derive the Mode 5 altitude.

   2) Altitude Resolution. This 1-bit enumeration subfield shall contain the altitude resolution. Mode 5 altitude shall be reported as either 25-foot (1) or 100-foot (0) resolution, whichever is applicable. The assignment of a bit value of 0 for 100-foot resolution (e.g., 12 700 feet, 12 800 feet, and 12 900 feet) and 1 for 25-foot resolution (e.g., 12 700 feet, 12 725 feet, 12 750 feet, or 12 775 feet) are the assignments made for the actual Attitude Resolution field of a Mode 5 Altitude transponder message as documented in AIMS 03-1000A.

f) *Barometric Altitude*. This field shall specify the Barometric Altitude of the entity:

   1) Barometric Altitude. The Barometric altitude shall be referenced to –1000 feet. If this field does not contain the reported Barometric altitude, it shall be set to 2047 decimal to indicate to use entity truth altitude to derive the Barometric altitude.

   2) Altitude Resolution. This 1-bit enumeration subfield shall contain the altitude resolution. Barometric altitude shall be reported as either 25-foot (1) or 100-foot (0) resolution, whichever is applicable. The assignment of a bit value of 0 for 100-foot resolution (e.g., 51 700 feet and 51 800 feet) and 1 for 25-foot resolution (e.g., 12 700 feet, 12 725 feet, and 12 750 feet) are the assignments made for the actual Attitude Resolution field of a Mode 5 Altitude transponder message as documented in AIMS 03-1000A.

The format of the Mode 5 Transponder Location IFF Data record shall be as shown in Table B.34.

Table B.34—Mode 5 Transponder Location IFF Data record

Field size (bits)	Field name	AIMS 03-1000A format	Bits
32	Record Type = 5001	32-bit enumeration	—
16	Record Length = 20	16-bit unsigned integer	—
16	Padding	16 bits unused	—
32	Latitude	Latitude	0 to 17
		Padding	18 to 31
32	Longitude	Longitude	0 to 17
		Padding	18 to 31

Table B.34—Mode 5 Transponder Location IFF Data record *(continued)*

Field size (bits)	Field name	AIMS 03-1000A format	Bits
16	Mode 5 Altitude	Mode 5 Altitude	0 to 10
		Padding	11
		Altitude Resolution—1-bit enumeration	12
		Padding	13 to 15
16	Barometric Altitude	Barometric Altitude	0 to 11
		Altitude Resolution—1-bit enumeration	12
		Padding	13 to 15
Total Mode 5 Transponder Location IFF Data record size = 160 bits			

B.2.31 Mode 5 Transponder Supplemental Data (SD) record

The Mode 5 Transponder Supplemental Data (SD) record shall provide supplemental data related to a Mode 5 transponder. This record shall be included in the Supplemental Data field of the Mode 5 Transponder Basic Data record, contained in Layer 3. Knowledge of AIMS 03-1000A and specific Mode 5 transponder specifications are required to properly implement the fields of this record. This is an 8-bit record. The fields of this record are as follows:

a) *Squitter On/Off Status*. This 1-bit enumeration field shall indicate whether Mode 5 squitter is On (1) or Off (0).

b) *Level 2 Squitter Status*. This 1-bit enumeration field (see [UID 353]) shall indicate whether Mode 5 Level 2 squitter is Disabled (0) or Enabled (1).

c) *IFF Mission*. This 3-bit enumeration field (see [UID 346]) shall identify the IFF mission of this Mode 5 transponder. Refer to AIMS 03-1000A to determine the applicable IFF mission. (Some of the IFF Mission enumerations are not applicable to a transponder.)

The format of the Mode 5 Transponder SD record shall be as shown in Table B.35.

Table B.35—Mode 5 Transponder SD record

Field name	Bits	Value
Squitter On/Off Status	0	Enumeration
Level 2 Squitter Status	1	Enumeration
IFF Mission	2 to 4	Enumeration
Padding	5 to 7	0
Total Mode 5 Transponder SD record size = 8 bits		

B.2.32 Mode 5 Transponder Status record

The Mode 5 Transponder Status record shall be used to communicate Mode 5 status information. It shall be represented by a 16-bit record. This record is a field in the Mode 5 Transponder Basic Data record included in Layer 3 of the IFF PDU.

The fields of this record shall be as follows:

a) *Mode 5 Reply*. This 4-bit enumeration field shall specify the validity of a reply that would be transmitted by a Mode 5 transponder if interrogated (see [UID 350]):

    1) The following field values shall be capable of being set: No Response (0), Valid (1), and Invalid (2). The Unable to Verify (3) value is optional. See appropriate military documentation for the criteria used to set each of these values.

    NOTE—Care needs to be taken when discussing the details of Mode 5 as certain information related to this capability are classified.

    2) If the Crypto Control IFF Data record is also implemented, appropriate fields in that record shall match the equivalent Mode 5 Reply values used in an exercise. See the exercise agreement for information regarding how to correlate these items.

b) *Line Test*. This 1-bit enumeration field shall indicate whether a line test is in progress and shall be set to either Not Enabled (0) or Enabled (1).

c) *Antenna Selection*. This 2-bit enumeration field (see [UID 351]) shall indicate the present Mode 5 transponder antenna selection and shall be set to either No Statement (0) if the Mode 5 antenna position is not available or to Top (1), Bottom (2), or Diversity (3).

d) *Crypto Control*. This 1-bit enumeration field shall indicate whether the Crypto Control IFF Data record is included to indicate that pseudo crypto control is present and be set to either Crypto Control IFF Data Record Not Present (0) or Crypto Control IFF Data Record Present (1).

e) *Lat/Long/Altitude Source*. This 1-bit enumeration field shall indicate the source of Mode 5 latitude, longitude, and altitude information. It shall be set to Compute Locally (0) or to Mode 5 Transponder Location IFF Data Record Present (1).

f) *Location Errors*. This 1-bit enumeration field (see [UID 423]) shall indicate whether there are location errors associated with Mode 5 latitude, longitude, or altitude. It shall be set to No Location Errors (0) or to Transponder Location Error IFF Data Record Present (1).

g) *Platform Type*. This 1-bit enumeration field (see [UID 396]) shall indicate the type of platform (air or ground) that is associated with this Mode 5 transponder. This field shall be set as follows:

    1) Ground Vehicle (0) if the Entity Type record, Domain field is Land (1) regardless of whether the entity is actually a land vehicle or some other type of land platform.

    2) Air Vehicle (1) if the Entity Type record, Domain field is Air (2) regardless of whether the entity is actually an aircraft or some other type of air platform (e.g., a blimp).

h) *Mode 5 Level Selection*. This 1-bit enumeration field (see [UID 412]) shall indicate whether Mode 5 Level 1 (0) or Mode 5 Level 2 (1) is currently selected for this Mode 5 transponder. (The Level 2 selection includes both Level 1 and Level 2 message formats.)

i) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode 5 transponder is On (1) or Off (0).

j) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode 5 transponder. This field shall be set to either No Damage (0) or Damaged (1).

k) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode 5 transponder. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode 5 Transponder Status record shall be as shown in Table B.36.

Table B.36—Mode 5 Transponder Status record

Field name	Bits	Value
Mode 5 Reply	0 to 3	Enumeration
Line Test	4	Enumeration
Antenna Selection	5 to 6	Enumeration
Crypto Control	7	Enumeration
Lat/Long/Altitude Source	8	Enumeration
Location Errors	9	Enumeration
Platform Type	10	Enumeration
Mode 5 Level Selection	11	Enumeration
Padding	12	Unused
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode 5 Transponder Status record size = 16 bits		

B.2.33 Mode C Altitude record

The Mode C Altitude record shall provide Mode C encoded altitude information. When used, this record shall be included as Parameter 5 of the Fundamental Operational Data record. This is a 16-bit record. The fields of this record are as follows:

a) *Altitude Indicator*. This 1-bit enumeration field shall indicate one of two conditions as follows (see [UID 340]):
   1) Positive/Negative Indicator. If the Mode C Altitude field contains the Mode C altitude, then the Altitude Indicator shall indicate either a Positive Altitude Above MSL (0) or a Negative Altitude Below MSL (1).
   2) Alternate Mode C Indicator. If the Mode C Altitude field contains the value 2047, then the Altitude Indicator shall be set to 1.

b) *Mode C Altitude*. This field shall either contain the actual Mode C altitude or indicate to use the Alternate Mode C altitude source (i.e., derived from entity truth data). A value of 2047 indicates to use Alternate Mode C Altitude. When actual Mode C altitude is present, it shall be in the range 0 feet to 126 000 feet in 100-foot increments based on the Neg/Pos Indicator. It shall be represented by a 11-bit unsigned integer.

c) *On/Off Status*. This 1-bit enumeration field shall indicate whether Mode C is On (1) or Off (0). The other fields of this record are independent of the On/Off Status field and shall remain unaffected when the status changes.

d) *Damage Status*. This 1-bit enumeration field shall indicate the damage status of the Mode C equipment and be set to either No Damage (0) or Damaged (1).

e) *Malfunction Status*. This 1-bit enumeration field shall indicate the malfunction status of the Mode C equipment and be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode C Altitude record shall be as shown in Table B.37.

Table B.37—Mode C Altitude record

Field name	Bits	Value
Altitude Indicator	0	Enumeration
Mode C Altitude	1 to 11	Unsigned integer
Padding	12	0
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode C Altitude record size = 16 bits		

B.2.34 Mode C Interrogator Status record

The Mode C Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Mode C.

When used, this record shall be included as Parameter 5 of the Fundamental Operational Data record for an interrogator capable of interrogating Mode C. This is a 16-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode C interrogation capability is On (1) or Off (0).

b) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode C interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

c) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode C interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode C Interrogator Status record shall be as shown in Table B.38.

Table B.38—Mode C Interrogator Status record

Field name	Bits	Value
Padding	0 to 12	0
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode C Interrogator Status record size = 16 bits		

B.2.35 Mode S Aircraft Identification record

Mode S transponder requirements include the capability to report an Aircraft Identification. This is the International Civil Aviation Organization (ICAO) flight identity of an aircraft. According to ICAO requirements, the flight identity will be either the flight ID (i.e., call sign) listed in Item 7 of an ICAO flight plan or, if no flight plan has been filed, the registration number (also referred to as the tail number) of the aircraft. When Aircraft Identification is being reported in the IFF PDU, it shall be specified using the Mode S Aircraft Identification record. This record shall consist of an eight-element American Standard Code for Information Interchange (ASCII) character string consisting of the letters A to Z and the numbers 0 to 9. An aircraft identification may be less than eight characters. This string shall begin with the most significant octet located at the lowest address. Unused characters shall be indicated by Null (0) values. The format of the Mode S Aircraft Identification record shall be as shown in Table B.39.

Table B.39—Mode S Aircraft Identification record

Field size (bits)	Field name	Data type
8	1st Character	8-bit unsigned integer
8	2nd Character	8-bit unsigned integer
	• • •	
8	8th Character	8-bit unsigned integer
Total Mode S Aircraft Identification record size = 64 bits		

B.2.36 Mode S Altitude record

The Mode S Altitude record shall be used to specify the Mode S altitude associated with a Mode S transponder. It is based on the fields of the ICAO Mode S altitude. The fields of this record shall be as follows:

a) *Mode S Altitude*. This field shall indicate the Mode S altitude. It shall be either set to a value that indicates to use entity truth altitude (2047 decimal), to compute the received Mode S altitude, or may contain the actual Mode S altitude as determined by the Mode S transponder model. When it is reporting actual Mode S altitude, it shall be represented as a binary coded altitude referenced to –1000 feet. All altitudes greater than 50 175 feet shall be reported with 100-foot resolution. A receiving simulation is not required to process an actual Mode S altitude value by this standard and may always compute it from the entity truth altitude.

   Note that reporting actual Mode S altitude in this record should be limited to the Interactive Mode due to the requirement to report Mode S altitude whenever it changes by either 25 feet or 100 feet depending on the Resolution field setting. Normally, this field should contain the computed Mode S altitude based on the truth altitude of the associated entity.

b) *Altitude Resolution*. This 1-bit enumeration field (see [UID 361]) shall indicate the Mode S altitude resolution as being reported in either 100-foot (0) or 25-foot (1) increments.

The format of the Mode S Altitude record shall be as shown in Table B.40.

Table B.40—Mode S Altitude record

Field name	Bits	Value
Mode S Altitude	0 to 10	11-bit unsigned integer
Altitude Resolution	11	Enumeration
Padding	12 to 15	4 bits unused
Total Mode S Altitude record size = 8 bits		

B.2.37 Mode S Interrogator Basic Data record

The Mode S Interrogator Basic Data record is included in the Mode S Interrogator format for Layer 4 of the IFF PDU. This record shall consist of fields and other records as follows:

a) *Mode S Interrogator Status*. This field shall indicate the status of a Mode S interrogator. It shall be represented by a Mode S Interrogator Status record (see B.2.39).

b) *Mode S Levels Present*. This field shall indicate the Mode S levels present. All levels that would be able to be responded to for a Mode S interrogation shall be set. It shall be represented by a Mode S Levels Present record (see B.2.40).

c) *Padding*. The padding fields shown in Table B.41 are intended to maintain the same structure as the Mode S Transponder Basic Data field. This padding can be used for future fixed fields applicable to Mode S interrogators.

The format of the Mode S Interrogator Basic Data record is shown in Table B.41.

Table B.41—Mode S Interrogator Basic Data record

Field size (bits)	Field name	Data type
8	Mode S Interrogator Status	Mode S Interrogator Status record
8	Padding	8 bits unused
8	Mode S Levels Present	Mode S Levels Present record
8	Padding	8 bits unused
160	Padding	Padding—5 × 32 bits unused (160 bits)
Total Mode S Interrogator Basic Data record size = 192 bits		

B.2.38 Mode S Interrogator Identifier record

B.2.38.1 Introduction

The Mode S Interrogator Identifier record shall provide Mode S Interrogator Code (IC) information. When used, this record shall be included as Parameter 4 of the Fundamental Operational Data record. This is a 16-bit record.

B.2.38.2 General requirements

The requirements for this record are as follows:

a) The assignment of Mode S interrogator codes is made by appropriate military and civilian agencies and has to be coordinated prior to a simulation exercise to prevent the assignment of the same Mode S interrogator codes to two or more Mode S interrogators, of the same IC Type in overlapping surveillance coverage. In real-world operations, this could deny the availability of Mode S data to one of the duplicate Mode S interrogators.

b) Both Interrogator Identifier (II) and Surveillance Identifier (SI) Codes shall be able to be set in the full range of numbers shown in Table B.42.

c) There is no requirement as part of this standard to be able to include a Secondary IC Code for a Mode S interrogator. However, if a simulation implements a Secondary IC Code, it shall use the Mode S Interrogator Identifier record for that purpose.

d) If only one IC Code is assigned to a Mode S Interrogator, then it shall be the Primary IC Code and bits 8–15 shall be set to all zeroes.

B.2.38.3 Record format

The fields of this record are as follows:

a) *Primary IC Code*. This field shall contain two subfields as follows:
   1) Interrogator Code (IC) Type. This field shall specify the IC type. It shall be represented by a 1-bit enumeration (see [UID 348]).
   2) Interrogator Code (IC). This field shall contain the interrogator code. It shall be a 7-bit decimal number as follows:
      i) Interrogator Identifier (II) Codes: 0 to 16 decimal.
      ii) Surveillance Identifier (SI) Codes: 0 to 64 decimal.

b) *Secondary IC Code*. This field shall contain two subfields as follows:
   1) Interrogator Code (IC) Type. This field shall specify the IC type. It shall be represented by a 1-bit enumeration (see [UID 348]).
   2) Interrogator Code. This field shall contain the interrogator code. It shall be a 7-bit decimal number.

The format of the Mode S Interrogator Identifier record shall be as shown in Table B.42.

Table B.42—Mode S Interrogator Identifier record

Field name	Bits	Value
Primary IC Code	0 to 7	—
IC Type	0	Enumeration
IC Code	1 to 7	II: 0 to 16 decimal SI: 0 to 64 decimal
Secondary IC Code	8to 15	—
IC Type	8	Enumeration
IC Code	9 to 15	II: 0 to 16 decimal SI: 0 to 64 decimal
Total Mode S Interrogator Identifier record size = 16 bits Legend IC Type: II - Interrogator Identifier SI = Surveillance Identifier		

B.2.39 Mode S Interrogator Status record

The Mode S Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Mode S.

This record is included in the Mode S Interrogator Basic Data record (see B.2.37) for an interrogator capable of interrogating Mode S. This is an 8-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode S interrogation capability is On (1) or Off (0).

b) *Transmit State*. This 3-bit enumeration field shall indicate the state of the Mode S interrogator (see [UID 347]). This field shall be set to No Statement (0) when the Mode S interrogator simulation is operating in the Regeneration Mode. It may be set to other values if operating in the Interactive Mode.

c) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode S interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).

d) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode S interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).The format of the Mode S Interrogator Status record shall be as shown in Table B.43.

Table B.43—Mode S Interrogator Status record

Field name	Bits	Value
On/Off Status	0	Enumeration
Transmit State	1 to 3	Enumeration
Damage Status	4	Enumeration
Malfunction Status	5	Enumeration
Padding	6 to 7	2 bits unused
Total Mode S Interrogator Status record size = 8 bits		

B.2.40 Mode S Levels Present record

The Mode S Levels Present record shall be used to indicate which Mode S levels this transponder or interrogator is able to support. This record shall be included for a Mode S transponder and interrogator. This is an 8-bit record. The fields of this record are as follows:

a) *Level Fields*. Each Level field shown in Table B.44 is a 1-bit field that shall denote whether a specific Mode S level or sublevel is Not Present (0) or Present (1).

The format of the Mode S Levels Present record shall be as shown in Table B.44.

Table B.44—Mode S Levels Present record

Field name	Bits	Value
Not used	0	1 bit unused
Level 1	1	Enumeration
Level 2 (Elementary Surveillance sublevel)	2	Enumeration
Level 2 (Enhanced Surveillance sublevel)	3	Enumeration
Level 3	4	Enumeration
Level 4	5	Enumeration
Not used	6, 7	Unused
Total Mode S Levels Present record size = 8 bits		

Table B.45 indicates the allocation of Mode S data to IFF PDU Layers. An “X” in a Level column indicates that level includes that Mode S Data element.

Table B.45—Mode S Transponder Levels

Mode S data element	Level 1	Level 2 elementary surveillance	Level 2 enhanced surveillance	IFF PDU layer	Section
Mode 3/A Code	X	X	X	1	B.2.21
Mode C Altitude	X	X	X	1	B.2.33
Aircraft Address	X	X	X	4	7.6.5.5.3
Capability Report	—	X	X	4	7.6.5.5.3
Altitude Reporting *Mode C Altitude*	—	X	X	4	B.2.33
Flight Status *Aircraft Present Domain*	—	X	X	4	B.2.41
Indicated Air Speed	—	—	X	4	B.2.6
Mach Number	—	—	X	4	B.2.6
Ground Speed	—	—	X	4	B.2.6
Magnetic Heading	—	—	X	4	B.2.6
Track Angle Rate	—	—	X	4	B.2.6
True Track Angle	—	—	X	4	B.2.6
True Airspeed	—	—	X	4	B.2.6
Vertical Rate	—	—	X	4	B.2.6
Squitter	X	X	X	4	B.2.31
— Acquisition (Short) Squitter	X	X	X	4	B.2.47 B.2.48
— Extended Squitter	X	X	X	4	

B.2.41 Mode S Transponder Basic Data record

The Mode S Transponder Basic Data record is included in the Mode S Transponder format for Layer 4 of the IFF PDU. This record consists of fields and other records as follows:

a) *Mode S Status*. This field shall specify the status of the Mode S transponder and shall be represented by a Mode S Transponder Status record (see B.2.42).

b) *Mode S Levels Present*. This field shall indicate the Mode S levels present. All levels that would be able to be responded to for a Mode S interrogation shall be set. It shall be represented by a Mode S Levels Present record (see B.2.40).

c) *Aircraft Present Domain*. This field shall indicate the present domain of the aircraft as either No Statement (0), Airborne (1), or On Ground/Surface (2). Surface means on a water platform surface such as a ship deck or on the water surface directly (e.g., pontoon-equipped aircraft). This field is referred to as Flight Status in ICAO Publications. It shall be represented by an 8-bit enumeration (see [UID 356]).

d) *Aircraft Identification*. This field shall indicate the aircraft identification. It shall be represented by the Mode S Aircraft Identification record (see B.2.35).

e) *Aircraft Address*. This field shall contain a unique ICAO Mode S Aircraft Address. When a military aircraft has both a permanent and an operational Mode S Aircraft Address, only one address will be active at a time. The active aircraft address shall be included in this field. This field shall be represented by a 32-bit unsigned integer.

f) *Aircraft Identification Type*. This field shall indicate the aircraft identification type. It shall be represented by an 8-bit enumeration (see [UID 357]).

g) *DAP Source*. This field shall indicate the Downlink of Aircraft Parameters (DAP) source. It shall be represented by the DAP Source record (see B.2.6).

h) *Mode S Altitude*. This field shall provide the Mode S altitude. It shall be represented by the Mode S Altitude record (see B.2.36).

i) *Capability Report*. This field shall contain the Capability Report. The numeric values are from Annex 10 to ICAO Vol. IV, para 3.1.2.5.2.2.1, except for No Statement (see Clause 2). If no specific capability is indicated, this field shall be set to No Statement (255). It shall be represented by an 8-bit enumeration (see [UID 358]).

The format of the Mode S Transponder Basic Data record is shown in Table B.46.

Table B.46—Mode S Transponder Basic Data record

Field size (bits)	Field name	Data type
16	Mode S Status	Mode S Transponder Status record
8	Mode S Levels Present	Mode S Levels Present record
8	Aircraft Present Domain	8-bit enumeration
64	Aircraft Identification	Mode S Aircraft Identification record
32	Aircraft Address	32-bit unsigned integer
8	Aircraft ID Type	8-bit enumeration
8	DAP Source	DAP Source record
16	Mode S Altitude	Mode S Altitude record
8	Capability Report	8-bit enumeration
8	Padding	8 bits unused
16	Padding	16 bits unused
Total Mode S Transponder Basic Data record size = 192 bits		

B.2.42 Mode S Transponder Status record

The Mode S Transponder Status record shall be used to communicate Mode S status information. It shall be represented by a 16-bit record in the Mode S Transponder Basic Data record included in Layer 4 of the IFF PDU. If this Mode S transponder does not have a squitter capability, all fields identified in item a) through item h) below shall be set to zero.

The fields of this record shall be as follows:

a) *Squitter Status*. This 1-bit enumeration field shall indicate whether the squitter is On (1) or Off (0).

b) *Squitter Type*. This 3-bit enumeration field (see [UID 354]) indicates the type of Mode S squitter and shall be set to one of the following values: Not Capable (0), Acquisition (Short) (1), or Extended (2).

c) *Squitter Record Source*. This 1-bit enumeration field (see [UID 355]) shall indicate whether the squitter records are only present as Layer 4 IFF Data Records (0) or are also included as Layer 5 GICB IFF Data Records (1).

d) *Airborne Position Report Indicator*. This 1-bit enumeration field shall indicate whether there is a Squitter Airborne Position Report IFF Data Record either Present (1) or Not Present (0).

e) *Airborne Velocity Report Indicator*. This 1-bit enumeration field shall indicate whether there is a Squitter Airborne Velocity Report IFF Data Record either Present (1) or Not Present (0).

f) *Surface Position Report Indicator*. This 1-bit enumeration field shall indicate whether there is a Squitter Surface Position Report IFF Data Record either Present (1) or Not Present (0).

g) *Identification Report Indicator*. This 1-bit enumeration field shall indicate whether there is a Squitter Identification Report IFF Data Record either Present (1) or Not Present (0).

h) *Event-Driven Report Indicator*. This 1-bit enumeration field shall indicate whether there is a Squitter Event-Driven Report IFF Data Record either Present (1) or Not Present (0).

i) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Mode S transponder is On (1) or Off (0).

j) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Mode S transponder. This field shall be set to either No Damage (0) or Damaged (1).

k) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Mode S transponder. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Mode S Transponder Status record shall be as shown in Table B.47.

Table B.47—Mode S Transponder Status record

Field name	Bits	Value
Squitter Status	0	Enumeration
Squitter Type	1 to 3	Enumeration
Squitter Record Source	4	Enumeration
Airborne Position Report Indicator	5	Enumeration
Airborne Velocity Report Indicator	6	Enumeration
Surface Position Report Indicator	7	Enumeration
Identification Report Indicator	8	Enumeration
Event-Driven Report Indicator	9	Enumeration
Padding	10 to 12	3 bits unused
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Mode S Transponder Status record size = 16 bits		

B.2.43 Modifier record

The Modifier record shall be used to indicate special reply indicators and other status information for a transponder. It shall be included in Data Field 2 of the Fundamental Operational Data record when present in the IFF PDU. See Table B.60 for the list of special reply indicators applicable for a given system type. See B.5.1.2.5 for the issuance and receipt rules associated with this record. This is an 8-bit record. The fields of this record shall be as follows:

a) *Military Emergency*. This 1-bit enumeration field shall indicate whether the transponder has set Military Emergency to On (1) or Off (0).

b) *Identification of Position (I/P)*. This 1-bit enumeration field shall indicate whether the transponder has set the I/P to On (1) or Off (0). This is also referred to as the Special Process Indicator (SPI), "Squawk Flash" or "Squawk Ident." When set in real life, it sends a certain number of special pulses in conjunction with a normal transponder reply that causes a special display to appear at a surveillance system to indicate the location of the aircraft.

c) *Unmanned Aircraft (X)*. This 1-bit enumeration field shall indicate whether the transponder has set the Unmanned Aircraft reply to On (1) or Off (0).

The format of the Modifier record shall be as shown in Table B.48.

Table B.48—Modifier record

Field name	Bits	Value
Not used	0	Unused
Military Emergency	1	Enumeration
Identification of Position (I/P)	2	Enumeration
STI[a] (deprecated)	3	Unused
Unmanned Aircraft (X)	4	Enumeration
Not used	5	Unused
Not used	6	Unused
Not used	7	Unused
Total Modifier record size = 8 bits		

[a]STI was introduced in DIS Version 6, but there is no known use for this field. It is unclear as to the meaning of STI. It could have meant another term for I/P (e.g., Special Transponder Indicator) or could have meant Special Test Instruction. It has been deprecated.

B.2.44 RRB Code record

The RRB Code record shall provide RRB information. When used, this record shall be included as Parameter 1 of the Fundamental Operational Data record. This is a 16-bit record. The following fields shall be included in an RRB Code record:

a) *RRB Code*. This field shall contain the decimal RRB code. See RRB documentation for the range of decimal values. The RRB code shall be included regardless of whether the On/Off Status is set to On (1) or Off (0).

b) *Power Reduction Indicator*. This 1-bit enumeration field shall indicate whether there is a power reduction in effect. This field shall be set to either Off (0) or On (1).

c) *Radar Enhancement Indicator*. This 1-bit enumeration field shall indicate whether there is radar enhancement. This field shall be set to either Off (0) or On (1).
d) *On/Off Status*. This 1-bit enumeration field shall indicate whether the RRB transponder is On (1) or Off (0). The other fields of this record are independent of the On/Off Status field and shall remain unaffected when the status changes.
e) *Damage Status*. This 1-bit enumeration field shall indicate the damage status of the RRB transponder equipment and be set to either No Damage (0) or Damaged (1).
f) *Malfunction Status*. This 1-bit enumeration field shall indicate the malfunction status of the RRB transponder equipment and be set to either No Malfunction (0) or Malfunction (1).

The format of the RRB Code record shall be as shown in Table B.49.

Table B.49—RRB Code record

Field name	Bits	Value
RRB Code	0 to 4	Unsigned integer
Padding	5 to 10	0
Power Reduction Indicator	11	Enumeration
Radar Enhancement Indicator	12	Enumeration
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total RRB Code record size = 16 bits		

B.2.45 Soviet Interrogator Status record

The Soviet Interrogator Status record shall be used to indicate the status of an interrogator to interrogate Soviet transponders.

This record shall be included as Parameter 1 of the Fundamental Operational Data record for an interrogator capable of interrogating Soviet transponders. This is a 16-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Soviet interrogation capability is On (1) or Off (0).
b) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Soviet interrogation capability. This field shall be set to either No Damage (0) or Damaged (1).
c) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Soviet interrogation capability. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Soviet Interrogator Status record shall be as shown in Table B.50.

Table B.50—Soviet Interrogator Status record

Field name	Bits	Value
Padding	0 to 12	13 bits unused
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Soviet Interrogator Status record size = 16 bits		

B.2.46 Soviet Transponder Status record

The Soviet Transponder Status record shall be used to indicate the status of a Soviet transponder.

This record shall be included as Parameter 1 of the Fundamental Operational Data record for a Soviet transponder. This is a 16-bit record. The fields of this record are as follows:

a) *On/Off Status*. This 1-bit enumeration field shall indicate whether the Soviet transponder is On (1) or Off (0).

b) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the Soviet transponder. This field shall be set to either No Damage (0) or Damaged (1).

c) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the Soviet transponder. This field shall be set to either No Malfunction (0) or Malfunction (1).

The format of the Soviet Transponder Status record shall be as shown in Table B.51.

Table B.51—Soviet Transponder Status record

Field name	Bits	Value
Padding	0 to 12	13 bits unused
On/Off Status	13	Enumeration
Damage Status	14	Enumeration
Malfunction Status	15	Enumeration
Total Soviet Transponder Status record size = 16 bits		

B.2.47 Squitter Airborne Position Report IFF Data record

The functional data transmitted in an ICAO Mode S Squitter Airborne Position Report shall be communicated using the Squitter Airborne Position Report IFF Data record. This record is included as an IFF Data record in the Mode S Transponder format of Layer 4 of the IFF PDU. This replicates the functional data contained in a Squitter Airborne Position Report transmitted by a Mode S transponder in accordance with ICAO standards. If one or more data values are included instead of indicating to compute them based on truth data, a simulation shall output this record whenever such data values change. If it is desired to transmit an exact replication of this squitter report, it may be sent using the Layer 5 GICB IFF Data record. Where reference is made to using data values from an appropriate International Civil Aviation Organization (ICAO) technical publication for a specific data field in this record, the specific data and associated numerical assignments from the ICAO publication shall be used (see Clause 2).

The fields and subfields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5003.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Squitter Airborne Position Report*. This field shall specify the appropriate data values or to compute the values using truth data. It shall be represented by a 16-bit record. The subfields shall be as follows:

    1) Aircraft Address Source. This 1-bit enumeration subfield shall indicate the source of the Aircraft Address (see [UID 364]).

    2) Format Type. This 5-bit enumeration subfield shall indicate the Format Type (see [UID 363]).

    3) ICAO Spare. This 2-bit subfield shall represent the spare field that exists in a Squitter Airborne Position Report.

    4) Surveillance/Status. This 2-bit enumeration subfield (see [UID 365]) shall indicate the surveillance status associated with this report and shall be set as follows:

        i) No Information. The status shall be set to No Information (0) if none of the other status values are currently applicable.

        ii) Emergency/Loss of Comm. The status shall be set to Emergency/Loss of Comm. (1) if the Mode 3/A code in the Mode 3/A Code record is currently set to 7500 octal, 7600 octal, or 7700 octal.

        iii) SPI. The status shall be set to SPI (2) if the Modifier record I/P field is set to On (1).

        iv) ATCRB Code Change. The status shall be set to ATCRB Code Change (3) if the Mode 3/A code in the Mode 3/A Code record has changed in the last 10 s.

    5) Turn Rate Source. This 2-bit enumeration subfield (see [UID 366]) shall indicate whether the aircraft is turning or whether the receiving simulation is to make that determination by examining entity truth data as follows:

        i) Compute Locally. If set to Compute Locally (0), it shall mean that the receiving simulation is to calculate which of the other two turn rate values, Less than 1-Degree Turn or Not Turning (1) or 1-Degree or Greater Turn Rate (2), is applicable.

        ii) Less than 1-Degree Turn or Not Turning. If set to Less than 1-Degree Turn or Not Turning (1), then this turn rate value has been provided and does not have to be computed locally.

        iii) 1-Degree or Greater Turn Rate. If set to 1-Degree or Greater Turn Rate (2), then the turn rate value has been provided and does not have to be computed locally.

    6) Altitude Source. This 2-bit enumeration subfield (see [UID 360]) shall be either set to Compute Locally (0) or to Mode 5 Transponder Location IFF Data Record Available (1) that contains the altitude information.

    7) Time Type Source. This 2-bit enumeration subfield (see [UID 367]) shall indicate whether this squitter report was output on an even or odd second or whether the receiving simulation is to determine that based on its Mode S interrogator model:

        i) Compute Locally. If set to Compute Locally (0), it shall mean that the receiving simulation is to calculate which of the other two time type values, Even Second (1) or Odd Second (2), is applicable.

ii) Even Second. If set to Even Second (1), then this time type value has been provided and does not have to be computed locally.

iii) Odd Second. If set to Odd Second (2), then this time type value has been provided and does not have to be computed locally.

8) Lat/Long Source. This 1-bit enumeration subfield (see [UID 360]) shall be either set to Compute Locally (0) or to Mode 5 Transponder Location IFF Data Record Available (1) that contains the Lat/Long information.

The format of the Squitter Airborne Position Report IFF Data record shall be as shown in Table B.52.

Table B.52—Squitter Airborne Position Report IFF Data record

Field size (bits)	Squitter Airborne Position Report fields		Bits
32	Record Type = 5003	32-bit enumeration	—
16	Record Length = 8	16-bit unsigned integer	—
16	Squitter Airborne Position Report	Aircraft Address Source—1-bit enumeration	0
		Format Type—5-bit enumeration	1 to 5
		ICAO Spare—2 bits unused	6 to 7
		Surveillance/Status—2-bit enumeration	8 to 9
		Turn Rate Source—2-bit enumeration	10 to 11
		Altitude Source—1-bit enumeration	12
		Time Type Source—2-bit enumeration	13 to 14
		Lat/Long Source—1-bit enumeration	15
Total Squitter Airborne Position Report IFF Data record size = 64 bits			

B.2.48 Squitter Airborne Velocity Report IFF Data record

The data transmitted in an ICAO Mode S Squitter Airborne Velocity Report shall be communicated using the Squitter Airborne Velocity Report IFF Data record. This record is included as an IFF Data record in the Mode S Transponder format of Layer 4 of the IFF PDU. This replicates the functional data contained in a Squitter Airborne Velocity Report transmitted by a Mode S transponder in accordance with ICAO standards. If one or more data values are included instead of indicating to compute them based on truth data, a simulation shall output this record whenever such data values change. If it is desired to transmit an exact replication of this squitter report, it may be sent using the Layer 5 GICB IFF Data record. Where reference is made to using data values from an appropriate International Civil Aviation Organization (ICAO) technical publication for a specific data field in this record, the specific data and associated numerical assignments from the ICAO publication shall be used (see Clause 2).

The fields and subfields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5004.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Squitter Airborne Velocity Report*. This field shall specify the appropriate data values or to compute the values using truth data. It shall be represented by a 16-bit record. The subfields shall be as follows:

1) Aircraft Address Source. This 1-bit enumeration subfield shall indicate the source of the Aircraft Address (see [UID 364]).

2) Format Type. This 5-bit enumeration subfield shall indicate the Format Type (see [UID 363]).

3) Subtype. This 3-bit enumeration subfield shall indicate the subtype associated with this report. (See the appropriate ICAO publication for the subtype values.)

4) Sync Status. This 1-bit enumeration subfield shall indicate the sync status as being either Synced (1) or Not Synced (0).

5) Ground Track Source. This 1-bit enumeration subfield (see [UID 360]) shall indicate the ground track information as follows:

   i) Compute Locally. If set to Compute Locally (0), it shall mean that the receiving simulation is to calculate ground track data from truth data.

   ii) GICB IFF Data Record Available. If set to GICB IFF Data Record Available (1), then this indicates that a GICB IFF data record is available that contains an ICAO Squitter Airborne Velocity Report with the Ground Track Source information.

6) TRN Rate Source. This 1-bit enumeration subfield (see [UID 360]) shall indicate the Turn (TRN) rate source: whether the receiving simulation is to compute the TRN Rate from truth data or whether a GICB IFF Data record that contains an ICAO Squitter Airborne Velocity Report is available:

   i) Compute Locally. If set to Compute Locally (0), it shall mean that the receiving simulation is to calculate the TRN rate from truth data.

   ii) GICB IFF Data Record Available. If set to GICB IFF Data Record Available (1), then this indicates that a GICB IFF data record is available that contains an ICAO Squitter Airborne Velocity Report with the TRN rate information.

7) VRT Rate Source. This 1-bit enumeration subfield (see [UID 360]) shall indicate the Vertical Rate of Climb (VRT) rate source and shall be set as follows:

   i) Compute Locally. If set to Compute Locally (0), it shall mean that the receiving simulation is to calculate VRT data from truth data.

   ii) GICB IFF Data Record Available. If set to GICB IFF Data Record Available (1), then this indicates that a GICB IFF data record is available that contains an ICAO Squitter Airborne Velocity Report with the VRT rate information.

8) Padding. There is a 3-bit padding field.

The format of the Squitter Airborne Velocity Report IFF Data record shall be as shown in Table B.53.

Table B.53—Squitter Airborne Velocity Report IFF Data record

Field size (bits)	Squitter Airborne Velocity Report fields		Bits
32	Record Type = 5004	32-bit enumeration	—
16	Record Length = 8	16-bit unsigned integer	—

Table B.53—Squitter Airborne Velocity Report IFF Data record *(continued)*

Field size (bits)	Squitter Airborne Velocity Report fields		Bits
16	Squitter Airborne Velocity Report	Aircraft Address Source—1-bit enumeration	0
		Format Type—5-bit enumeration	1 to 5
		Subtype—3-bit enumeration	6 to 8
		Sync Status—1-bit enumeration	9
		Ground Track Source—1-bit enumeration	10
		TRN Rate Source—1-bit enumeration	11
		VRT Rate Source—1-bit enumeration	12
		Padding—3 bits unused	13 to 15
Total Squitter Airborne Velocity Report IFF Data record size = 96 bits			

B.2.49 Squitter Event-Driven Report IFF Data records

The functional data transmitted in an ICAO Mode S Squitter Event-Driven Report shall be communicated using the Squitter Event-Driven Report IFF Data record. This record is included as an IFF Data record in the Mode S Transponder format of Layer 4 of the IFF PDU. This replicates the functional data contained in a Squitter Event-Driven Report transmitted by a Mode S transponder in accordance with ICAO standards. If one or more data values are included instead of indicating to compute them based on truth data, a simulation shall output this record whenever such data values change. If it is desired to transmit an exact replication of this squitter report, it may be sent using the Layer 5 GICB IFF Data record. Where reference is made to using data values from an appropriate International Civil Aviation Organization (ICAO) technical publication for a specific data field in this record, the specific data and associated numerical assignments from the ICAO publication shall be used (see Clause 2).

The fields and subfields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5008.
b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)
c) *Squitter Event Identification*. This field shall indicate the ICAO Downlink Format (DF) number assigned to this type of event data. It shall be represented by a 16-bit enumeration.
d) *Squitter Event Data fields*. These are variable length fields to represent the specific event data being sent.

The format of the Squitter Event-Driven Report IFF Data record shall be as shown in Table B.54.

Table B.54—Squitter Event-Driven Report IFF Data record

Field size (bits)	Field name	Data type
32	Record Type = 5008	32-bit enumeration
16	Record Length	16-bit unsigned integer (6 + K + P)
16	Squitter Event Identification	16-bit enumeration
$8S_1$	Squitter Event Data Field #*1*	Squitter Event Data Field—S_1 octets
	• • •	
$8S_N$	Squitter Event Data Field #*N*	Squitter Event Data Field—S_N octets
$8P$	Padding	Padding to 32-bit boundary—P octets
Total Squitter Event-Driven Report IFF Data record size = 48 +8K + 8P bits where K is the length of the system specific data in octets, which is $2 + \sum_{i=1}^{N} S_i$ S_i is the length in octets of Squitter Event Data Field i P is the number of padding octets, which is $\lceil K/4 \rceil 4 - K$ $\lceil x \rceil$ is the largest integer $< x + 1$.		

B.2.50 Squitter Identification Report IFF Data record

The data transmitted in an ICAO Mode S Squitter Identification Report shall be communicated using the Squitter Identification Report IFF Data record. This record is included as an IFF Data record in the Mode S Transponder format of Layer 4 of the IFF PDU. This replicates the functional data contained in a Squitter Identification Report transmitted by a Mode S transponder in accordance with ICAO standards. If one or more data values are included instead of indicating to compute them based on truth data, a simulation shall output this record whenever such data values change. If it is desired to transmit an exact replication of this squitter report, it may be sent using the Layer 5 GICB IFF Data record. Where reference is made to using data values from an appropriate International Civil Aviation Organization (ICAO) technical publication for a specific data field in this record, the specific data and associated numerical assignments from the ICAO publication shall be used (see Clause 2).

The fields and subfields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5006.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Squitter Identification Report*. This field shall specify the appropriate data values or to compute the values using truth data. It shall be represented by a 16-bit record. The subfields shall be as follows:

1) Aircraft Address Source. This 1-bit enumeration subfield shall indicate the source of the Aircraft Address (see [UID 364]).
2) Format Type. This 5-bit enumeration subfield shall indicate the Format Type (see [UID 363]).
3) Aircraft Type/Wake. This 3-bit enumeration subfield shall provide aircraft type and wake data (see [UID 368]). It shall be set to No Statement (0) if no aircraft type or wake information is available.
4) Aircraft ID Source. This 1-bit enumeration subfield shall indicate the data source for the Aircraft ID and whether a GICB IFF Data record that contains an ICAO Squitter Identification Report is available.
5) ICAO Spare. This 6-bit subfield represents the ICAO Spare field contained in a Squitter Identification Report.

The format of the Squitter Identification Report IFF Data record shall be as shown in Table B.55.

Table B.55—Squitter Identification Report IFF Data record

Field size (bits)	Squitter Identification Report fields		Bits
32	Record Type = 5006	32-bit enumeration	—
16	Record Length = 8	16-bit unsigned integer	—
16	Squitter Identification Report	Aircraft Address Source—1-bit enumeration	0
		Format Type—5-bit enumeration	1 to 5
		Aircraft Type/Wake—3-bit enumeration	6 to 8
		Aircraft ID Source—1-bit enumeration	9
		ICAO Spare—6 bits unused	10 to 15
Total Squitter Identification Report IFF Data record size = 64 bits			

B.2.51 Squitter Surface Position Report IFF Data record

The data transmitted in a ICAO Mode S Squitter Surface Position Report shall be communicated using the Squitter Surface Position Report IFF Data record. This record is included as an IFF Data record in the Mode S Transponder format of Layer 4 of the IFF PDU. This report is transmitted when an aircraft is on the ground or aboard ship. This replicates the functional data contained in an Squitter Surface Position Report transmitted by a Mode S transponder in accordance with ICAO standards. If one or more data values are included instead of indicating to compute them based on truth data, a simulation shall output this record whenever such data values change. If it is desired to transmit an exact replication of this squitter report, it may be sent using the Layer 5 GICB IFF Data record. Where reference is made to using data values from an appropriate International Civil Aviation Organization (ICAO) technical publication for a specific data field in this record, the specific data and associated numerical assignments from the ICAO publication shall be used (see Clause 2).

The fields and subfields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5005.
b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Squitter Surface Position Report*. This field shall specify the appropriate data values or to compute the values using truth data. It shall be represented by a 16-bit record. The subfields shall be as follows:
   1) Aircraft Address Source. This 1-bit enumeration subfield shall indicate the source of the Aircraft Address (see [UID 364]).
   2) Format Type. This 5-bit enumeration subfield shall indicate the Format Type (see [UID 363]).
   3) Movement. This 3-bit enumeration subfield shall provide movement data.
   4) Heading Source. This 1-bit enumeration subfield (see [UID 360]) shall be set as follows:
      i) Compute Locally. If set to Compute Locally (0), it shall mean that the receiving simulation is to calculate heading data from truth data.
      ii) GICB IFF Data Record Available. If set to GICB IFF Data Record Available (1), then this indicates that a GICB IFF data record is available that contains an ICAO Squitter Surface Position Velocity Report with the heading information.
   5) VRT Rate Source. This 1-bit enumeration subfield (see [UID 360]) shall be set as follows:
      i) Compute Locally. If set to Compute Locally (0), it shall mean that the receiving simulation is to calculate VRT data from truth data.
      ii) GICB IFF Data Record Available. If set to GICB IFF Data Record Available (1), then this indicates that a GICB IFF data record is available that contains an ICAO Squitter Surface Position Velocity Report with the VRT Rate information.

d) *ICAO Spare*. This field represents the ICAO Spare field in the actual report. It shall be represented by a 32-bit unsigned integer:
   1) ICAO Spare. This 24-bit subfield represents the ICAO Spare field contained in a Squitter Surface Position Report.
   2) Padding. This 8-bit subfield is a padding field to meet Distributed Interactive Simulation (DIS) alignment requirements.

The format of the Squitter Surface Position Report IFF Data record shall be as shown in Table B.56.

Table B.56—Squitter Surface Position Report IFF Data record

Field size (bits)	Squitter Surface Position Report fields		Bits
32	Record Type = 5005	32-bit enumeration	—
16	Record Length = 12	16-bit unsigned integer	—
16	Squitter Surface Velocity Report	Aircraft Address Source—1-bit enumeration	0
		Format Type—5-bit enumeration	1 to 5
		Movement—3-bit enumeration	6 to 8
		Heading Source—1-bit enumeration	9
		VRT Rate Source—1-bit enumeration	10
		Padding—5 bits unused	11 to 15
32	ICAO Spare	ICAO Spare—24 bits unused	0 to 23
		Padding—8 bits unused	24 to 31
Total Squitter Surface Position Report IFF Data record size = 96 bits			

B.2.52 System Status record

The System Status record shall be used to indicate the overall on/off status of the system type, the operational status, and the capability status of components listed in parameters 1 through 6 of the Fundamental Operational Data record. This is an 8-bit record. The fields of this record shall be as follows:

a) *System On/Off Status*. This 1-bit enumeration field shall indicate whether the overall system is On (1) or Off (0).

b) *Parameter 1 Capable*. This 1-bit enumeration field shall indicate whether there is a system component defined for Parameter 1 of the Fundamental Operational Data record. If there is, this field shall be set to Capable (0). Otherwise, the field shall be set to Not Capable (1).

c) *Parameter 2 Capable*. This 1-bit enumeration field shall indicate whether there is a system component defined for Parameter 2 of the Fundamental Operational Data record. If there is, this field shall be set to Capable (0). Otherwise, the field shall be set to Not Capable (1).

d) *Parameter 3 Capable*. This 1-bit enumeration field shall indicate whether there is a system component defined for Parameter 3 of the Fundamental Operational Data record. If there is, this field shall be set to Capable (0). Otherwise, the field shall be set to Not Capable (1).

e) *Parameter 4 Capable*. This 1-bit enumeration field shall indicate whether there is a system component defined for Parameter 4 of the Fundamental Operational Data record. If there is, this field shall be set to Capable (0). Otherwise, the field shall be set to Not Capable (1).

f) *Parameter 5 Capable*. This 1-bit enumeration field shall indicate whether there is a system component defined for Parameter 5 of the Fundamental Operational Data record. If there is, this field shall be set to Capable (0). Otherwise, the field shall be set to Not Capable (1).

g) *Parameter 6 Capable*. This 1-bit enumeration field shall indicate whether there is a system component defined for Parameter 6 of the Fundamental Operational Data record. If there is, this field shall be set to Capable (0). Otherwise, the field shall be set to Not Capable (1).

h) *Operational Status*. This 1-bit enumeration field shall indicate whether the system is operational or whether it has experienced a system failure. This status is independent of the System On/Off Status field. This status shall be set to System Failed (1) if no Parameter 1 to 6 or Modifier record data can be provided. Otherwise, the status shall be set to Operational (0).

The format of the System Status record shall be as shown in Table B.57.

Table B.57—System Status record

Field name	Bits	Value
System On/Off Status	0	Enumeration
Parameter 1 Capable	1	Enumeration
Parameter 2 Capable	2	Enumeration
Parameter 3 Capable	3	Enumeration
Parameter 4 Capable	4	Enumeration
Parameter 5 Capable	5	Enumeration
Parameter 6 Capable	6	Enumeration
Operational Status	7	Enumeration
Total System Status record size = 8 bits		

B.2.53 TCAS/ACAS Status record

The TCAS/ACAS Status record shall be used to specify the status of a TCAS or an ACAS system. These are essentially the same system with the TCAS system used in the United States and the ACAS used in Europe. Other countries of the world have adopted either one or the other system. A TCAS/ACAS system may interrogate Mode A/C and Mode S transponders.

The TCAS/ACAS Status record shall be contained in Parameter 6 of the Fundamental Operational Data record when an interrogator or transponder system type has such a capability. In this case, the Parameter 6 Capable field of the System Status record of the Fundamental Operational Data record shall be set to Capable (0).

See Table B.58 for the fields required to be implemented. If a field is not required to be implemented, it shall be set to zero. This record shall be represented by a 16-bit record. The fields of this record shall be as follows:

a) *Basic/Advanced Indicator*. This 1-bit enumeration field (see [UID 341]) shall indicate whether this TCACS/ACAS Status record contains Basic (0) or Advanced (1) data. Basic data is the minimum required to be set by any TCAS/ACAS capable interrogator or transponder. Advanced data means that the record contains additional data fields for a higher fidelity simulation of TCAS/ACAS capabilities.

b) *TCAS/ACAS Indicator*. This 1-bit enumeration field shall indicate whether the record applies to a TCAS (0) or an ACAS (1) system (see [UID 342]).

c) *Software Version*. This 3-bit enumeration field shall indicate the software version of the TCAS/ACAS system if known or set to No Statement (0) if not known (see [UID 343]).

d) *TCAS/ACAS Type*. This 3-bit enumeration field shall indicate the type of TCAS/ACAS system other than a TCAS I or II system, if known, or shall be set to No Statement (0) if not known (see [UID 344]).

e) *TCAS I/II Type*. This 1-bit enumeration field shall indicate whether a TCAS system is a TCAS I (0) or TCAS II (1) system when the TCAS/ACAS Type field does not indicate a TCAS III system (see [UID 345]). If the TCAS/ACAS Type field contains a nonzero value, then this field shall be set to No Statement (0).

f) *On/Off Status*. This 1-bit enumeration field shall indicate whether the TCAS/ACAS system is On (1) or Off (0).

g) *Damage Status*. This 1-bit enumeration field shall indicate whether there is damage to the TCAS/ACAS system. This field shall be set to either No Damage (0) or Damaged (1).

h) *Malfunction Status*. This 1-bit enumeration field shall indicate whether there is a malfunction of the TCAS/ACAS system. This field shall be set to either No Malfunction (0) or Malfunction (1).

NOTE—A TCAS record was defined for Parameter 6 and documented in SISO-REF-010 after Version 6 of the standard was published. This has necessitated retaining the fields defined for this basic TCAS record, which occupied bits 12 to 15 and then adding the additional fields in bits 1 to 11 to support advanced TCAS/ACAS simulations.

The format and field implementation requirements (required or not applicable) for the TCAS/ACAS Status record shall be as shown in Table B.58.

Table B.58—TCAS/ACAS Status record

Field name	Bits	Basic	Advanced
Basic/Advanced Indicator	0	R	R
TCAS/ACAS Indicator	1	NA	R
Software Version	2 to 4	NA	R
Padding	5 to 8	NA	NA
TCAS/ACAS Type	9 to 11	NA	R
TCAS I/II Type	12	R	R
On/Off Status	13	R	R
Damage Status	14	R	R
Malfunction Status	15	R	R
Total TCAS/ACAS Status record size = 16 bits			
Legend			
R—Required; NA—Not Applicable.			

B.2.54 Transponder Location Error IFF Data record

The specification of errors in location and altitude information transmitted by a Mode 5 or Mode S transponder shall be communicated using the Transponder Location Error IFF Data record. Errors are defined as delta values from the truth location of the entity platform. They reflect either human or navigation system errors that might be present and manifest themselves in the data sent in a Mode 5 or Mode S reply or squitter report. This shall be an optional record for issuance as there is no requirement within this standard to model errors in Mode 5 or Mode S transponder location information. If Mode 5 or S location errors are modeled, then this record shall be issued to convey the magnitude of the error for one or more of the location values that could have an error. Fields with no errors shall have all bits of that field set to zero.

All simulations having a Mode 5, Mode S, or Mode 5/S interrogator model shall process this record when received in a Mode 5 or Mode S transponder IFF PDU and shall apply any included errors when using entity truth data to derive Mode 5 or S latitude, longitude, or altitude information.

The fields of this record shall be as follows:

a) *Record Type*. This field shall specify the identification of the record and shall be represented by a 32-bit enumeration. It shall be set to a value of 5002.

b) *Record Length*. This field shall indicate the record length expressed as the total number of octets in the record. It shall be represented by a 16-bit unsigned integer. (See 6.2.43.)

c) *Delta Latitude*. This field shall specify the delta error in latitude from the true latitude computed for the entity with a Mode 5 or S transponder. Delta latitude shall be represented as two’s complement coded latitude with the Most Significant Bit (MSB) equal to –90 degrees. All bits of this field shall be set to zero if there is no error in latitude.

d) *Delta Longitude*. This field shall specify the delta error in longitude from the true longitude computed for the entity. Delta longitude shall be represented as two's complement coded longitude with the Most Significant Bit (MSB) equal to –180 degrees. All bits of this field shall be set to zero if there is no error in longitude.

e) *Delta Mode 5 Altitude*. This field shall specify the delta error in Mode 5 Altitude of the entity with a Mode 5 or S transponder. It shall be represented by a Delta Mode 5 Altitude record (see B.2.8). All bits of this field shall be set to zero if there is no error in Mode 5 altitude.

f) *Delta Barometric Altitude*. This field shall specify the delta error in barometric altitude of the entity. It shall be represented by a Delta Barometric Altitude record (see B.2.7). All bits of this field shall be set to zero if there is no error in barometric altitude.

The format of the Transponder Location Error IFF Data record shall be as shown in Table B.59.

Table B.59—Transponder Location Error IFF Data record

Field size (bits)	Field name	Value
32	Record Type = 5002	32-bit enumeration
16	Record Length = 20	16-bit unsigned integer
16	Padding	16 bits unused
32	Delta Latitude	32-bit two's complement integer
32	Delta Longitude	32-bit two's complement integer
16	Delta Mode 5 Altitude	Delta Mode 5 Altitude record
16	Delta Barometric Altitude	Delta Barometric Altitude record
Total Transponder Location Error IFF Data record size = 160 bits		

B.3 Operation

B.3.1 General

A basic IFF system consists of an interrogator subsystem and a transponder subsystem. The interrogator transmits challenges (also called interrogations) on a specific radio frequency. When a transponder receives an interrogation, it transmits a response on a different radio frequency. Some transponders send out information without being interrogated. Military IFF systems and the civilian ATCRB system (ATCRBS) are mainly carried by aircraft, ships, and ground facilities. Most aircraft only have transponders. However, some aircraft, such as Airborne Warning and Control System (AWACS), and some fighters and Naval ships have both transponders and interrogators.

Mode S is unique in that it operates basically in two modes—All-Call and Roll-Call. An aircraft is first acquired by responding to an All-Call interrogations, and thereafter, it will respond to interrogations from a specific Mode S interrogator. (Unlike civilian and military ATCRBS and Mark X/XII interrogators, which cannot be uniquely identified, Mode S interrogators are assigned 2-digit identifiers.)

Table B.60 indicates the specific system types and associated capabilities for interrogator and transponder systems that are currently defined when this version of the standard was issued. A system shall implement the special replies and mode types as shown in Table B.60. As new systems are required to be represented in a distributed simulation environment, they will be added to this annex along with any system-specific information content, issuance, and receipt requirements.

Table B.60—System types and modes

System type No.	System type name	MHz	Special reply			Mode types	Section	Notes
			IP	ME	UA			
1	Mark X/XII/ATCRBS Transponder	1090	—	—	—		B.5.2	
	Mark X	1090	X	X	—	1, 2, 3/A,C	B.5.2.2	6
	Mark XII	1090	X	X	X	1, 2, 3/A, 4, C, S	B.5.2.3	5
	ATCRBS	1090	X		X	3/A, C, S	B.5.2.4	4
2	Mark X/XII/ATCRBS Interrogator	1030	—	—	—	1, 2, 3/A, 4, C	B.5.3	
3	Soviet Transponder	Unk	—	—	—	Not specified*	B.5.4	1
4	Soviet Interrogator	Unk	—	—	—	Not specified*	B.5.5	1
5	RRB Transponder	Unk	—	—	—	RRB Code	B.5.6	2
6	Mark XIIA Interrogator	1030	—	—	—	1, E1, 2, 3/A , 4, C, 5, MS	B.5.7	3
7	Mode 5 Interrogator	1030	—	—	—	1, E1, 2, 3/A , 4, C, 5, MS	B.5.8	3
8	Mode S Interrogator	1030	—	—	—	3/A, C, S	B.5.9	3
9	Mark XIIA Transponder	1090	X	X	X	1, 2, 3/A, 4, C, 5, MS	B.5.10	3
10	Mode 5 Transponder	1090	X	X	X	5	B.5.11	3
11	Mode S Transponder	1090	X	—	—	3/A, C, S	B.5.12	3
12	Mark XIIA Combined Interrogator/ Transponder (CIT)	1030/ 1090	X	X	X	1, E1, 2, 3/A , 4, C, 5, MS	B.5.13	3
13	Mark XII Combined Interrogator/ Transponder (CIT)	1030/ 1090	X	X	X	1, E1, 2, 3/A , 4, C, MS	B.5.14	3
14	TCAS/ACAS Transceiver	1030/ 1090	—	—	—		B.5.15	7

Table B.60—System types and modes *(continued)*

System type No.	System type name	MHz	Special reply			Mode types	Section	Notes
			IP	ME	UA			

Notation:

X Required Implementation
IP Identification of Position (also referred to as requests to "Squawk Flash" or "Squawk Ident")
ME Military Emergency
UA Unmanned Aircraft
MS Military Mode S only implements a subset of Mode S capabilities

NOTE 1—Not specified*: There are no specific requirements related to Soviet modes and codes, although the IFF PDU fully supports such an implementation.

NOTE 2—An RRB transponder is not interrogated by an RRB interrogator but by radars. More details are provided with the RRB system type requirements.

NOTE 3—E1 refers to Enhanced Mode 1, which consists of 4096 codes. MS refers to Military Mode S because it only implements a subset of Mode S capabilities.

NOTE 4—Not all ATCRBS transponders have a Mode C and Mode S capability. The IFF PDU Layer 1, Fundamental Operational Data record will indicate whether these modes are present.

NOTE 5—Some Mark XII transponders have a Mode S capability. All Mark XIIA transponders have a Mode S capability. These military transponders do not necessarily implement all Mode S features.

NOTE 6—Mark X transponders have three subtypes and therefore do not have the same Mode 1, 2, and 3 capabilities. See the records for those modes and the System Type 1 requirements for specific capabilities.

NOTE 7—TCAS/ACAS. This system type is only used when a simulation needs to report TCAS/ACAS information in a separate IFF PDU to meet its fidelity requirements.

B.3.2 Transponder/Interrogator Interaction Overview

A brief overview is provided to explain how this standard supports simulating interactions between transponders and interrogators especially regarding data reports. The terms *messages*, *reports*, *data reports*, *message reports*, *message formats,* and *tactical data report formats* are all commonly used when discussing IFF interactions. The term *report* will be used in this subclause to mean any IFF digital message either broadcast by a transponder or an interrogator, or transmitted or received as part of an uplink/downlink protocol such as civilian Mode S, ADS-B, or military Mode 5. The basic interaction between simulated interrogators and transponders is explained. Two simulation techniques are provided for transponder/interrogator representation. They are called the Regeneration Mode and the Interactive Mode (see 5.7.6.5.1):

a) *Regeneration Mode*. Certain types of transponders (e.g., Mode 5 and Mode S transponders) have the capability to transmit specific reports. Each report has a unique identifier:

   1) The Regeneration Concept. In the Regeneration Mode, interrogator simulations do not interrogate entities except for special conditions. They are, however, required to transmit an *interrogator* IFF PDU initially, upon changes in their status and as heartbeats (see 4.2.6). The same is true for transponders issuing *transponder* IFF PDUs. A *transponder* IFF PDU reflects the data that a transponder would provide in reply to any type of interrogation it would respond to, or would provide in a squitter. An interrogator model determines which transponder modes and reports it would be requesting in a specific interrogation and then retrieves the associated transponder data it would receive in a reply from its local database. This substantially reduces network and computer processing requirements as there may be hundreds of transponder- and interrogator-equipped entities present in a distributed simulation environment.

2) Continuously Updating Report Data Fields. A report may contain both continuously updating data fields (e.g., position and altitude) in addition to data that does not change (e.g., aircraft address) or changes infrequently. To support the Regeneration Mode, a report contained in this standard will indicate for a continuously updating field whether an optional IFF Data record is included to provide the associated real-time data or if entity truth data should be used to compute the value.

3) Report Formats. A report is represented in an IFF PDU using one of three methods. This standard provides support for additional composite and emulated reports by creating IFF Data records for them:

   i) Replicated Reports. The report is replicated by using the data fields of the report but not necessarily with the same field sizes. Field content may be represented by options to replicate the data. For example, an Altitude field may have an indicator that specifies whether the receiving model should compute data locally (using entity truth data) or that a separate record is included that contains the actual altitude value. Other fields may contain the same enumerations as the actual report.

   ii) Composite Reports. This is the case for military reports whose formats are not publicly available. For example, a record is provided in a Mode 5 *transponder* IFF PDU that identifies which reports are capable of being transmitted and data fields for the identified report types are contained in various records included in the PDU. A receiving simulation, such as a Mode 5 interrogator model, uses the data records to reconstruct the content of each report. See AIMS 03-100A (Clause 2) for more information regarding military Mode 5 reports.

   iii) Emulated Reports. An emulated report is a report that contains all of the data of an actual report, although the field sizes may differ due to data representation rules associated with the DIS protocol. If the report contains *continuously updating* fields, the content of those fields will represent the actual real-time data values. In this case, the IFF PDU needs to be output at a rate that will allow receiving simulations that process emulated reports to be able to use the data in a realistic manner. For example, if emulated reports are processed by a simulation gateway interfaced to an actual fielded system (e.g., an Aegis ship or Patriot unit), the data must appear realistic to the operational programs and operators.

4) Active Interrogations. Support is provided for an interrogator to send an IFF PDU that represents an *active interrogation*. An active interrogation is used when there is a need to convey a specific Mode 5 interrogation request that would cause a visual or aural indication at the receiving Mode 5 transponder due to its importance (see B.5.1.2.10).

5) Enumerations. Whenever possible, the enumerations associated with a report are included in this standard or in SISO-REF-010 (Clause 2). Otherwise ICAO or other government publications will need to be referenced.

b) *Interactive Mode*. When operating in the Interactive Mode, transponders and interrogators emulate actual interrogations and replies between a specific transponder and a specific interrogator (see 5.7.6.5.2). A separate set of requirements apply to the Interactive Mode although both the Interactive Mode and the Regeneration Mode use many of the same records and data fields.

B.4 Layer transmit/receive requirements

The required and optional information layers to transmit for each transponder and interrogator system type shall be as specified in Table B.61.

Table B.61—IFF PDU transmit requirements

System type	Definition	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
1	Mark X/XII/ATCRBS Transponder	R	O	NA	NA	NA
2	Mark X/XII/ATCRBS Interrogator	R	O	NA	NA	NA
3	Soviet Transponder	R	O	NA	NA	NA
4	Soviet Interrogator	R	O	NA	NA	NA
5	RRB Transponder	R	O	NA	NA	NA
6	Mark XIIA Interrogator	R	O	R	R	O
7	Mode 5 Interrogator	R	O	R	NA	NA
8	Mode S Interrogator	R	O	NA	R	O
9	Mark XIIA Transponder	R	O	R	R	O
10	Mode 5 Transponder	R	O	R	NA	NA
11	Mode S Transponder	R	O	NA	R	O
12	Mark XIIA CIT	R	O	R	R	O
13	Mark XII CIT	R	O	NA	NA	O
14	TCAS/ACAS Transceiver	R	O	NA	NA	O
Legend R Required Transmit NA Not Applicable O Optional Transmit CIT Combined Interrogator/Transponder						

The system types and information layers that each IFF interrogator model is required to receive and process shall be as specified in Table B.62 when operating in the Regeneration Mode. In this mode, transponder models have no requirement to process any received transponder or interrogator IFF PDUs and interrogator models have no requirement to process any received interrogator IFF PDUs in order to be in compliance with this standard. The requirements for the Interactive Mode are not included in this table. Nothing in this standard shall preclude a simulation transponder or interrogator model from processing any IFF PDU to support other requirements. Two examples include (1) to implement an interactive exchange of interrogations and replies to support high-fidelity simulations, or (2) to model interference when there are many interrogators and transponders in the same geographic area.

Table B.62—IFF PDU receipt requirements

Local simulation interrogator model		Received transponder IFF PDU		
System type	**System name**	**System type**	**System name**	**Minimum layers to process**
2	Mark X/XII/ATCRBS/Mode S Interrogator			
	Any Mark X Interrogator	1	Mark X (A) Mark XII	L1 L1
	Any Mark X (SIF) or (A) Interrogator	1	Mark X (A) Mark XII	None
	Any Mark XII Interrogator	1	Mark X Mark X (SIF) Mark X (A) ATCRBS Mode S Mark XIIA	L1 L1 L1 L1 L1 L1
	Any ATCRBS Interrogator	1	Mark X (SIF) Mark X (A) ATCRBS Mode S Mark XIIA	L1 L1 L1 L1 L1
4	Soviet Interrogator	3	Any	L1
6	Mark XIIA Interrogator	1, 9, 10, 11, 13	Mode S Mark X Mark XII ATCRBS Mode 5/S	L1, L4 L1 L1 L1 L1, L3, L4
7	Mode 5 Interrogator	9, 10	Mode 5/S Mode 5	L1, L3 L1, L3
8	Mode S Interrogator	9 5	Mode 5/S Mode S	L1, L4 L1, L4
12	Mark XIIA CIT (interrogator)	1,5, 9	Mark X, XII Mode 5/S Mode S	L1, L3, L4 L1, L3, L4 L1, L4
13	Mark XII CIT (interrogator)	1,5, 9	Mark X, XII Mode 5/S Mode S	L1, L3, L4 L1, L3, L4 L1, L4

B.5 System types

B.5.1 General

B.5.1.1 Introduction

This subclause lists the system-specific information content, issuance, and receipt rules that shall be adhered to in addition to those specified in 5.7.6 and 7.6.5 for specific military and civilian interrogator and transponder systems, including TCAS systems. Where a conflict exists between the requirements specified in those paragraphs and the paragraphs contained herein, the requirements specified herein shall take precedence. Every system type for which a system type enumeration exists has a paragraph in this subclause to indicate whether there are unique requirements for it.

The following notations are used:

a) *Required (R)*. This layer, field, or enumerated value within a field, shall be implemented.

b) *Conditionally Required (CR)*. This layer, field, or record shall be included for a system type if the manufacturer of the IFF System has included a nonoptional, additional capability that is not required by government or industry standards for the system type. This notation is indicated in the layer requirements table for a system type when applicable for an IFF PDU layer, as a whole, and in the Information Content subparagraphs for each system type where detailed inclusion requirements for layers, and for fields and records within a layer, are specified:

  1) Layer 1 CR Records. The TCAS/ACAS Status record shall be included if the entity has a TCAS or ACAS system or subsystem (see B.2.53) regardless of whether the System Type = TCAS/ACAS Transceiver (14).

  2) Layer 3. This layer shall be included if the IFF System includes a Mode 5 capability.

  3) Layer 4. This layer shall be included if the IFF System includes a Mode S capability.

  4) Layer 5. This layer shall be included if either one, or both, of the following conditions exist:

    i) The IFF System is transmitting associated IFF digital messages that emulate the format or transmission frequency of such messages in actual operations.

    ii) The IFF System is operating in the Interactive Mode.

c) *Optional (O)*. This layer, field, or enumerated field value shall be optional to implement.

d) *Not Defined*. This layer is not defined for any system type.

e) *Not Applicable (NA)*. This layer, field, or enumerated field value is not applicable to this system type, and there is no case (including a record for a field) where it should be noted otherwise. For example, if a layer is *Conditionally Required* or a field is *Optional*, those designations would be used instead of *Not Applicable*.

f) *Available*. This field is available to be used by a system type provided any data or record already specified for this field is not applicable to the system type.

If a field is not implemented, then all bits of that field shall be set to zero.

B.5.1.2 Special issuance and receipt rules

B.5.1.2.1 Introduction

The subclause lists issuance and receipt rules that apply to specific data associated with two or more system types but that do not apply to all system types. This is done to avoid the repetition of such rules for each system type.

B.5.1.2.2 Mode 4 issuance and receipt rules

An interrogator or a transponder simulation that transmits or receives and processes Mode 4 for a system type shall adhere to the following Regeneration and Interactive Issuance and Receipt requirements:

a) *Regeneration Mode—Issuance Rules*
   1) Change/Options fields. The Alternate Mode 4 Indicator (B.2.4) shall always be set to Present (1) if this transponder or interrogator is capable of transmitting Mode 4 information regardless of whether or not a Mode 4 pseudo crypto code is contained in the Mode 4 Code record. Other fields shall be set as appropriate.
   2) Data Field 1. Alternate Mode 4 Challenge/Reply field (see B.2.1.3.2). If a system type is required to implement this field, the following issuance rules shall apply:
      i) Transponder/Interrogator. A value of No Statement (0) shall be set if this interrogator or transponder system type does not have a Mode 4 capability. Otherwise, this field shall be set as shown in item ii) and item iii) below.
      ii) Transponder. Valid (1), Invalid (2), or No Response (3). The Unable to Verify (4) reply is optional. (See B.2.1.3.2.) The Mode 4 reply value for an IFF system is set based on the planned interaction between a system's Mode 4 transponder and any received Mode 4 interrogations to support DIS exercise objectives.
      iii) Interrogator. Valid (1) or Invalid (2).
   3) Fundamental Operational Data Record, Parameter 4, Mode 4 Code Record. Any transponder or interrogator simulation that transmits Mode 4 for a system type either shall set the Mode 4 Code field to a simulated Mode 4 pseudo crypto code value or indicates that only an Alternate Mode 4 Challenge/Reply value is present.
   4) Mode 4 Pseudo Crypto Code. The inclusion of a Mode 4 pseudo crypto code is not a requirement of this standard. However, if a pseudo crypto code is implemented by an interrogator or a transponder, it shall adhere to the following issuance rules:
      i) If a Mode 4 pseudo crypto code is being transmitted, a simulation shall be capable of making changes to the pseudo crypto code, as required, either automatically or manually during a simulation exercise.
      ii) Whenever the Mode 4 Code record contains a pseudo crypto code, the Alternate Mode 4 Challenge/Reply field in the IFF PDU shall be set to the corresponding matching value.
      iii) If the Crypto Control IFF Data record is included in an IFF PDU, and it has an active Mode 4 Pseudo Crypto Code, the value of that field shall match the pseudo crypto code value contained in the Mode 4 Code.
      iv) The use of actual Mode 4 pseudo crypto code requires pre-exercise coordination so that the appropriate pseudo crypto code value is set for each IFF system that has a Mode 4 pseudo crypto code capability. The Alternate Mode 4 Challenge/Reply and Mode 4 pseudo crypto code correlation schema will be provided by an appropriate military organization and is beyond the scope of this standard.

   NOTE—Care needs to be taken in discussing Mode 4 as some of the implementation details are classified.

b) *Regeneration Mode—Receipt Rules*
   1) A transponder simulation in the Regeneration Mode is not required to process a received interrogator or transponder IFF PDU.
   2) An interrogator simulation that receives and processes Mode 4 data shall store the Alternate Mode 4 Challenge/Reply value received in an IFF PDU for a remote entity even though a received IFF PDU may also contain a Mode 4 pseudo crypto code. The Mode 4 pseudo crypto code may also be stored.
   3) An interrogator simulation may use either the Alternate Mode 4 Challenge/Reply or the Mode 4 pseudo crypto code to determine the Mode 4 reply status. However, if the Alternate Mode 4

Challenge/Reply value is No Response, then that shall be used as the Mode reply status regardless of any Mode 4 Pseudo Code comparison results.

4) A Mode 4 interrogator model will interpret the received reply based on its own internal processes to derive an appropriate reply value.

c) *Interactive Mode—Issuance Rules*

1) Interactive Mode 4 Interrogation. An interrogator that issues a Mode 4 interrogation (challenge) in the Interactive Mode shall adhere to the following requirements:

i) The Interactive Mode requirements specified in 5.7.6.5 shall apply.

ii) Layer 1 shall include the present regeneration data field values associated with the *regeneration* IFF PDU for this interrogator entity, including Mode 4 and Alternate Mode 4 Challenge/Reply field values, except for the following fields of the Change/Options record:

— The Change Indicator shall be set to Not Initial Report/No Change (0).
— The Heartbeat Indicator shall be set to Not a Heartbeat (0).
— The Simulation Mode shall be set to Interactive (1).
— The Interactive Capable field shall be set to Capable (1).

iii) Layer 2 may be included if implemented and deemed appropriate.

iv) Layer 5 shall be included and contain, as a minimum, a Basic Interactive IFF Data record. The fields of this record shall be set as appropriate.

2) Interactive Mode 4 Reply. An *interactive* IFF PDU reply shall contain the following layers and Mode 4 data:

i) The Interactive requirements specified in 5.7.6.5 shall apply.

ii) Layer 1. This layer is required. The Alternate Mode 4 Challenge/Reply field shall be set to the value that represents the evaluation of the received Mode 4 interrogation. Parameter 4 shall contain a Mode 4 Code record that either includes a Mode 4 Pseudo Crypto Code or the indication that only Alternate Mode 4 data is available regardless of the Parameter 4 content of the received *interactive* IFF PDU interrogation.

iii) Layer 2. This layer shall be included if it has been included in the *regeneration* IFF PDU issued by this IFF system.

iv) Layer 5. This layer shall contain two IFF Data records in the following order:

— Basic Interactive
— Interactive Mode 4 Reply

d) *Interactive Mode—Receipt Rules*

1) A transponder simulation that has at least one of its entities currently capable of replying to an *interactive* IFF PDU that contains a Mode 4 interrogation shall monitor for an *interactive* IFF PDU to determine whether the Entity ID in the Interrogated Entity ID matches one of its entities. It shall then reply to the *interactive* IFF PDU containing the Mode 4 interrogation as specified in item c2) above.

2) An interrogator simulation that has sent an *interactive* IFF PDU representing a Mode 4 interrogation shall monitor for an *interactive* IFF PDU that is a reply to that interrogation:

i) If a reply is received and the Interrogating Entity ID, Interrogated Entity ID, Interactive Event ID match the sent IFF PDU interrogation, and the Transmission Indicator is set to Interrogation Reply (2), the reply shall be further processed. Otherwise, the received *interactive* IFF PDU shall be discarded.

ii) The Alternate Mode 4 Challenge/Reply information from Layer 1 represents the entity transponder's evaluation of the received Interactive Mode 4 interrogation.

3) An interrogator simulation may use either the Alternate Mode 4 Challenge/Reply or the Mode 4 pseudo crypto code, if present, to determine the Mode 4 reply status. However, if the Alternate

Mode 4 Challenge/Reply value is No Response (3), then that shall be used as the Mode 4 reply status regardless of any Mode 4 Pseudo Code comparison results.

B.5.1.2.3 Mode C Altitude issuance and receipt rules

Any simulation that transmits or receives and processes Mode C altitude for a system type shall adhere to the following requirements:

a) *Issuance Rules*
   1) Change/Options field. The Alternate Mode C Indicator shall always be set to one if this transponder is capable of transmitting Mode C information regardless of whether or not the actual Mode C altitude is also being transmitted.
   2) The output of actual Mode C altitude information is not a requirement of this standard.
   3) If actual Mode C altitude data is being transmitted in the Mode C Altitude record (B.2.33), an IFF PDU shall be transmitted when there is a change of ±100-feet in the entity's truth altitude to emulate the actual transmission requirements for Mode C data in the real world.

      NOTE—Due to the increased number of IFF PDUs that result from transmitting actual Mode C altitude, care has to be taken not to overload the simulation network.

b) *Receipt Rules*
   1) An interrogator simulation that receives and processes Mode C data shall have the capability to derive the Mode C altitude from remote entity truth data using an appropriate algorithm regardless of whether it also implements the capability to use an actual Mode C altitude when contained in the Mode C Altitude record (B.2.33).
   2) When both the Alternate Mode C Indicator is set and the actual Mode C altitude is received in the Mode C Altitude record (B.2.33), an interrogator simulation may use either source as the preferred source to use.

B.5.1.2.4 Mode 5 issuance and receipt rules

An interrogator or a transponder simulation that transmits or receives and processes Mode 5 for a system type shall adhere to the following Regeneration and Interactive Issuance and Receipt requirements:

a) *Regeneration Mode Issuance Rules*
   1) Mode 5 Status—Mode 5 Reply field. The Mode 5 Reply field shall always be set to an appropriate value regardless of whether or not the Crypto Control IFF Data record is also implemented. This field is required to be set for any transponder that has a Mode 5 capability.
   2) Crypto Control IFF Data record. If the optional Crypto Control IFF Data record is implemented for a transponder, the values included in that field shall be correlated with the equivalent values used in the Mode 5 Reply field. A Mode 5 capable interrogator may include this record in the IFF PDU to report its status.
   3) A Mode 5 capable interrogator simulation shall include the Crypto Control IFF Data record in its IFF PDU to report its status when it is in the Interactive Mode. Otherwise, inclusion of the Crypto Control IFF Data record is optional.
   4) The use of the Crypto Control IFF Data record requires pre-exercise coordination so that the appropriate pseudo crypto control information is set for each IFF system that has a Mode 5 capability. There is no requirement that all entities for a simulation have identical Crypto Control IFF Data records.
   5) The Mode 5 Reply value shall be set for a given IFF system based on exercise requirements and shall not be set based on the results of comparing a local Mode 5 pseudo crypto control information with a received Crypto Control IFF Data record in a Mode 5 interrogator IFF PDU.

      NOTE—Care needs to be taken in discussing Mode 5 as some of the implementation details are classified.

b) *Regeneration Mode Receipt Rules*
   1) An interrogator simulation that receives and processes Mode 5 Reply data shall store the Mode 5 Reply value received in a transponder IFF PDU for a remote IFF system even though the IFF PDU also contains a Crypto Control IFF Data record, which it prefers to process.
   2) An interrogator simulation may use either the Mode 5 reply value or compare the optional pseudo crypto control information that may be available to determine the Mode 5 Reply status except that if a Mode 5 Reply of No Response (0) is received, it shall take precedence over and be used regardless of the comparison results regarding pseudo crypto control information.
c) *Interactive Mode Issuance Rules*
   1) Interactive Mode 5 Interrogation. An interrogator that issues a Mode 5 interrogation (challenge) in the Interactive Mode shall adhere to the following requirements:
      i) The Interactive Mode requirements specified in 5.7.6.5 shall apply.
      ii) Layer 1 shall include the present regeneration data field values associated with the *regeneration* IFF PDU for this interrogator IFF system, except for the following fields of the Change/Options record:
         — The Change Indicator shall be set to Not Initial Report/No Change (0).
         — The Heartbeat Indicator shall be set to Not a Heartbeat (0).
         — The Simulation Mode shall be set to Interactive (1).
         — The Interactive Capable field shall be set to Capable (1).
         — Layer 2 may be included if implemented and deemed appropriate.
      iii) Layer 3 shall be included to reflect present Mode 5 regeneration data.
      iv) Layer 5 shall be included and contain, as a minimum, a Basic Interactive IFF Data record. The fields of this record shall be set as appropriate.
   2) Interactive Mode 5 Reply. An *interactive* IFF PDU reply shall contain the following layers and Mode 5 data:
      i) The Interactive requirements specified in 5.7.6.5 shall apply.
      ii) Layer 1. This layer is required.
      iii) Layer 2. This layer shall be included if it is included in the *regeneration* IFF PDU issued by this IFF system.
      iv) Layer 3. This layer shall be included and reflect regeneration Mode 5 data.
      v) Layer 5. This layer shall contain, as a minimum, the following IFF Data records in the order shown:
         — Basic Interactive
         — Interactive Mode 5 Reply
d) *Interactive Mode Receipt Rules*
   1) A transponder simulation that has at least one of its entities currently capable of replying to an *interactive* IFF PDU that contains a Mode 5 interrogation shall monitor for an *interactive* IFF PDU to determine whether the Entity ID in the Interrogated Entity ID matches one of its entities. It shall then reply to the *interactive* IFF PDU containing the Mode 5 interrogation as specified in item c2) in B.5.1.2.4.
   2) An interrogator simulation that has sent an *interactive* IFF PDU representing a Mode 5 interrogation shall monitor for an *interactive* IFF PDU that is a reply to that interrogation:
      i) If a reply is received (i.e., the Interrogating Entity ID, Interrogated Entity ID, and Interactive Event ID values match), and the Transmission Indicator is set to Interrogation Reply (2), then the reply shall be further processed. Otherwise, the received *interactive* IFF PDU shall be discarded.
      ii) The Interactive Mode 5 Reply IFF Data record represents the transponder's evaluation of the received interactive Mode 5 interrogation.

3) An interrogator simulation may use either the Layer 3, Mode 5 Reply field value or the Crypto Control IFF Data record, if present, to determine the Mode 5 reply status. However, if the Mode 5 Reply value is No Response (0), then that shall be used as the Mode 5 reply status regardless of any Mode 5 Crypto Code comparison results.

B.5.1.2.5 Special replies issuance and receipt rules

Any simulation that transmits or receives and processes special replies for a system type shall adhere to the following special issuance and receipt rules. The special reply indicators applicable to each transponder system and processed by each interrogator are shown in Table B.60. Special replies are contained in the Modifier record (see B.2.43):

a) *Special Reply Issuance Rules*
   1) Identification of Position (I/P) Special Reply
      i) When an IFF I/P Special Reply is initiated at a simulation (i.e., by a virtual pilot, Computer-Generated Forces (CGF) simulation operator, or activated by other logic), the IFF I/P Reply Timer (IFF_IP_REPLY_TIMER) shall be set and an IFF PDU shall be immediately issued with the Modifier record Identification of Position (I/P) field set to On (1).
      ii) When the IFF I/P Reply Timer expires, an IFF PDU shall be immediately issued with the Modifier record Identification of Position (I/P) field set to Off (0).
      iii) Any initiation of an I/P Special Reply while the IFF I/P Reply Timer is still active for a transponder associated with an entity shall cause the timer to be reset to begin the timeout process again. An IFF PDU shall also be immediately issued with the Modifier record Identification of Position (I/P) field set to On (1).
      iv) Any IFF PDU output for the transponder due to heartbeat or changed data requirements shall continue to show the Identification of Position (I/P) field set to On (1) until the IFF I/P Reply Timer has expired.
   2) Military Emergency Special Reply
      i) An IFF PDU shall be immediately issued whenever this special reply is initially set to On (1) or is set to Off (0) after having been set to On (1).
      ii) A military emergency special reply when set shall continue to be set until cleared.
   3) Unmanned Aircraft (X) Special Reply
      i) An IFF PDU shall be immediately issued whenever this special reply is initially set to On (1) or is set to Off (0) after having been set to On (1).
      ii) An Unmanned Aircraft special reply when set shall continue to be set until cleared.
b) *Special Reply Receipt Rules*
   1) Identification of Position (I/P) Special Reply
      i) When an IFF PDU is received with the Layer 1 Modifier record Identification of Position (I/P) field initially set to On (1), an interrogator that is capable of receiving this reply shall process it in accordance with the ICAO required number of replies a transponder would send based on its interrogation rate.

         NOTE—The IFF I/P Reply Timer (IFF_IP_REPLY_TIMER) should be set to a value that will allow all interrogators in a DIS exercise to receive and process an I/P Special Reply taking into account the simulation network reliability and receive processing delays.
   2) Military Emergency Special Reply
      i) When an IFF PDU is received with the Military Emergency reply set to On (1), any interrogator that is capable of receiving such a reply shall process it in accordance with the interrogator system requirements.
   3) Unmanned Aircraft (X) Special Reply

i) When an IFF PDU is received with the Unmanned Aircraft reply set to On (1), any interrogator that is capable of receiving such a reply shall process it in accordance with the interrogator system requirements.

B.5.1.2.6 Mode 5/S DAP issuance and receipt rules

Mode 5 and Mode S transponder systems transmit additional aircraft information including both static and potentially rapidly changing dynamic data such as flight maneuver and location information as Downlink Aircraft Parameters (DAPs). Some of these parameters are static or change infrequently. Other parameters may change frequently depending on the flight situation. Such rapidly changing data, even in the Regeneration Mode, could easily adversely affect bandwidth usage. Therefore, all data fields of Mode 5/S records that contain potentially rapidly changing data include an enumerated value to indicate that a receiving simulation is to use entity truth data to calculate the field value when the sending simulation has not included an IFF Data record containing the information. This enumerated value is referred to as the Compute Locally (0) option. A receiving simulation in the Regeneration Mode may always compute such values locally even if a DAP source field value is set to IFF Data Record Available (1). The algorithms and logic used to convert truth data to the value that would otherwise be contained in a field are beyond the scope of this standard. Typically, such algorithms will require access to entity truth and other database information in order to properly recreate the actual data value.

NOTE—The exercise agreement is expected to include a standard set of algorithms and other database information necessary to regenerate rapidly changing data fields contained in Mode 5 and Mode S transponder IFF PDUs. Otherwise, there is the possibility that a simulation, or simulation gateway interfaced to a fielded unit, will not have the same data value that could lead to interoperability issues.

a) *Mode 5/S Compute Locally Issuance Rules*
   1) A Mode 5 or Mode S transponder simulation that is operating in the Regeneration Mode shall set any field that includes a Compute Locally option to that enumeration value.
   2) A Mode 5 or Mode S transponder simulation operating in the Interactive Mode may either set the field to the actual value or set it to the Compute Locally enumerated value, if that option is available for the field.
b) *Mode 5/S Compute Locally Receipt Rules*
   1) A Mode 5 or Mode S interrogator model that is operating in the Regeneration Mode may always locally compute a field value where the option exists to compute the field locally even if the field is not set to Compute Locally.
   2) A Mode 5 or Mode S interrogator model that is operating in the Interactive Mode may either use an actual value received for a field or may choose to locally compute it if that fields has that option available.
   3) If a data field does not have the Compute Locally enumeration option, then the receiving simulation shall always use the value that is received in the field.

B.5.1.2.7 Test Mode field

The following requirements shall be applicable to any system type that is required to implement the Test Mode field of the Change/Options record (see B.2.4):

a) *Test Mode Issuance Rules*

   A system type that implements the Test Mode field shall adhere to the following issuance rules:

   1) When the Test Mode is set to On (1), other fields in the IFF PDU shall be set in accordance with the technical publication that specifies the transmit characteristics of this system type when in the test mode:
      i) Any change in data for other fields due to being in the test mode shall be reflected in the same IFF PDU that sets the Test Mode to On (1).

ii) A full IFF PDU containing all data for all applicable layers shall be issued when the Test Mode is set to On (1).

iii) A simulation may continue to store local IFF data that is no longer being sent due to the system being in the test mode.

2) When the Test Mode is set to Off (0) after having been set to On (1), any fields that changed will be restored and contain current data again if the field has been cleared due to being in the test mode.

3) Nothing in this standard shall preclude a simulation from including one or more IFF Data records to provide additional test mode-related data in an IFF PDU when in the test mode.

4) Heartbeat and data change requirements shall still be applicable, but the data shall continue to reflect what would appropriately be sent when in the test mode.

NOTE—If, for example, a specific transponder system operating in the Test Mode (On) ceases to output normal modes and codes, then those modes and codes are no longer contained in the IFF PDU and any receiving simulation that previously stored such codes is required to clear them from their database.

b) *Test Mode Receipt Rules*

There are no specific receipt rules for processing an IFF PDU when in the test mode.

B.5.1.2.8 System Designator field

The System Designator field is used to indicate whether multiple transponder, interrogator, or combined systems are associated with an entity. The requirements related to setting and processing this field are specified in item d) in 5.7.6.1.

B.5.1.2.9 System Mode field

The System Mode field of the System ID record is used to emulate settings that are available for a master switch that controls the overall operation of a transponder. Not all transponders may have all the modes defined for the System Mode field, and a system may have a setting that is not covered in this standard. The System ID record description contains a description of the System Mode field values and their relationship to the System On/Off Status field in the Fundamental Operational Data record (see 6.2.39).

NOTE—The System Mode field had no enumerations defined for it in SISO-REF-010 or in this standard prior to Protocol Version 7, and the System On/Off Status field was used to emulate the Master Switch for a transponder with two settings: On (1) or Off (0). The System Mode field values were defined to support additional fidelity requirements for the Master Switch setting in Protocol Version 7. Mode 5 and Mode S transponder systems are required to implement this field, and it is optional to be implemented by other systems.

a) *Issuance Rules*

If a system type is required to implement the System Mode field, it will be so indicated in the Information Content paragraph for that system type in B.5. (It may also be optionally implemented for some system types.) If a simulation implements the System Mode field, the following issuance rules shall apply:

1) All values applicable to the system type and system name shall be implemented including being able to be set and changed for an active entity during an exercise.

2) The System Mode field, if implemented, shall be correlated with the System On/Off Status field as specified in B.2.1.3.2 such that if a system mode value indicates an off condition, the System On/Off Status shall be set to Off (0), and if the System Mode field is a value that indicates the transponder is on (i.e., that the transponder would be emitting, either in response to an interrogation, or due to the squitter function being on), then the System On/Off Status field shall be set to On (1). The change to one field shall occur simultaneously with any changes required to be made to the other field such that only one IFF PDU shall be output when changes occur.

b) *Receipt Rules*
   1) A transponder is not required to process the System Mode field.
   2) An interrogator that processes the System Mode field shall adhere to the following rules:
      i) No Statement (0). If this value is received, no further processing of this field shall occur and only the System On/Off Status field and component On/Off Status fields shall be used to determine whether specific data in this IFF PDU should be processed.
      ii) Off (1). If this value is received, the transponder shall be considered off.
      iii) Standby (2). If this value is received, the transponder shall be considered off.
      iv) Normal/On (3). If this value is received, the transponder shall be considered on and operating as defined by the commercial or military specifications for this system type and system name. This is the usual mode that a transponder is in when turned on.
      v) Emergency (4). If this value is received, the transponder shall be considered on and operating in accordance with applicable military specifications for this system type and system name.
      vi) Low or Low Sensitivity (5). If this value is received, the transponder shall be considered on and operating as defined by the commercial or military specifications for this system type and system name.
   3) If there is a discrepancy between the System On/Off Status field value in the Fundamental Operational Data record and the System Mode value, the System On/Off Status field value shall take precedence.

B.5.1.2.10 Active Interrogation issuance and receipt rules

A Mode 5 interrogator has the option of issuing an IFF PDU to indicate an active interrogation when operating in the Regeneration mode. An active interrogation is used when there is a need to convey a specific Mode 5 interrogation request that would cause a visual or aural indication at the receiving Mode 5 transponder due to its importance. An example is the case of a Mode 5 lethal interrogation (see AIMS 03-1000A). If a Mode 5 interrogator or transponder implements active interrogations, the following issuance and receipt rules shall apply:

a) *Mode 5 Interrogator*
   1) The Active Interrogation Indicator (AII) shall be set to Active (1) in the PDU Status record for this *interrogator* IFF PDU (see 6.2.67).
   2) The Mode 5 Interrogator Status record, Message Formats Status field, shall be set to Active Interrogation (1).
   3) The Mode 5 Interrogator Basic Data record shall be set as follows:
      i) The Interrogated Entity ID field shall be set to the specific entity being interrogated. This is regardless of whether Layer 2 emissions data is also included in this IFF PDU.
      ii) The Mode 5 Message Formats record contained in the Mode 5 Message Formats Present field shall only identify the specific message format(s) requested in this active interrogation.
   4) The IFF PDU shall be sent at least once at the start of the interrogation. The IFF PDU may be present as appropriate to model actual interrogation time to ensure that it is received by the interrogated entity.
   5) IFF Layer 2 emissions data may also be included to support higher fidelity Mode 5 interrogators and transponders.

b) *Mode 5 Transponder*

   Although a Mode 5 transponder will normally not have a requirement to process a Mode 5 interrogator IFF PDU in the Regeneration mode, there may be circumstances when it would be beneficial to do so. The capability to process an active interrogation by a Mode 5 transponder when it is operating in the Regeneration mode is optional.

The following information is provided to support a Mode 5 transponder that implements receiving an active interrogation while in the Regeneration mode:

1) The Active Interrogation Indicator (AII) field of the PDU Status record of the received *interrogator* IFF PDU when set to Active (1) indicates that this is an active interrogation.

2) The Message Formats Status field of the Mode 5 Interrogator Status record will indicate an Active Interrogation (1). One or more of the message formats in the Mode 5 Message Formats Present field may cause a visual or aural alarm at a receiving Mode 5 transponder due to its importance.

3) The Interrogated Entity ID field of the Mode 5 Interrogator Basic Data record will identify the entity being interrogated. This can be especially useful for lower fidelity transponder models as an indication that they are the intended target of the interrogation.

   NOTE—It is possible that other entities besides the intended target of an interrogation may receive the interrogation signal. A simulated Mode 5 transponder that receives an active interrogation IFF PDU, but is not the intended target of the interrogation, may perform additional processing to determine whether it would have received this interrogation. IFF Layer 2 emissions data, if present, could be used in such processing.

4) A Mode 5 transponder that processes an active interrogation IFF PDU is not required to issue a *transponder* IFF PDU in reply. However, nothing in this standard precludes it from doing so.

B.5.2 System Type 1—Mark X/XII/ATCRBS Transponder

B.5.2.1 General

A Mark X, XII, or ATCRBS Transponder shall comply with the requirements specified herein. Although System Type 1 encompasses three types of transponders, only one of the three types of transponders shall be represented by the information conveyed in an IFF PDU. The type of transponder shall be specified by the System Name except that a simulation that has implemented DIS Protocol Version 6 may specify the System Name as a Generic Mark X/XII/ATCRBS (6) or Generic Mark X/XII/ATCRBS/Mode S (7). However, System Name 7 shall not be used to indicate the implementation of a Mode S capability as defined by a Mode S Transponder (System Type 11). Some ATCRBS capabilities are included in both the military Mark X and Mark XII systems. The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.63.

Table B.63—System Type 1 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	CR	CR	CR
Data Change	R	O	CR	CR	CR
Heartbeat	R	O	CR	CR	CR
Legend R Required O Optional CR Conditionally Required NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.2.2 Mark X Transponder

B.5.2.2.1 Information content

If the System Type 1, System Name indicates a Mark X, Mark X (SIF), or Mark X (A) transponder, the following Information Content requirements shall apply unless otherwise indicated:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status record. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Optional*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*
      iv) Data Field 2. Modifier record. *Required*
      v) Parameter 1. Mode 1 Code record. *Required*
      vi) Parameter 2. Mode 2 Code record. *Required*
      vii) Parameter 3. Mode 3/A Code record. *Required*
      viii) Parameter 4. *Not Applicable*
      ix) Parameter 5. Mode C Altitude record. Mark X, Mark X (SIF). *Not Applicable*; Mark X (A). *Required*
      x) Parameter 6. *Not Applicable*

b) Layer 2. *Optional*
   1) Layer Header. *Required*
   2) Beam Data. *Required*
   3) Secondary Operational Data. *Required*
   4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layers 3 and 4. *Conditionally Required*

d) Layer 5. *Conditionally Required*

e) Layers 6 and 7. *Not Defined*

B.5.2.2.2 Issuance rules

There are no additional issuance rules specific to this system type.

B.5.2.2.3 Receipt rules

There are no additional receipt rules specific to this system type.

B.5.2.3 Mark XII Transponder

B.5.2.3.1 Information content

If the System Type 1 indicates a Mark XII transponder, the following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
    1) PDU Header
        i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
        ii) PDU Status. *Not Applicable*
    2) Emitting Entity ID. *Required*
    3) Event ID. *Optional*
    4) Antenna Location. *Required*
    5) System ID
        i) System Type. *Required*
        ii) System Name. *Required*
        iii) System Mode. *Optional*
        iv) Change/Options record. *See B.2.4 for required fields to implement*
    6) System Designator. *Required*
    7) System-Specific Data. *Not Applicable*
    8) Fundamental Operational Data
        i) System Status. *Required*
        ii) Data Field 1. Alternate Mode 4 Challenge/Reply. *Required*
        iii) Information Layers. *Required*
        iv) Data Field 2. Modifier record. *Required*
        v) Parameter 1. Mode 1 Code record. *Required*
        vi) Parameter 2. Mode 2 Code record. *Required*
        vii) Parameter 3. Mode 3/A Code record. *Required*
        viii) Parameter 4. Mode 4 Code record. *Required*
        ix) Parameter 5. Mode C Altitude record. *Required*
        x) Parameter 6. TCAS/ACAS Status record. *Required if a TCAS/ACAS system is present*

b) Layer 2. *Optional*
    1) Layer Header. *Required*
    2) Beam Data. *Required*
    3) Secondary Operational Data. *Required*
    4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layers 3 and 4. *Conditionally Required*

d) Layer 5. *Conditionally Required*

e) Layers 6 and 7. *Not Defined*

B.5.2.3.2 Issuance rules

In addition to general transponder issuance requirements specified elsewhere, the following additional rules shall apply:

a) *Mode 4 Reply*. See B.5.1.2.2 for the issuance rules regarding setting and sending a Mode 4 reply.

b) *Mode C Altitude*. See B.5.1.2.3 for the issuance rules regarding setting and sending Mode C altitude.

c) *Special Replies*. See B.5.1.2.5 for the issuance rules regarding setting and sending a special reply.

B.5.2.3.3 Receipt rules

There are no additional receipt rules specific to this system type.

B.5.2.4 ATCRBS Transponder

B.5.2.4.1 Information content

If the System Type 1, System Name indicates an ATCRBS transponder, the following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Optional*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*
      iv) Data Field 2. Modifier record (I/P Only field). *Required*
      v) Parameter 1. *Not Applicable*
      vi) Parameter 2. *Not Applicable*
      vii) Parameter 3. Mode 3/A Code record. *Required*
      viii) Parameter 4. *Not Applicable*

ix) Parameter 5. Mode C Altitude record. *Not Applicable*

x) Parameter 6. *Not Applicable*

b) Layer 2. *Optional*

1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layers 3 and 4. *Conditionally Required*

d) Layer 5. *Conditionally Required*

e) Layers 6 and 7. *Not Defined*

B.5.2.4.2 Issuance rules

In addition to general transponder issuance requirements specified elsewhere, the following additional rules shall apply:

a) *Mode C Altitude*. See B.5.1.2.3 for the issuance rules regarding setting and sending Mode C altitude.

b) *Special Replies*. See B.5.1.2.5 for the issuance rules regarding setting and sending a special reply.

B.5.2.4.3 Receipt rules

There are no additional issuance rules specific to this system type.

B.5.3 System Type 2—Mark X/XII/ATCRBS Interrogator

B.5.3.1 General

The specific Mark X/XII/ATCRBS Interrogator information content, issuance, and receipt rules are contained in this subclause. Although System Type 2 encompasses three types of interrogators, only one of the three types of interrogators shall be represented by the information conveyed in an IFF PDU. The type of interrogator shall be specified by the System Name except that a simulation that has implemented DIS Protocol Version 6 may specify the System Name as a Generic Mark X/XII/ATCRBS (6) or Generic Mark X/XII/ATCRBS/Mode S (7). However, System Name 7 shall not be used to indicate the implementation of a Mode S capability as defined by a Mode S interrogator (System Type 8). Some ATCRBS capabilities are included in both the military Mark X and Mark XII systems. The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.64.

Table B.64—System Type 2 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	CR	CR	CR

Table B.64—System Type 2 layer requirements *(continued)*

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Data Change	R	O	CR	CR	CR
Heartbeat	R	O	CR	CR	CR
Legend R Required O Optional CR Conditionally Required NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.3.2 Mark X Interrogator

B.5.3.2.1 Information content

A Mark X Interrogator IFF PDU shall contain the following information. All fields listed for a layer shall be implemented and shall contain at least one valid value unless otherwise indicated. Any optional fields or other fields that are not applicable shall have all bits of the field set to zero:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Optional*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*
      iv) Data Field 2. *Not Applicable*
      v) Parameter 1. Mode 1 Interrogator Status record. *Required*
      vi) Parameter 2. Mode 2 Interrogator Status record. *Required*
      vii) Parameter 3. Mode 3/A Interrogator Status record. *Required*
      viii) Parameter 4. *Not Applicable*
      ix) Parameter 5. *Not Applicable*

        x) Parameter 6. *Not Applicable*
- b) Layer 2. *Optional*
    1) Layer Header. *Required*
    2) Beam Data. *Required*
    3) Secondary Operational Data. *Required*
    4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*
- c) Layers 3 and 4. *Conditionally Required*
- d) Layer 5. *Conditionally Required*
- e) Layers 6 and 7. *Not Defined*

B.5.3.2.2 Issuance rules

There are no additional issuance rules specific to this system type.

B.5.3.2.3 Receipt rules

There are no additional receipt rules specific to this system type.

B.5.3.3 Mark XII Interrogator

B.5.3.3.1 Information content

A Mark XII Interrogator IFF PDU shall contain the following information. All fields listed for a layer shall be implemented and shall contain at least one valid value unless otherwise indicated. Any optional fields or other fields that are not applicable shall have all bits of the field set to zero:

- a) Layer 1. *Required*
    1) PDU Header
        - i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
        - ii) PDU Status. *Not Applicable*
    2) Emitting Entity ID. *Required*
    3) Event ID. *Optional*
    4) Antenna Location. *Required*
    5) System ID
        - i) System Type. *Required*
        - ii) System Name. *Required*
        - iii) System Mode. *Optional*
        - iv) Change/Options record. *See B.2.4 for required fields to implement*
    6) System Designator. *Required*
    7) System-Specific Data. *Not Applicable*
    8) Fundamental Operational Data
        - i) System Status. *Required*
        - ii) Data Field 1 (Alternate Mode 4 Challenge/Reply). *Required*
        - iii) Information Layers. *Required*
        - iv) Data Field 2. *Not Applicable*
        - v) Parameter 1. Mode 1 Interrogator Status record. *Required*

vi) Parameter 2. Mode 2 Interrogator Status record. *Required*

vii) Parameter 3. Mode 3/A Interrogator Status record. *Required*

viii) Parameter 4. Mode 4 Interrogator Status record. *Required*

ix) Parameter 5. Mode C Interrogator Status record. *Required*

x) Parameter 6. TCAS/ACAS Status record. *Required if a TCAS/ACAS system is present*

b) Layer 2. *Optional*

1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layer 3 and 4. *Conditionally Required*

d) Layer 5. *Conditionally Required*

e) Layers 6 and 7. *Not Defined*

B.5.3.3.2 Issuance rules

There are no additional issuance rules specific to this system type.

B.5.3.3.3 Receipt rules

There are no additional receipt rules specific to this system type.

B.5.3.4 ATCRBS Interrogator

B.5.3.4.1 Information content

An ATCRBS Interrogator IFF PDU shall contain the following information. All fields listed for a layer shall be implemented and shall contain at least one valid value unless otherwise indicated. Any optional fields or other fields that are not applicable shall have all bits of the field set to zero:

a) Layer 1. *Required*

1) PDU Header

i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*

ii) PDU Status. *Not Applicable*

2) Emitting Entity ID. *Required*

3) Event ID. *Optional*

4) Antenna Location. *Required*

5) System ID

i) System Type. *Required*

ii) System Name. *Required*

iii) System Mode. *Optional*

iv) Change/Options record. *See B.2.4 for required fields to implement*

6) System Designator. *Required*

7) System-Specific Data. *Not Applicable*

8) Fundamental Operational Data

i) System Status. *Required*

ii) Data Field 1. *Not Applicable*

iii) Information Layers. *Required*

iv) Data Field 2. *Not Applicable*

v) Parameter 1. *Not Applicable*

vi) Parameter 2. *Not Applicable*

vii) Parameter 3. Mode 3/A Interrogator Status record. *Required*

viii) Parameter 4. *Not Applicable*

ix) Parameter 5. Mode C Interrogator Status record. *Required*

x) Parameter 6. TCAS/ACAS Status record. *Required if a TCAS/ACAS system is present*

b) Layer 2. *Optional*

1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layers 3 and 4. *Conditionally Required*

d) Layer 5. *Conditionally Required*

e) Layers 6 and 7. *Not Defined*

B.5.3.4.2 Issuance rules

There are no additional issuance rules specific to this system type.

B.5.3.4.3 Receipt rules

There are no additional receipt rules specific to this system type.

B.5.4 System Type 3—Soviet Transponder

B.5.4.1 General

The specific requirements for the Soviet transponder are specified herein.

The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.65.

Table B.65—System Type 3 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	NA	CR	CR

Table B.65—System Type 3 layer requirements *(continued)*

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Data Change	R	O	NA	CR	CR
Heartbeat	R	O	NA	CR	CR
Legend R Required O Optional CR Conditionally Required NA Not Applicable NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.4.2 Information content

The following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Optional*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*
      iv) Data Field 2. *Not Applicable*
      v) Parameter 1. Soviet Transponder Status record. *Required*
      vi) Parameter 2. *Not Applicable*
      vii) Parameter 3. Mode 3/A record. *Required if has a Mode 3/A capability*
      viii) Parameter 4. *Not Applicable*
      ix) Parameter 5. Mode C Altitude record. *Required if has a Mode C capability*
      x) Parameter 6. TCAS/ACAS Status record. *Required if has a TCAS/ACAS capability*

b) Layer 2. *Optional*
   1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layer 3. *Not Applicable*

d) Layers 4 and 5. *Conditionally Required*

e) Layers 6 and 7. *Not Defined*

B.5.4.3 Issuance rules

There are no additional issuance rules specific to this system type.

B.5.4.4 Receipt rules

There are no additional receipt rules specific to this system type.

B.5.5 System Type 4—Soviet Interrogator

B.5.5.1 General

The Soviet interrogator system is also referred to as a Red IFF interrogator. It is used by Russian-built IFF systems that are present in various countries. The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.66.

Table B.66—System Type 4 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	NA	CR	CR
Data Change	R	O	NA	CR	CR
Heartbeat	R	O	NA	CR	CR

Legend

R Required
O Optional
CR Conditionally Required
NA Not Applicable

NOTE—See B.5.1.1 for definitions of the codes used in this table.

B.5.5.2 Information content

The following minimum Information Content requirements shall apply:

a) Layer 1. *Required*

1) PDU Header

i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*

ii) PDU Status. *Not Applicable*

2) Emitting Entity ID. *Required*
3) Event ID. *Optional*
4) Antenna Location. *Required*
5) System ID
    i) System Type. *Required*
    ii) System Name. *Required*
    iii) System Mode. *Optional*
    iv) Change/Options record. *See B.2.4 for required fields to implement*
6) System Designator. *Required*
7) System-Specific Data. *Not Applicable*
8) Fundamental Operational Data
    i) System Status. *Required*
    ii) Data Field 1. *Not Applicable*
    iii) Information Layers. *Required*
    iv) Data Field 2. *Not Applicable*
    v) Parameter 1. Soviet Interrogator Status record. *Required*
    vi) Parameter 2. *Not Applicable*
    vii) Parameter 3. *Not Applicable*
    viii) Parameter 4. *Not Applicable*
    ix) Parameter 5. *Not Applicable*
    x) Parameter 6. *Not Applicable*

b) Layer 2. *Optional*
    1) Layer Header. *Required*
    2) Beam Data. *Required*
    3) Secondary Operational Data. *Required*
    4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*
c) Layer 3. *Not Applicable*
d) Layer 4. *Conditionally Required*
e) Layer 5. *Conditionally Required*
f) Layers 6 and 7. *Not Defined*

B.5.5.3 Issuance rules

There are no additional issuance rules specific to this system type.

B.5.5.4 Receipt rules

There are no additional receipt rules specific to this system type.

B.5.6 System Type 5—RRB Transponder

B.5.6.1 General

The specific requirements for the Reply Receiver ‘B’ (RRB) transponder are specified herein. The RRB transponder is interrogated by a primary radar and gives an edge of band response. Hence, there are no interrogators. An RRB transponder is fitted to a number of aircraft such as the Lynx HAS 3 helicopter and the Sea Harrier. An RRB receiver aboard a platform such as a ship receives the RRB transmission and displays an indication on a radar screen. The response contains 1 of 16 reply codes (normally interpreted as a decimal number, not octal). It also has a radar enhancement signal, which works somewhat similar to a Squawk Ident. The RRB transponder is in use by the British Royal Navy.

The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.67.

Table B.67—System Type 5 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	NA	NA	CR
Data Change	R	O	NA	NA	CR
Heartbeat	R	O	NA	NA	CR
Legend R Required O Optional CR Conditionally Required NA Not Applicable NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.6.2 Information content

If the System Type 5, System Name indicates an RRB transponder, the following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Optional*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*

iv) Data Field 2. *Not Applicable*

v) Parameter 1. RRB Code record. *Required*

vi) Parameter 2. *Not Applicable*

vii) Parameter 3. *Not Applicable*

viii) Parameter 4. *Not Applicable*

ix) Parameter 5. *Not Applicable*

x) Parameter 6. *Not Applicable*

b) Layer 2. *Optional*

1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layers 3 and 4. *Not Applicable*

d) Layer 5. *Conditionally Required*

e) Layers 6 and 7. *Not Defined*

B.5.6.3 Issuance rules

The following additional rules shall apply to an RRB transponder:

a) *Fundamental Operational Data Record: System Status*. All RRB transponders shall set the Parameter 1 Capable field of the System Status record to Capable (0) and shall set the Parameter 2 to 6 Capable fields to Not Capable (1).

b) *Fundamental Operational Data Record: System Status*. The System On/Off Status shall be set to On (1) if the On/Off Status (bit 13 in Parameter 1) is set to On (1), the Damage Status (bit 14 in Parameter 1) is set to No Damage (0), and the Malfunction Status (bit 15 in Parameter 1) is set to No Malfunction (0). Otherwise the System On/Off Status shall be set to Off (0).

B.5.6.4 Receipt rules

There are no additional receipt rules for this system type.

B.5.7 System Type 6—Mark XIIA Interrogator

B.5.7.1 General

Unique Mark XIIA interrogator requirements shall be as defined in this subclause. The IFF PDU general issuance and receipt rules specified in 5.7.6.3 and 5.7.6.4 shall apply unless superseded by the requirements contained herein.

The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.68.

Table B.68—System Type 6 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	R	NA	CR
Data Change	R	O	R	NA	CR
Heartbeat	R	O	R	NA	CR
Legend R Required O Optional CR Conditionally Required NA Not Applicable NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.7.2 Information content

A Mark XIIA Interrogator IFF PDU shall contain the following information. All fields listed for a layer shall be implemented and shall contain at least one valid value unless otherwise indicated. Any optional fields or other fields that are not used shall have all bits of the field set to zero:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Optional*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*
      iv) Data Field 2. *Not Applicable*
      v) Parameter 1. Mode 1 Interrogator Status record. *Required*
      vi) Parameter 2. Mode 2 Interrogator Status record. *Required*
      vii) Parameter 3. Mode 3/A Interrogator Status record. *Required*

viii) Parameter 4. Mode 4 Interrogator Status record. *Required*

ix) Parameter 5. Mode C Interrogator Status record. *Required*

x) Parameter 6. TCAS/ACAS Status record. *Required if TCAS/ACAS present*

b) Layer 2. *Optional*

1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layer 3. *Required*

1) Layer Header. *Required*

2) Reporting Simulation. *Required*

3) Mode 5 Interrogator Basic Data. *Required*

4) Number of Records. *Required*

5) IFF Data records. Crypto Control IFF Data record. *Required if this interrogator model has implemented a pseudo crypto capability*

d) Layer 4. *Not Applicable*

e) Layer 5. *Conditionally Required*

f) Layers 6 and 7. *Not Defined*

B.5.7.3 Issuance rules

See B.5.1.2.10 for the issuance rules related to active interrogations.

B.5.7.4 Receipt rules

The following IFF PDU receipt rules shall apply to a simulation that models a Mark XIIA interrogator in addition to those contained in 5.7.6.4. If there is a conflict, the rules contained herein shall apply:

a) *Mode 4 Reply*. See B.5.1.2.2 for the receipt rules regarding processing a Mode 4 reply.

b) *Mode C Altitude*. See B.5.1.2.3 for the issuance rules regarding setting and sending Mode C altitude.

c) *Mode 5 Reply*. See item b) in B.5.1.2.4 for the receipt rules regarding processing a Mode 5 reply.

d) A Mark XIIA interrogator shall process a received IFF PDU from transponder system types as specified in Table B.62:

1) The data fields shall be stored and replace existing data for the layer(s) received.

2) See item h) in 5.7.6.1 for transponder data requirements related to an interrogator whose System Status is Off (0).

e) Nothing in this standard shall preclude a Mark XIIA interrogator simulation from processing IFF PDUs representing other transponders, or interrogators, that are needed to support the simulation in addition to required transponder system types.

B.5.8 System Type 7—Mode 5 Interrogator

B.5.8.1 General

A Mode 5 Interrogator IFF PDU shall contain the following information. All fields listed for a layer shall be implemented and shall contain at least one valid value unless otherwise indicated. Any optional fields or other fields that are not used shall have all bits of the field set to zero.

The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.69.

Table B.69—System Type 7 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	R	NA	CR
Data Change	R	O	R	NA	CR
Heartbeat	R	O	R	NA	CR
Legend R Required O Optional CR Conditionally Required NA Not Applicable NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.8.2 Information content

The following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Optional*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*
      iv) Data Field 2. *Not Applicable*
      v) Parameter 1. Mode 1 Interrogator Status record. *Required*
      vi) Parameter 2. Mode 2 Interrogator Status record. *Required*
      vii) Parameter 3. Mode 3/A Interrogator Status record. *Required*
      viii) Parameter 4. Mode 4 Interrogator Status record. *Required*

ix) Parameter 5. Mode C Interrogator Status record. *Required*

x) Parameter 6. TCAS/ACAS Status record. *Required if TCAS/ACAS present*

b) Layer 2. *Optional*

1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layer 3. *Required*

1) Layer Header. *Required*

2) Reporting Simulation. *Required*

3) Mode 5 Interrogator Basic Data. *Required*

4) Number of Records. *Required*

5) IFF Data records. *Crypto Control IFF Data record required if this interrogator model has implemented a pseudo crypto capability*

d) Layer 4. *Not Applicable*

e) Layer 5. *Conditionally Required*

f) Layers 6 and 7. *Not Defined*

B.5.8.3 Issuance rules

See B.5.1.2.10 for the issuance rules related to active interrogations.

B.5.8.4 Receipt rules

The following IFF PDU receipt rules shall apply to a simulation that models a Mode 5 interrogator in addition to those contained in 5.7.6.4. If there is a conflict, the rules contained herein shall apply:

a) A Mode 5 interrogator shall process a received IFF PDU from transponder system types as specified in Table B.62:

1) The data fields shall be stored and replace existing data for the layer(s) received. If a layer previously stored is not received in a Change IFF PDU, the present data shall be retained.

2) The Change/Option record—Heartbeat Indicator shall be checked. If it is set to Heartbeat (1), this represents a full IFF PDU and any layers previously received that are not received shall be cleared.

3) See item h) in 5.7.6.1 for transponder data requirements related to an interrogator whose System Status is Off (0).

4) The relationship and required processing of the System On/Off Status field, System Mode field, and On/Off Status fields of various component records (e.g., Mode 1 Code record) are specified in item b) in B.5.1.2.9.

b) Nothing in this standard shall preclude a Mode S interrogator simulation from processing IFF PDUs representing other transponders, or interrogators, that are needed to support the simulation in addition to required transponder system types.

B.5.9 System Type 8—Mode S Interrogator

B.5.9.1 General

A Mode S Interrogator IFF PDU shall contain the following information. All fields listed for a layer shall be implemented and shall contain at least one valid value unless otherwise indicated. Any optional fields or other fields that are not used shall have all bits of the field set to zero. The layers to be included in a transmitted IFF PDU for this system type shall be as shown in Table B.70.

Table B.70—System Type 8 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	NA	NA	CR
Data Change	R	O	NA	NA	CR
Heartbeat	R	O	NA	NA	CR
Legend R Required O Optional CR Conditionally Required NA Not Applicable NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.9.2 Information content

The following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Required*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*

iv) Data Field 2. *Not Applicable*

v) Parameter 1. *Not Applicable*

vi) Parameter 2. *Not Applicable*

vii) Parameter 3. Mode 3/A Interrogator Status record. *Required*

viii) Parameter 4. Mode S Interrogator Identifier record. *Required*

ix) Parameter 5. Mode C Interrogator Status record. *Required*

x) Parameter 6. TCAS/ACAS Status record. *Required if TCAS/ACAS present*

b) Layer 2. *Optional*
   1) Layer Header. *Required*
   2) Beam Data. *Required*
   3) Secondary Operational Data. *Required*
   4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layer 3. *Not Applicable*

d) Layer 4. *Required*

e) Layer 5. *Conditionally Required*

f) Layers 6 and 7. *Not Defined*

B.5.9.3 Issuance rules

There are no additional issuance rules specific to this system type.

B.5.9.4 Receipt rules

The following additional IFF PDU receipt rules shall apply to a simulation that models a Mode S interrogator in addition to those contained in 5.7.6.4. If there is a conflict, the rules contained herein shall apply:

a) A Mode S interrogator shall process a received IFF PDU from transponder system types as specified in Table B.62.

b) See item h) in 5.7.6.1 for transponder data requirements related to an interrogator whose System Status is Off (0).

c) Nothing in this standard shall preclude a Mode S interrogator simulation from processing IFF PDUs representing other transponders, or interrogators, that are needed to support the simulation in addition to required transponder system types.

B.5.10 System Type 9—Mark XIIA Transponder

B.5.10.1 General

The IFF PDU general issuance and receipt rules specified in 5.7.6.3 and 5.7.6.4 shall also apply unless superseded by requirement specified herein. The requirements related to which layers are transmitted under various conditions shall be as specified in Table B.71.

Table B.71—System Type 9 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	R	R	CR
Data Change	R	O	R	R	CR
Heartbeat	R	O	R	R	CR

Legend

R Required
O Optional
CR Conditionally Required

NOTE—See B.5.1.1 for definitions of the codes used in this table.

B.5.10.2 Information content

The Mark XIIA Transponder IFF PDU shall contain the following information:

a) Layer 1. *Required*
  1) PDU Header
    i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
    ii) PDU Status. *Not Applicable*
  2) Emitting Entity ID. *Required*
  3) Event ID. *Optional*
  4) Antenna Location. *Required*
  5) System ID
    i) System Type. *Required*
    ii) System Name. *Required*
    iii) System Mode. *Required*
    iv) Change/Options record. *See B.2.4 for required fields to implement*
  6) System Designator. *Required*
  7) System-Specific Data. *Not Applicable*
  8) Fundamental Operational Data
    i) System Status. *Required*
    ii) Data Field 1. Alternate Mode 4 Challenge/Reply. *Required if this aircraft has Mode 4 in addition to Mode 5*
    iii) Information Layers. *Required*

iv) Data Field 2. Modifier record. *Required*

v) Parameter 1. Mode 1 Code record. *Required*

vi) Parameter 2. Mode 2 Code record. *Required*

vii) Parameter 3. Mode 3/A Code record. *Required*

viii) Parameter 4. Mode 4 Code record. *Required*

ix) Parameter 5. Mode C Altitude record. *Required*

x) Parameter 6. TCAS/ACAS Status record. *Required if a TCAS/ACAS system is present*

b) Layer 2. *Optional*
   1) Layer Header. *Required*
   2) Beam Data. *Required*
   3) Secondary Operational Data. *Required*
   4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layer 3. *Required*
   1) Layer Header. *Required*
   2) Reporting Simulation. *Required*
   3) Mode 5 Transponder Basic Data. *All fields and associated records are required*
   4) Number of IFF Data Records. *Required*
   5) IFF Data records. *As Applicable*

d) Layer 4. Required
   1) Layer Header. *Required*
   2) Reporting Simulation. *Required*
   3) Mode S Transponder Basic Data. *All fields and associated records are required*
   4) Number of IFF Data Records. *Required*
   5) IFF Data records. *As Applicable*

e) Layer 5. *Conditionally Required*

f) Layers 6 and 7. *Not Defined*

B.5.10.3 Issuance rules

The following additional IFF PDU issuance rules shall apply to a simulation that models a Mark XIIA transponder:

a) When a Mark XIIA Transponder System Status is set to Off (0), the following requirements shall be met:
   1) If a remote entity, the stored IFF data may be cleared or retained.
   2) If a local entity, the stored IFF data shall be retained.

b) *Mode 4 Reply*. See B.5.1.2.2 for the issuance rules regarding setting and sending a Mode 4 reply.

c) *Mode C Altitude*. See B.5.1.2.3 for the issuance rules regarding setting and sending Mode C altitude.

d) *Mode 5 Reply*. See B.5.1.2.4 for the issuance rules regarding setting and sending a Mode 5 reply.

e) *Special Replies*. See B.5.1.2.5 for the issuance rules regarding setting and sending a special reply.

f) *Mode 5 DAP Data*. See B.5.1.2.6 for the issuance rules related to Mode 5/S downlink aircraft parameters.

g) *Test Mode*. See B.5.1.2.7 for the issuance rules related to implementation of the Test Mode.

h) *Fundamental Operational Data Record, Parameter 1, Mode 1 Code Record*. The Mode 1 Code shall be set identical to the Enhanced Mode 1 Code value as specified in B.2.9.

i) *System Mode*. As a minimum, the following System Mode field values in the System ID record shall be able to be set: Off (1), Standby (2), and Normal/On (3). No Statement (0) shall not be valid for this field. A status of Off (1) shall indicate that the transponder is off, no power is being applied; it would not respond to an interrogation emitting any data and would not respond to any interrogation.

B.5.10.4 Receipt rules

See B.5.1.2.10 for the receipt rules related to active interrogations.

B.5.11 System Type 10—Mode 5 Transponder

B.5.11.1 General

This system type shall only be used when an entity only has a Mode 5 transponder without also having a Mode S capability. A Mode 5 transponder, however, may have a Mode 4 and ATCRB modes. The layers required to be transmitted for a given condition shall be as shown in Table B.72.

Table B.72—System Type 10 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	R	NA	CR
Data Change	R	O	R	NA	CR
Heartbeat	R	O	R	NA	CR
Legend R Required O Optional CR Conditionally Required NA Not Applicable NOTE—See B.5.1.1 for definitions of the codes used in this table.					

B.5.11.2 Information content

The following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Required*

iv) Change/Options record. *See B.2.4 for required fields to implement*

6) System Designator. *Required*

7) System-Specific Data. *Not Applicable*

8) Fundamental Operational Data

i) System Status. *Required*

ii) Data Field 1. Alternate Mode 4 Challenge/Reply. *Required if this aircraft has Mode 4 in addition to Mode 5*

iii) Information Layers. *Required*

iv) Data Field 2. Modifier record. *Required*

v) Parameter 1. Mode 1 Code record. *Required if has a Mode 1 capability*

vi) Parameter 2. Mode 2 Code record. *Required if has a Mode 2 capability*

vii) Parameter 3. Mode 3/A Code record. *Required if has a Mode 3/A capability*

viii) Parameter 4. Mode 4 Code record. *Required if also has a Mode 4 capability*

ix) Parameter 5. Mode C Altitude record. *Required if has a Mode C capability*

x) Parameter 6. TCAS/ACAS Status record. *Required if has a TCAS/ACAS system*

b) Layer 2. *Optional*

1) Layer Header. *Required*

2) Beam Data. *Required*

3) Secondary Operational Data. *Required*

4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layer 3. *Required*

1) Layer Header. *Required*

2) Reporting Simulation. *Required*

3) Mode 5 Transponder Basic Data. *All fields and associated records are required*

4) Number of IFF Data Records. *Required*

5) IFF Data records. *As Applicable*

d) Layer 4. *Not Applicable*

e) Layer 5. *Conditionally Required*

f) Layers 6 and 7. *Not Defined*

B.5.11.3 Issuance rules

The following additional IFF PDU issuance rules shall apply to a simulation that models a Mode 5 transponder:

a) When a Mode 5 Transponder System Status is set to Off (0), the following requirements shall be met:

1) If a remote entity, the stored IFF data may be cleared or retained.

2) If a local entity, the stored IFF data shall be retained.

b) *Mode 4 Reply*. See B.5.1.2.2 for the issuance rules regarding setting and sending a Mode 4 reply.

c) *Mode C Altitude*. See B.5.1.2.3 for the issuance rules regarding setting and sending Mode C altitude.

d) *Mode 5 Reply*. See B.5.1.2.4 for the issuance rules regarding setting and sending a Mode 5 reply.

e) *Special Replies*. See B.5.1.2.5 for the issuance rules regarding setting and sending a special reply.

f) *Mode 5 DAP Data*. See B.5.1.2.6 for the issuance rules related to Mode 5/S downlink aircraft parameters.

g) *Test Mode*. See B.5.1.2.7 for the issuance rules related to implementation of the Test Mode.

h) *Fundamental Operational Data Record, Parameter 1, Mode 1 Code Record*. The Mode 1 Code shall be set identical to the Enhanced Mode 1 Code value as specified in B.2.9.

i) *System Modes*. As a minimum, the following System Mode field values in the System ID record shall be able to be set: Off (1), Standby (2), and Normal/On (3). No Statement (0) shall not be valid for this field.

B.5.11.4 Receipt rules

See B.5.1.2.10 for the receipt rules related to active interrogations.

B.5.12 System Type 11—Mode S Transponder

B.5.12.1 General

A Mode S Transponder is capable of transmitting Mode S and Mode 3/A and Mode C data. Mode S transponder data shall be required to be transmitted using the IFF PDU for any aircraft entity type, including airborne units such as AWACS, that have a Mode S transponder capability. A simulation other than the owner of the aircraft entity may generate the IFF PDU containing this information. The minimum required layers for a Mode S transponder are Layers 1 and 4. A single IFF PDU shall contain all data associated with Mode S including any Mode 3/A, C, TACS/ACAS, and special reply data that the aircraft's Mode S transponder would be capable of producing.

The IFF PDU general issuance and receipt rules specified in 5.7.6.3 and 5.7.6.4 shall apply unless superseded by requirement specified herein.

A Mode S-capable transponder may also implement Mode S emissions characteristic data and Mode S GICB data link messages.

The required layers to be output for various conditions shall be as specified in Table B.73.

Table B.73—System Type 11 layer requirements

Situation	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Initial Report	R	O	NA	R	CR
Data Change	R	O	NA	R	CR
Heartbeat	R	O	NA	R	CR

Legend

R Required
O Optional
CR Conditionally Required
NA Not Applicable

NOTE—See B.5.1.1 for definitions of the codes used in this table.

B.5.12.2 Information content

The following minimum Information Content requirements shall apply:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*
      ii) System Name. *Required*
      iii) System Mode. *Required*
      iv) Change/Options record. *See B.2.4 for required fields to implement*
   6) System Designator. *Required*
   7) System-Specific Data. *Not Applicable*
   8) Fundamental Operational Data
      i) System Status. *Required*
      ii) Data Field 1. *Not Applicable*
      iii) Information Layers. *Required*
      iv) Data Field 2. Modifier record. *Required*
      v) Parameter 1. *Not Applicable*
      vi) Parameter 2. *Not Applicable*
      vii) Parameter 3. Mode 3/A Code record. *Required*
      viii) Parameter 4. *Not Applicable*
      ix) Parameter 5. Mode C Altitude record. *Required*
      x) Parameter 6. TCAS/ACAS Status record. *Required if a TCAS/ACAS system is present*

b) Layer 2. *Optional*
   1) Layer Header. *Required*
   2) Beam Data. *Required*
   3) Secondary Operational Data. *Required*
   4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*

c) Layers 3. *Not Applicable*

d) Layer 4. *Required*
   1) Layer Header. *Required*
   2) Reporting Simulation. *Required*
   3) Mode S Transponder Basic Data. *All fields and associated records are required*
   4) Number of IFF Data Records. *Require if records are present*
   5) IFF Data records. *As Applicable*

e) Layer 5. *Conditionally Required*

f) Layers 6 and 7. *Not Defined*

B.5.12.3 Issuance rules

The following additional IFF PDU issuance rules shall apply to a simulation that models a Mode S transponder:

a) *Mode C Altitude*. See B.5.1.2.3 for the issuance rules regarding setting and sending Mode C altitude.

b) *Special Replies*. See B.5.1.2.5 for the issuance rules regarding setting and sending a special reply.

c) *Mode 5 DAP Data*. See B.5.1.2.6 for the issuance rules related to Mode 5/S downlink aircraft parameters.

d) *Test Mode*. See B.5.1.2.7 for the issuance rules related to implementation of the Test Mode.

e) *System Modes*. As a minimum, the following System Mode field values in the System ID record shall be able to be set: Off (1), Standby (2), and Normal/On (3). No Statement (0) shall not be valid for this field.

B.5.12.4 Receipt rules

There are no additional receipt rules for this system type.

B.5.13 System Type 12—Mark XIIA Combined Interrogator/Transponder

B.5.13.1 General

The Mark XIIA Combined Interrogator/Transponder (CIT) is a single system that contains both an interrogator and a transponder. A simulation that implements this system type for either transmit or receive shall adhere to the requirements as stated in this subclause.

The IFF PDU general issuance and receipt rules specified in 5.7.6.3 and 5.7.6.4 shall apply unless superseded by requirements specified herein.

B.5.13.2 Information content

The information content requirements for System Type 6 shall apply when this CIT system is issuing the IFF PDU for the interrogator component and for System Type 9 when the IFF PDU is for the transponder component.

B.5.13.3 Issuance rules

The following additional IFF PDU issuance rules shall apply to a simulation that models a Mark XIIA Combined Transponder/Interrogator:

a) A separate, independent IFF PDU shall be output for the transponder and interrogator as specified in item g) in 5.7.6.1.

b) If this is the transponder part of the system, an IFF PDU when issued shall have the System Type set to 12, the Transponder/Interrogator Indicator in the Change/Options record set to Transponder (0) and the issuance requirements for System Type 9 shall apply. The Change/Options record fields shall be set as specified in Table B.5.

c) If this is the interrogator part of the system, an IFF PDU when issued shall have the System Type set to 12, the Transponder/Interrogator Indicator in the Change/Options record set to Interrogator (1) and the issuance requirements for System Type 6 shall apply. The Change/Options record fields shall be set as specified in Table B.5.

See B.5.1.2.10 for the issuance rules related to active interrogations.

B.5.13.4 Receipt rules

The following additional IFF PDU receipt rules shall apply to a simulation that models a Mark XIIA Combined Transponder/Interrogator:

a) When an IFF PDU is received whose System Type is 12, the PDU shall be processed based on the setting of the Transponder/Interrogator Indicator of the Change/Options record:
   1) If the Transponder/Interrogator Indicator in the Change/Options record is set to Transponder (0), then the IFF PDU shall be considered a System Type 9 transponder.
   2) If the Transponder/Interrogator Indicator in the Change/Options record is set to Interrogator (1), then the IFF PDU shall be considered a System Type 6 interrogator.
b) A Mark XIIA Combined Transponder/Interrogator is capable of interrogating and receiving replies from other system types. The specific system types that it is required to process are specified in Table B.62.
c) See item h) in 5.7.6.1 for transponder data requirements related to an interrogator whose System Status is Off (0).

See B.5.1.2.10 for the receipt rules related to active interrogations.

B.5.14 System Type 13—Mark XII Combined Interrogator/Transponder

B.5.14.1 General

The Mark XII Combined Interrogator/Transponder (CIT) is a single system that contains both a Mark XII interrogator and transponder. A simulation that implements this system type for either transmit or receive shall adhere to the requirements as stated in this subclause.

The IFF PDU general issuance and receipt rules specified in 5.7.6.3 and 5.7.6.4 shall apply unless superseded by requirements specified herein.

B.5.14.2 Information content

The information content requirements for System Type 2 shall apply when this CIT system is issuing the IFF PDU for the interrogator component and for System Type 1 when the IFF PDU is for the transponder component.

B.5.14.3 Issuance rules

The following additional IFF PDU issuance rules shall apply to a simulation that models a Mark XII Combined Transponder/Interrogator:

a) A separate, independent IFF PDU shall be output for the transponder and interrogator as specified in item g) in 5.7.6.1.
b) If this is the transponder part of the system, an IFF PDU when issued shall have the System Type set to 13, the Transponder/Interrogator Indicator in the Change/Options record set to Transponder (0), and the issuance requirements for System Type 1 shall apply. The Change/Options record fields shall be set as specified in Table B.5.
c) If this is the interrogator part of the system, an IFF PDU when issued shall have the System Type set to 13, the Transponder/Interrogator Indicator in the Change/Options record set to Interrogator (1) and the issuance requirements for System Type 2 shall apply. The Change/Options record fields shall be set as specified in Table B.5.

B.5.14.4 Receipt rules

The following additional IFF PDU receipt rules shall apply to a simulation that models a Mark XII Combined Transponder/Interrogator:

a) When an IFF PDU is received whose System Type is 13, the PDU shall be processed based on the setting of the Transponder/Interrogator Indicator of the Change/Options record:
   1) If the Transponder/ Interrogator Indicator in the Change/Options record is set to Transponder (0), then the IFF PDU shall be considered a System Type 1 transponder.
   2) If the Transponder/ Interrogator Indicator in the Change/Options record is set to Interrogator (1), then the IFF PDU shall be considered a System Type 2 interrogator.
b) A Mark XII Combined Transponder/Interrogator is capable of interrogating and receiving replies from other system types. The specific system types that it is required to process are specified in Table B.62.
c) See item h) in 5.7.6.1 for transponder data requirements related to an interrogator whose System Status is Off (0).

B.5.15 System Type 14—TCAS/ACAS Transceiver

B.5.15.1 General

The TCAS/ACAS Transceiver is a single system consisting of a transmitter and a receiver. Depending on whether it is a TCAS/ACAS I or TCAS/ACAS II system, it may respond to an interrogation from another TCAS/ACAS system. As a TCAS/ACAS system is typically associated with an ATCRBS or military transponder, to reduce the quantity of IFF PDUs generated during an exercise, the Layer 1, Fundamental Operational Data record has set aside Parameter 6 to contain the TCAS/ACAS Status record if the entity has a TCAS/ACAS system aboard. In this case, there is no need to transmit a separate IFF PDU with System Type 14 for a TCAS/ACAS Transceiver. A separate IFF PDU with System Type set to 14 may be issued for a TCAS/ACAS Transceiver if high-fidelity data is required, such as to support the interaction between TCAS/ACAS systems and cockpit TCAS/ACAS displays.

The IFF PDU general issuance and receipt rules specified in 5.7.6.3 and 5.7.6.4 shall apply unless superseded by requirements specified herein.

B.5.15.2 Information content

The Information Content requirements for System Type 14 are only specified to the extent that a requirement is applicable to all systems or where specific data applicable to other system types is also applicable to System Type 14. This subclause, and others in Annex B, will be updated to reflect the additional information content requirements for System Type 14 when developed:

a) Layer 1. *Required*
   1) PDU Header
      i) Protocol Version, Exercise ID, PDU Type, Protocol Family, Timestamp, and Length. *Required*
      ii) PDU Status. *Not Applicable*
   2) Emitting Entity ID. *Required*
   3) Event ID. *Optional*
   4) Antenna Location. *Required*
   5) System ID
      i) System Type. *Required*

   ii) System Name. *Required*
   iii) System Mode. *Required*
   iv) Change/Options record. *See B.2.4 for required fields to implement*
6) System Designator. *Required*
7) System-Specific Data. *Not Applicable*
8) Fundamental Operational Data
   i) System Status. *Required*
   ii) Data Field 1. *Available*
   iii) Information Layers. *Required*
   iv) Data Field 2. *Available*
   v) Parameter 1. *Available*
   vi) Parameter 2. *Available*
   vii) Parameter 3. Mode 3/A Record. *Required*
   viii) Parameter 4. *Available*
   ix) Parameter 5. Mode C Altitude record. *Required if has a Mode C capability*
   x) Parameter 6. TCAS/ACAS Status record. *Required*

b) Layer 2. *Optional*
   1) Layer Header. *Required*
   2) Beam Data. *Required*
   3) Secondary Operational Data. *Required*
   4) IFF Fundamental Parameter Data. *All fields and records are required except for the System-Specific Data field*
c) Layer 3. *Not Applicable*
d) Layer 4. *Not Applicable*
e) Layer 5. *Available for IFF data records needed to support high-fidelity TCAS/ACAS systems*
f) Layers 6 and 7. *Not Defined*

B.5.15.3 Issuance rules

There are no unique issuance rules defined for System Type 14.

B.5.15.4 Receipt rules

There are no unique receipt rules defined for System Type 14.

Annex C

(normative)

Radio systems

C.1 Scope

This annex contains the Transmitter and Signal Protocol Data Unit (PDU) information content, associated records, and issuance and receipt requirements related to specific radio systems. (There are no Receiver PDU requirements unique to a specific radio system.) This is a normative annex, and the requirements specified here shall be adhered to as if they were in the body of the standard. They are in an annex because of the amount of information involved and for ease of reference. Table C.1 lists the radio systems currently included in the standard.

Table C.1—Radio systems

Radios listed in the Transmitter PDU—Radio System field
Generic Radio or Simple Intercom
HAVE QUICK (HQ)
HAVE QUICK II (HQII)
HAVE QUICK IIA (HQIIA)
Single Channel Ground and Airborne Radio System (SINCGARS)
Close Combat Tactical Trainer (CCTT) SINCGARS
Enhanced Position Location Reporting System (EPLRS)
Joint Tactical Information Distribution System (JTIDS)/ Multi-Information Distribution System (MIDS)
NOTE—The enumerations for these radios systems are defined in the Radio System field of the Transmitter PDU (see [UID 163]). In the case of two or more radios sharing a single system enumeration, discriminators are defined in this annex to identify the specific radio. Unique requirements for a radio system not included in this annex will be included in a future edition of the standard as they are identified and used in a distributed simulation environment.

The terms *System* and *System Type* are used interchangeably in this annex.

The Transmitter PDU requirements specified in 5.8.3 and the Signal PDU requirements specified in 5.8.4 are also applicable to the radio system types included in this annex and shall be adhered to in addition to the requirements specified herein. If a conflict exists between the requirements specified in 5.8 and those contained herein, the requirements specified herein shall have precedence.

C.2 Generic radio

A generic radio shall be identified by both the Transmitter PDU Modulation Type record—Radio System field indicating a system type of Generic Radio or Simple Intercom (1) and by the Modulation Type record—Major Modulation field set to any nonzero value.

There are no other specific Transmitter, Signal, or Receiver PDU requirements unique to a generic radio.

C.3 Simple intercom

C.3.1 General

This clause defines a Simple Intercom. A Simple Intercom shall be identified by both the Transmitter PDU Modulation Type record—Radio System field indicating a system type of Generic Radio or Simple Intercom (1) and by the Modulation Type record—Major Modulation field set to No Statement (0).

A Simple Intercom uses the radio Transmitter PDU and Signal PDU to implement a simple voice intercom capability using a priority scheme instead of using the PDUs designed specifically to support intercoms (i.e., the Intercom Signal PDU and the Intercom Control PDU).

C.3.2 Transmitter PDU

C.3.2.1 General

Detailed information for a Simple Intercom shall be communicated using the Transmitter PDU as described in the following subclauses.

C.3.2.2 Specific field requirements

Intercom voice communications using the Simple Intercom method shall be implemented by setting the fields of the Transmitter PDU as follows (fields not shown may be set to appropriate values):

a) *Antenna Pattern Type*. This field shall be set to Omni-directional (0).

b) *Frequency*. This field shall represent the Channel Number of the Simple Intercom with valid values being integers in the range 1 to 100 000.

c) *Transmit Frequency Bandwidth*. This field shall be set to zero.

d) *Modulation Type*

   1) Major Modulation. This field shall be set to No Statement (0).

   2) Detail. This field shall be set to the Intercom Priority associated with the transmission. The Intercom Priority (Priority) Range shall be between 0 and 255. The following priority designations shall be used:

      0 = Exempt—Always heard regardless of the highest priority in use
      1 = Highest Priority
      255 = Lowest Priority

e) *Radio System*. This field shall be set to Generic Radio or Simple Intercom (1).

f) *Crypto System*. This field shall be set to No Encryption Device (0).

g) *Crypto Key Identification (ID)*. The Pseudo Crypto Key and Crypto Mode fields shall be set to zero.

h) *Length of Modulation Parameters*. This field shall be set to zero.

C.3.2.3 Issuance rules

A Transmitter PDU shall be issued for a Simple Intercom transmission regardless of the Intercom Priority.

C.3.2.4 Receipt rules

When the Transmitter PDU is associated with a Simple Intercom and the receiving intercom is tuned (set) to the same frequency (channel) as the transmitting intercom, the following rules shall apply:

a) The incoming Signal PDU digital voice data associated with a Transmitter PDU shall be made audible to the operator based on the Detail field of the modulation record (priority) setting as follows:
   1) If the priority = 0, then the voice transmission shall be mixed in and always heard regardless of the highest priority in the range 1 to 255 of any other simultaneous voice transmission by one or more users on the channel.
   2) If a transmission on the channel has a priority in the range of 1 to 255 only, users with the highest priority shall be heard. The transmissions from lower priority users occurring simultaneously shall be muted (i.e., will not be processed and made audible).
   3) Received transmissions shall be continuously monitored, and only the highest priority (plus any priority = 0) transmission shall be heard. This means that someone may be interrupted and no longer heard if a higher priority user begins transmitting on the same channel.
b) Propagation effects shall not be applied to intercom transmissions.
c) All Transmission PDUs received with the same priority as that held by the receiving intercom shall be heard.

C.3.3 Signal PDU

There are no unique Signal PDU requirements for Simple Intercom.

C.4 HAVE QUICK radios

C.4.1 General

HAVE QUICK is a code name for the family of Frequency Hopping (FH) (spread spectrum) schemes used in Ultra-high Frequency (UHF) Amplitude Modulation (AM) tactical radios in the 225–400 MHz spectrum to improve jamming resistance. Radio systems identified as HAVE QUICK in the Radio System field of the Modulation Type record of the Transmitter PDU are covered by the requirements herein. Basic and High-Fidelity options are presented:

a) *Basic Option*. A Basic HAVE QUICK radio is suitable for an exercise that does not require high-fidelity representation of HAVE QUICK radio settings. It is, however, capable of interfacing to other Basic HAVE QUICK radios, as well as to High-Fidelity HAVE QUICK radios. However, it is not able to support the use of actual frequency planning data including the operational NET number and a Multiple Word of Day (MWOD) formatted as described herein. It does not support simulating realistic voice degradation due to time drift and the HAVE QUICK time resynchronization process. In addition, a Basic HAVE QUICK cannot interface directly with a live HAVE QUICK radio in a mixed sim/live HAVE QUICK radio environment.
b) The Basic option shall use the Basic HAVE QUICK Modulation Parameters (MP) record. It shall use an MWOD Index number to support interoperability between Basic HAVE QUICK radios and to allow High-Fidelity HAVE QUICK radios to use the number to index into actual Word-of-Day (WOD) tables.
c) *High-Fidelity Option*. A simulated High-Fidelity HAVE QUICK radio is capable of supporting actual frequency planning data including using the operational NET number and a MWOD formatted as described herein. The High-Fidelity option shall be used if there is a requirement for a

simulated HAVE QUICK radio to interface with a live HAVE QUICK radio or when it is necessary to emulate the HAVE QUICK radio settings contained in actual communications plans.

d) The High-Fidelity option shall use the High-Fidelity HAVE QUICK Variable Transmitter Parameters (VTP) record. The High-Fidelity option shall also require the implementation of the Basic HAVE QUICK MP record to be compatible with Basic simulated HAVE QUICK radios.

e) *PDU Status*. A radio simulation that issues a Transmitter PDU and Signal PDU for a High-Fidelity HAVE QUICK radio shall set the PDU Status field in the PDU Header for these two PDUs as follows:

   1) If the originating radio is a simulated radio, the PDU Status record Live-Virtual-Constructive (LVC) Indicator field (bits 1 to 2) in the PDU header shall be set to Virtual (2) or Constructive (3) whichever is appropriate (see 6.2.67).

   2) If the originating radio is a live radio being forwarded by the radio simulation, the PDU Status record LVC Indicator field (bits 1 to 2) shall be set to Live (1) (see 6.2.67).

C.4.2 Records

C.4.2.1 General

The records associated with the simulation of HAVE QUICK radios are specified herein. Unused and reserved bits in these records shall be set to zero.

C.4.2.2 Basic HAVE QUICK MP record

The Basic option shall implement the Basic HAVE QUICK MP record. This record shall contain the following fields:

a) *NET ID record*. This field shall contain the NET ID being used. The NET ID shall be represented by a NET ID record specified in C.4.2.4. The values used for this record are provided by the exercise agreement.

b) *MWOD Index*. This is an index into the "MWOD Index - WOD Correlation Table" provided by the exercise agreement. When exercise agreement data is not available or MWOD Index is not known, the default maintenance and training MWOD Index value of 1 shall be used, which corresponds to the WOD values shown in Table C.2.

c) *Time of Day*. The Time of Day (TOD) field for the Basic HAVE QUICK MP record is used to specify a representative Time of Day value for a given radio. A value of zero shall be considered a match to any other value received.

Table C.2—MWOD index—WOD correlation table

Basic fidelity HAVE QUICK MP record	High-fidelity HAVE QUICK VTP record Sample communication plan provided by exercise agreement						
MWOD index	**WOD 1**	**WOD 2**	**WOD 3**	**WOD 4**	**WOD 5**	**WOD 6**	**WOD 7**
MAYTAG, NET Number T13.225 Default HAVE QUICK II Maintenance and Training MWOD							
1	300.025	376.000	359.050	314.025	347.050	382.050	301.000
TAC1, NET Number A10.225 TAC2, NET Number A11.225 CAS, NET Number A12.225							

Table C.2—MWOD index—WOD correlation table *(continued)*

Basic fidelity HAVE QUICK MP record	High-fidelity HAVE QUICK VTP record Sample communication plan provided by exercise agreement						
MWOD index	**WOD 1**	**WOD 2**	**WOD 3**	**WOD 4**	**WOD 5**	**WOD 6**	**WOD 7**
10	380.075	335.050	367.025	391.075	312.000	331.025	304.000
11	300.025	371.000	359.075	314.025	384.050	387.075	305.000
12	340.075	376.025	321.075	309.050	396.050	385.025	306.000

NOTE 1—Each MWOD Index value needs to be unique.

NOTE 2—The MWOD Index value of 1 is always assigned the values shown and is always present as the first row in the table.

NOTE 3—WOD7 is used only to establish the calendar day (C/D), also called the OP Day, and is not used in the VTP record and is generally hidden from view of the operator, which enters the calendar day on the radio and the radio selects the appropriate MWOD by comparing the C/D to Segment 7. This is to provide an automatic rollover to the next day's MWOD at 0000 Zulu.

NOTE 4—For this example, TAC1, TAC2, and CAS are on different NET numbers of the same MWOD for three operational calendar days of the, 4th, 5th, and 6th that are contained in WOD7.

NOTE 5—HAVE QUICK radio terminology is confusing and is sometimes specific to the make/model of the HAVE QUICK radio and vehicle in which it is installed. Some users will see WOD numbers called out as SEGMENT numbers, and both terminologies are in common practice.

The format of the Basic HAVE QUICK MP record is shown in Table C.3.

Table C.3—Basic HAVE QUICK MP record

Field size (bits)	Field name	Data type
16	NET ID record	16-bit record
16	MWOD Index	16-bit unsigned integer
16	Reserved	16 bits reserved
8	Reserved	8 bits reserved
8	Reserved	8 bits reserved
32	Time of Day	32-bit unsigned integer
32	Padding	32 bits unused
Total Basic HAVE QUICK MP record size = 128 bits		

C.4.2.3 High-Fidelity HAVE QUICK VTP record

The High-Fidelity option shall implement the High-Fidelity HAVE QUICK VTP record. This record shall contain the following fields:

a) *Record Type*. This field shall be set to the enumeration for the High-Fidelity HAVE QUICK VTP record and shall be represented by a 32-bit enumeration.

b) *Record Length*. This field shall specify the length in octets of this record and shall be set to 40. This field shall be represented by a 16-bit unsigned integer.

c) *TOD Transmit Indicator*. This field shall indicate whether a TOD is being transmitted. This is also referred to as a Mickey. This field shall be represented by an 8-bit enumeration (see [UID 297]).

d) *TOD Delta*. This field shall specify the difference between the transceiver internal clock and the reference clock, which is the amount of clock drift that has occurred. This field is expressed in units of hundreds of microseconds and shall be represented by a 32-bit signed integer. The reference clock is an internally simulated clock that simulates a drift or difference from the reference value of zero, with which all simulated HAVE QUICK radios are assumed to be synchronized. This drift shall be placed in the TOD Delta field of the transmitting radio. Receiving radios may then use the TOD Delta field to emulate audio effects (such as clipped or garbled audio) due to HAVE QUICK time mismatches. If the TOD Delta field is zero, the radio is considered to be synchronized with the reference clock for all of the High-Fidelity HAVE QUICK radios participating in the exercise:

   Example Usage. A TOD Delta value of 30 indicates that the sending transmitter's clock has drifted 3 ms from the reference clock. If the TOD Transmit Indicator is set to zero, it indicates that the associated Signal PDU is not sending a TOD Sync signal.

e) *NET ID record*. This field shall contain the NET ID being used. The NET ID shall be represented by a NET ID record specified in C.4.2.4. The values used for this record are provided by the exercise agreement.

f) *WOD 1 to WOD 6*. Each of these fields shall indicate a WOD. All six WODs collectively are called an MWOD. Each WOD contains six decimal digits. The first digit can only be a 2 or a 3, while the second third and fourth can be 0 through 9. The last two digits can only be 00, 25, 50, or 75. Predefined maintenance and training WODs shall be used when exercise agreement data is not known or not available as shown in Table C.2. WOD notation has a decimal between the third and fourth digit. Example notation: "300.025". WODs are sometimes specified with segments or referred to as segment numbers or CH numbers, and the relationship is as follows:

   WOD 1 = segment 1 = CH20

   WOD 2 = segment 2 = CH19

   WOD 3 = segment 3 = CH18

   WOD 4 = segment 4 = CH17

   WOD 5 = segment 5 = CH16

   WOD 6 = segment 6 = CH15

The format of the High-Fidelity HAVE QUICK VTP record shall be as shown in Table C.4.

Table C.4—High-Fidelity HAVE QUICK VTP record

Field size (bits)	Field name	Data type
32	Record Type = 3000	32-bit enumeration
16	Record Length = 40	16-bit unsigned integer
16	Padding	16 bits unused
16	NET ID record	16-bit record

Table C.4—High-Fidelity HAVE QUICK VTP record *(continued)*

Field size (bits)	Field name	Data type
8	TOD Transmit Indicator	8-bit enumeration
8	Padding	8 bits unused
32	TOD Delta	32-bit signed integer
32	WOD 1	32-bit unsigned integer
32	WOD 2	32-bit unsigned integer
32	WOD 3	32-bit unsigned integer
32	WOD 4	32-bit unsigned integer
32	WOD 5	32-bit unsigned integer
32	WOD 6	32-bit unsigned integer
Total High-Fidelity HAVE QUICK VTP record size = 320 bits		

C.4.2.4 NET ID record

The NET ID for HAVE QUICK radios shall be represented by a NET ID record.

This record shall contain the following fields:

a) *Net Number*. This field shall indicate the net number ranging from 0 to 999 decimal and indicated by a 10-bit unsigned integer.

b) *Frequency Table*. This field shall indicate the Frequency Table portion of the NET ID in use and shall be represented by a 2-bit enumeration (see [UID 299]).

c) *Mode*. This field shall indicate the mode and shall be represented by a 2-bit enumeration (see [UID 298]).

The format of the NET ID record shall be as shown in Table C.5.

Table C.5—NET ID record

Field	Bits	Value
Net Number	0 to 9	Unsigned integer
Frequency Table	10 to 11	Enumeration
Mode	12 to 13	Enumeration
Padding	14 to 15	2 bits unused
Total NET ID record size = 16 bits		

The NET ID record represents an Operational Net in the format of NXX.XYY, where:

N = Mode
XXX = Net Number
YY = Frequency Table

C.4.3 Transmitter PDU

C.4.3.1 General

Detailed transmission characteristics for HAVE QUICK radios shall be communicated using the Transmitter PDU as described herein.

C.4.3.2 Information content

The additional information requirements for the Transmitter PDU specific to a HAVE QUICK radio shall be the inclusion of an appropriate MP and VTP record based on whether the Basic or High-Fidelity option is being modeled. A simulation that only implements the Basic option need only store the local data associated with the Basic HAVE QUICK MP radio to support transmit and receipt rules. A simulation that has implemented the High-Fidelity option shall store all the local data necessary to generate both the Basic HAVE QUICK MP record and the High-Fidelity HAVE QUICK VTP record for transmission and shall use the appropriate data from each to compare received records as specified in C.4.3.4.

C.4.3.3 Issuance rules

C.4.3.3.1 General requirements

The following requirements shall apply to both Basic and High-Fidelity HAVE QUICK radios:

a) A HAVE QUICK radio is in the FH mode when the Modulation Type record: Spread Spectrum field: FH Mode subfield is set to On (1).

b) *Specific Field Requirements*

   The Transmitter PDU field values specific to a HAVE QUICK radio shall be set as follows:

   1) Power. This field shall be set to the radio average effective radiated power. If not known, then this value shall be set to 40 (40 dBm) (10 Watts).

   2) Modulation Type record

      i) Spread Spectrum. When the radio is set to FH mode, then the Frequency Hopping subfield shall be set to On (1); otherwise, this field shall be set to Off (0). The Pseudo Noise subfield shall be set to either On (1) or Off (0), and the Time Hopping subfield shall be set to On (1).

      ii) Major Modulation. This field shall be set to Amplitude (1).

      iii) Detail. This field shall be set to AM (Amplitude Modulation) (2).

      iv) Radio System. This field shall be set to the appropriate enumeration for the type of HAVE QUICK radio system being simulated: HAVE QUICK (2), HAVE QUICK II (3), or HAVE QUICK IIA (4).

   3) Length of Modulation Parameters. This field shall be set to the length of the HAVE QUICK MP record in octets.

   4) Number of Variable Transmitter Parameter Records. This field shall be set to the number of VTP records included.

C.4.3.3.2 Basic Fidelity HAVE QUICK radio

The Basic HAVE QUICK MP record shall be included regardless of the value of the Modulation Type record: Spread Spectrum field.

C.4.3.3.3 High-Fidelity HAVE QUICK radio

Both the High-Fidelity HAVE QUICK VTP record and the Basic HAVE QUICK MP record shall be included regardless of whether the Modulation Type record: Spread Spectrum field indicates that the radio is in the FH mode.

C.4.3.4 Receipt rules

Upon receipt of a Transmitter PDU whose Modulation Type record—Radio System field indicates a HAVE QUICK radio, the following requirements shall be met:

a) When the Transmitter PDU, Modulation Type, Spread Spectrum field, Frequency Hopping flag (bit 0) is set to 1, the Transmitter PDU Frequency and Transmit Frequency Bandwidth fields shall be ignored. A Transmitter PDU that is received for a HAVE QUICK radio, and its associated Signal PDU(s), shall be discarded if any of the following conditions exist:
   1) The receiving simulation has only implemented the Basic option and a Basic HAVE QUICK MP record is not received.
   2) The receiving simulation has implemented the High-Fidelity option, and neither a Basic HAVE QUICK MP record or a High-Fidelity HAVE QUICK VTP record is included in the received Transmitter PDU.
   3) The receiving simulation is in HAVE QUICK I (System 2 HQ) mode and a HAVE QUICK II (System 3) or HAVE QUICK IIa (System 4) Transmitter PDU is received.

b) A check shall be made to see whether the local and received HAVE QUICK radios are both operating in the FH mode:
   1) If both the local and received HAVE QUICK radios are in the FH mode, then it will be further processed as specified in item i) or item ii) below and depending on whether the local HAVE QUICK radio has implemented the Basic or High-Fidelity HAVE QUICK option:
      i) If one is in the FH mode and the other is not, then the audio or data contained in the associated Signal PDU(s) shall be either discarded and no voice or other data processed or the simulation of the intelligible effects of such a mismatch may be modeled.
      ii) If both radios are in the non-FH mode, then the associated Signal PDU shall be processed as a normal radio.

c) If the receiving simulation has only implemented the Basic option, the following requirements shall apply:
   1) The contents of the NET ID, MWOD Index and Time of Day fields of the Basic HAVE QUICK MP record shall be compared with the values of the local HAVE QUICK radio:
      i) If all three field values match the same local field values, then the associated Signal PDUs voice or other data shall be audible or otherwise processed.
      ii) If any of the three field values do not match the same local field values, then the associated Signal PDUs shall be either discarded and no voice or other data processed or the emulation of the intelligible effects of such a mismatch may be modeled.
   2) Any High-Fidelity HAVE QUICK VTP record received shall be ignored.

d) If the receiving simulation has implemented the High-Fidelity option, the following requirements shall apply:
   1) If only the Basic HAVE QUICK MP record is received, the receiving simulation shall use this record and process it as specified in item c1) in C.4.3.4.

2) If the High-Fidelity HAVE QUICK VTP record is present, the Basic HAVE QUICK MP record shall be ignored and the High-Fidelity HAVE QUICK VTP record shall be processed as follows:
   i) TOD Delta. The applications shall compute the difference between the received TOD delta and local TOD delta fields and model the effects of the mismatch. If the difference is greater than HQ_TOD_DIFF_THRSH, the audio or data contained in the associated Signal PDU(s) shall be either discarded and no voice or other data processed or the simulation of the intelligible effects of such a mismatch may be modeled.
   ii) NET ID Record. If any field value does not match the corresponding local field value, then the associated Signal PDUs shall be either discarded and no voice or other data shall be processed or the intelligible effects, if any, that could result from the mismatches may be modeled.
   iii) WOD 1 through WOD 6. If any WOD field does not match the local WOD field, then the associated Signal PDUs shall be either discarded and no voice or other data shall be processed or the intelligible effects, if any, that could result from the mismatches may be modeled.
3) The requirements related to modeling effects are beyond the scope of this standard.

C.4.3.5 Manual TOD synchronization

C.4.3.5.1 General

A simulation that has implemented the High-Fidelity option shall have the capability to perform a Manual TOD Synchronization (Sync), a procedure referred to as a "Mickey." A simulation that has only implemented the Basic option may not perform a Manual TOD Synchronization. The issuance and receipt rules for the Mickey are described in the following subclauses.

C.4.3.5.2 Manual TOD synchronization issuance rules

A simulation originating a *Manual TOD Sync* shall comply with the following requirements:

a) A manual TOD sync transmission shall consist of an issuance of a transmitter PDU to indicate a radio transmission, and the TOD Transmit Indicator field transitioning from zero to one followed by Signal PDUs containing the audio of a "Mickey" tone, followed by a Transmitter PDU to indicate the TOD Transmit Indicator field transitioning from one to zero. The TOD Transmit Indicator field shall not transition from one to zero until after the audio of a "Mickey" tone has been transmitted. Source data for the Mickey is normally obtained from a recording of a live "Mickey."
b) A manual TOD sync transmission shall only be issued when the Transmitter PDU: Modulation Type record: Spread Spectrum field: FH Mode subfield is set to Off (0) and the Transmitter PDU: Transmit State field is set to On and Transmitting (2).
c) The manual TOD sync transmission may be part of a normal voice communications transmission or as a separate transmission specifically for manual TOD sync.

C.4.3.5.3 Manual TOD synchronization receipt rules

A simulation that has implemented the High-Fidelity option shall process a Manual TOD Sync according to the following requirements:

a) Upon reception of a High-Fidelity HAVE QUICK Transmitter PDU and the High-Fidelity HAVE QUICK VTP record: TOD Transmit Indicator field is set to TOD Transmission in Progress (1), and the application is set to receive a manual TOD sync and the Transmitter PDU: Modulation Type record: Spread Spectrum field: FH Mode subfield is set to Off (0), the value of the internal clock value described in item d) of C.4.2.3 shall be set to the value of the received Transmitter PDU: TOD delta field upon completion of the received TOD sync transmission.

b) The successful reception of a manual TOD sync, TOD Delta value depends on, among other things, a receiver model, which is beyond the scope of this standard.

C.4.3.6 Mixed basic and high-fidelity HAVE QUICK radio environment correlation

When low- and high-fidelity HAVE QUICK radios are present in an exercise, High-Fidelity HAVE QUICK radios shall process a Basic HAVE QUICK MP record and correlate it with the High-Fidelity HAVE QUICK VTP record values. Table C.6 shows the relationship among the fields in these two records.

Table C.6—Basic/High-Fidelity HAVE QUICK correlation

Basic HAVE QUICK MP record		High-fidelity HAVE QUICK VTP record		Process C.4.3.6
Field size (bits)	**Field name**	**Field name**	**Field size (bits)**	
		Record Type	32	N/A
		Record Length	16	N/A
16	NET ID record	NET ID record	16	Item a) below
16	MWOD Index			Item c) below
		WOD 1 = Segment 1 = CH20	32	—
		WOD 2 = Segment 2 = CH19	32	—
		WOD 3 = Segment 3 = CH18	32	—
		WOD 4 = Segment 4 = CH17	32	—
		WOD 5 = Segment 5 = CH15	32	—
		WOD 6 = Segment 6 = CH16	32	—
		TOD Transmit Indicator		
16	Reserved			N/A
8	Reserved			N/A
8	Reserved			N/A
32	Time of Day	TOD Delta	32	N/A

This correlation shall be accomplished as follows

a) *NET ID record*. The NET ID record of the received Basic HAVE QUICK MP record (see Table C.3) shall be compared with the High-Fidelity NET ID record. If there is a match, proceed to item b) below. If there is no match, then the associated Signal PDUs shall be either discarded and no voice or other data processed or the simulation of the intelligible effects of such a mismatch may be modeled.

b) *Time of Day*. The Time of Day field of the Basic Fidelity HAVE QUICK MP record shall be ignored by the High-Fidelity HAVE QUICK radio. Basic Fidelity HAVE QUICK radios shall always be considered to be in HAVE QUICK time sync with High-Fidelity radios.

c) *MWOD Index*. The MWOD Index value from the Basic Fidelity HAVE QUICK MP record shall be used as an index into the MWOD Index WOD Correlation Table (see Table C.2). If the received MWOD Index value matches an MWOD Index number in the table, then the MWOD and WOD values shall be considered correlated. If there is no match, then the associated Signal PDUs shall be either discarded and no voice or other data processed or the emulation of the intelligible effects of such a mismatch may be modeled.

d) *Reserved Fields*. All reserved fields of the Basic Fidelity HAVE QUICK MP record shall be ignored.

C.4.3.7 Mixed sim/live HAVE QUICK radio environment

C.4.3.7.1 General

Live HAVE QUICK radios may be interfaced to simulated HAVE QUICK radios if there is a simulated High-Fidelity HAVE QUICK radio present that has the capability to interface to them. Basic HAVE QUICK Radios may interface with live radios through a High-Fidelity HAVE QUICK radio if that radio has such a capability. There is no requirement in this standard for a High-Fidelity HAVE QUICK radio to be able to interface to a live HAVE QUICK radio or to be capable of interfacing live HAVE QUICK radios to other simulated HAVE QUICK radios. Figure C.1 shows an example of a mixed sim/live HAVE QUICK radio environment.

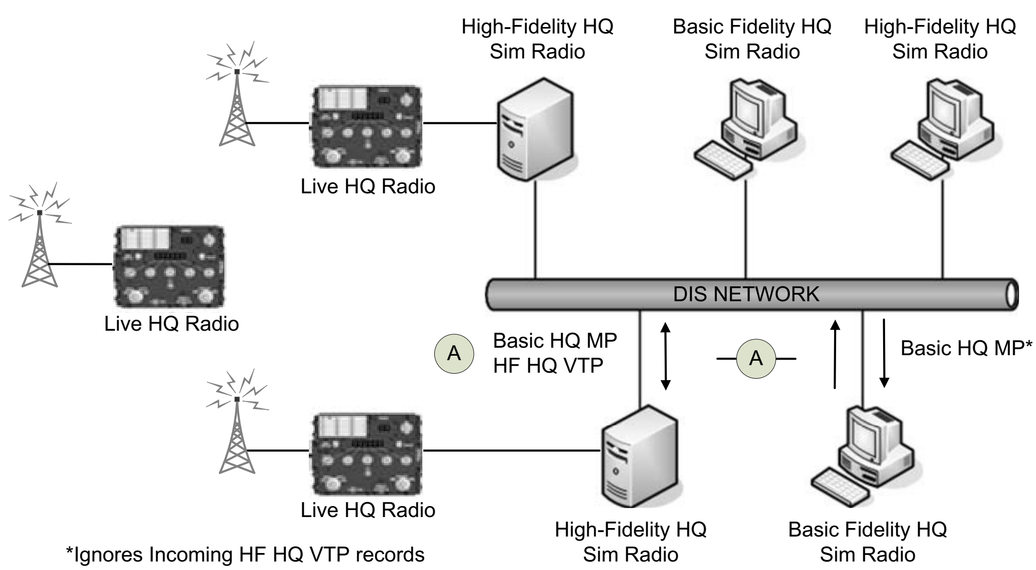


Figure C.1—Mixed live/simulation HAVE QUICK environment

C.4.3.7.2 Forwarding live and sim HAVE QUICK radio transmissions

If a High-Fidelity HAVE QUICK radio has the capability to interface to a live HAVE QUICK radio and also has the capability to forward transmissions between the live HAVE QUICK radios and the simulated HAVE QUICK radios, the following requirements shall be met:

a) All live HAVE QUICK transmissions shall be forwarded onto the Distributed Interactive Simulation (DIS) network as Transmitter and Signal PDUs.

b) All received Basic and High-Fidelity HAVE QUICK Transmitter and Signal PDUs shall be forwarded onto the live HAVE QUICK radio net, except the Basic Fidelity HAVE QUICK Radio

Start Transmitter PDU and associated Signal PDU and End Transmitter PDU shall be discarded if either of the following conditions exists:

1) The MWOD Index contained in the received Start Transmitter PDU does not match an MWOD Index value in the receiving simulation's MWOD—WOD Correlation Table.
2) The fields of the NET ID record do not match the corresponding live HAVE QUICK Radio NET ID values.

C.4.4 Signal PDU

There are no unique Signal PDU requirements for HAVE QUICK radios except that a "Mickey' tone shall be included in the Signal PDU when it is a *Manual TOD Sync* transmission.

C.5 SINCGARS radio

The SINCGARS (Single Channel Ground and Airborne Radio System) radio is a reliable, secure, frequency hopping radio that can be used to transmit both voice and data including an interface to support transmission of EPLRS data. It comes in various configurations including vehicle-mounted, backpack, airborne, and handheld versions. It is a Combat Net Radio widely used by all the U.S. services and allied forces to support combat operations including the air-to-ground mission.

A SINCGARS radio shall be identified by the Radio System field of the Transmitter PDU Modulation Type record indicating a SINCGARS radio other than a CCTT SINCGARS (6) radio. There are no specific requirements currently defined for a SINCGARS radio.

C.6 CCTT SINCGARS radio

C.6.1 General

The CCTT SINCGARS radio is the implementation of a SINCGARS radio simulation by the U.S. Army Close Combat Tactics Trainer (CCTT) system. This has become the de facto standard for the functional fidelity needed to support training exercises that do not require a higher fidelity representation of that radio's operation.

A CCTT SINCGARS radio shall be identified by the Radio System field of the Transmitter PDU Modulation Type record being set to a CCTT SINCGARS (6) radio.

C.6.2 Transmitter PDU

C.6.2.1 General

Detailed information for CCTT SINCGARS radios shall be communicated by using the Transmitter PDU and the CCTT SINCGARS Modulation Parameters (MP) record as described in the following subclauses.

C.6.2.2 Information content

There are no additional information requirements specific to a CCTT SINCGARS radio except for the requirement to use the CCTT SINCGARS MP record.

C.6.2.3 Modulation Parameters record

CCTT SINCGAR radios shall use the CCTT SINCGARS MP record. It shall contain the following fields:

a) *FH Net ID*. This field shall identify the frequency hopping network identity.

b) *Hop Set ID*. This field shall identify the set of frequencies used when creating a hopping pattern. The frequencies in the hop set are transmitted by a method outside the scope of this standard.

c) *Lockout Set ID*. This field shall identify the set of frequencies that are excluded from the hopping pattern. The frequencies in the lockout set are transmitted by a method outside the scope of this standard.

d) *Start of Message*. This field shall specify whether the radio is starting or continuing a transmission. A CCTT SINCGARS radio shall not receive any part of an FH transmission if the Start of Message is missed. Start of message may be missed if the radio was transmitting, receiving a different radio, or otherwise not ready to start receiving a transmission. This field shall be represented by an 8-bit enumeration (see [UID 170]).

e) *Clear Channel*. This field shall specify that the transmission is not subject to propagation loss, interference, COMSEC, or any other form of signal degradation. Clear channels are used for interoperability with older systems and for control purposes. This field shall be represented by an 8-bit enumeration (see [UID 171]).

f) *FH Synchronization Time Offset*. This field shall identify the offset to exercise time in seconds for the clock in the CCTT SINCGARS radio.

   Note that actual SINCGARS radio clocks have to be synchronized for the radios to communicate. If no synchronization simulation is required, all bits of this field shall be set to zero.

g) *Transmission Security Key*. This field shall identify the transmission security key that is used when generating the hopping pattern. It shall be set to zero if no key value is included.

The format of the CCTT SINCGARS MP record shall be as shown in Table C.7.

Table C.7—CCTT SINCGARS MP record

Field size (bits)	Field name	Data type
16	FH Net ID	16-bit unsigned integer
16	Hop Set ID	16-bit unsigned integer
16	Lockout Set ID	16-bit unsigned integer
8	Start of Message	8-bit enumeration
8	Clear Channel	8-bit enumeration
32	FH Synchronization Time Offset	32-bit unsigned integer
16	Transmission Security Key	16-bit unsigned integer
16	Padding	16 bits unused
Total CCTT SINCGARS MP record size = 128 bits		

C.6.2.4 Specific field requirements

The Transmitter PDU field values that are specific to CCTT SINCGARS radios shall be set as follows:

a) *Frequency*. When in frequency hopping mode, the frequency shall be set to 312 500 000 Hertz (312.5 MHz). If the radio is not in frequency hopping mode, then the frequency shall be set to the radio transmit frequency.

b) *Transmit Frequency Bandwidth*. This field shall be set to the simulated radio bandwidth. If not known, then this value shall be set to 25 000 (25 kHz).

c) *Power*. This field shall be set to the radio average effective radiated power. If not known, then this value shall be set to 40 (40 dBm).

d) *Modulation Type Record*

   1) Spread Spectrum. When the radio is set to frequency hopping mode, then the Frequency Hopping subfield shall be set to On (1). All other subfields shall be set to Off (0).

   2) Major Modulation. This field shall be set to Amplitude and Angle (2).

   3) Detail. This field shall be set to Amplitude and Angle (1) for Major Modulation 2.

   4) Radio System. This field shall be set to CCTT SINCGARS (6).

e) *Length of Modulation Parameters*. This field shall be set to the length of the CCTT SINCGARS MP record in octets.

C.6.2.5 Issuance rules

A CCTT SINCGARS MP record shall always be included in the Transmitter PDU.

C.6.2.6 Receipt rules

Upon receipt of a Transmitter PDU whose Modulation Type record—Radio System field is set to CCTT SINCGARS (6), the field values of the CCTT SINCGARS MP record, excluding the FH Synchronization Time Offset field, shall be compared with the values for those fields stored for the CCTT SINCGARS radio at the receiving simulation:

a) If all fields match the same local field values, then the associated Signal PDUs shall be processed.

b) If all fields do not match the same local field values, then the associated Signal PDUs shall be discarded and no voice or other data shall be audible.

C.6.3 Signal PDU

There are no unique Signal PDU requirements for CCTT SINCGARS radios.

C.7 EPLRS

The Enhanced Position Location Reporting System (EPLRS) is a digital radio that provides secure, electronic warfare-resistant data communications primarily in support of the U.S. Army Battle Command System. Major components of the EPLRS system are the EPLRS radio and Net Control Station, which establishes and controls the network of individual radios. Each real-world radio in the network has unique real-world time slots during which it can transmit to both Net Control Station and other radios. Operational data to support situation awareness and specific combat missions are exchanged including position, navigation, and identification information. United States Air Force F-16 and A-10 fighter aircraft are equipped with the Situational Awareness Data Link (SADL), which uses EPLRS as the digital link to the Army in support of various air-to-ground missions.

An EPLRS radio shall be identified by the Modulation Type record—Radio System field of the Transmitter PDU indicating an EPLRS (7) radio.

C.8 JTIDS/MIDS radio

C.8.1 General

The Joint Tactical Information Distribution System (JTIDS) and Multi-Information Distribution System (MIDS), which is a smaller, improved version of JTIDS, are used to exchange Link-16 data link information. Link-16 is a major joint and allied forces tactical data link that supports battlefield situation awareness and combat operations.

The requirements specified in the Simulation Interoperability Standards Organization (SISO) Link-16 Simulation Standard (SISO-STD-002) shall be used in conjunction with the requirements specified herein to simulate Link-16 communications using the Transmitter and Signal PDUs. In the event that the requirements specified in SISO-STD-002 conflict with those contained in this standard, this standard shall be considered superseding.

The enumerations associated with JTIDS/MIDS radios shall be as contained in SISO-REF-010. In the event that the enumeration requirements specified in SISO-STD-002 conflict with those contained in SISO-REF-010, the SISO-REF-010 document shall be considered superseding.

C.8.2 Transmitter PDU

C.8.2.1 General

All Transmitter PDU information requirements, issuance rules, and receipt rules for radios specified in 5.8 and 7.7 of this standard shall also apply to JTIDS/MIDS radios.

C.8.2.2 Information content

Unique JTIDS/MIDS radio information content specific to Transmitter PDU fields, subfields, and records shall be as defined in the SISO Link-16 Simulation Standard (SISO-STD-002).

C.8.2.3 Modulation Parameters records

JTIDS/MIDS radios shall use the JTIDS/MIDS Modulation Parameters (MP) records specified in the SISO Link-16 Simulation Standard (SISO-STD-002). Multiple JTIDS/MIDS MP records, each with unique information content requirements, are defined in SISO-STD-002.

C.8.2.4 Issuance rules

Unique Transmitter PDU issuance rules for JTIDS/MIDS radios shall be as specified in the SISO Link-16 Simulation Standard (SISO-STD-002).

C.8.2.5 Receipt rules

Unique Transmitter PDU receipt rules for JTIDS/MIDS radios shall be as specified in the SISO Link-16 Simulation Standard (SISO-STD-002).

C.8.3 Signal PDU

The information content, issuance rules, and receipt rules unique to JTIDS/MIDS radios shall be as defined in the SISO Link-16 Simulation Standard (SISO-STD-002). All Signal PDU information requirements, issuance rules, and receipt rules for radios specified in this standard shall also be applicable to JTIDS/MIDS radios.

Annex D

(normative)

Objects

D.1 Scope

This annex defines the characteristics and interactions of objects and object identifiers.

D.2 Object types

D.2.1 Introduction

The term "object" is used to define a physical object in the synthetic battlespace that may be uniquely referenced using an object identifier. The most common type of object is an entity. Entities are uniquely designated by an Entity Identifier. Other objects may define the synthetic environment that makes up the battlespace including characteristics of terrain, water, the atmosphere, and space.

D.2.2 Objects

D.2.2.1 General

This subclause defines all the objects that may be present in a Distributed Interactive Simulation (DIS) synthetic environment.

D.2.2.2 Object types and identifiers

Table D.1 lists objects and their associated identifiers and Protocol Data Units (PDUs).

Table D.1—Objects

***Object type/*identifier**	**PDUs**	**Description**
Entity Entity Identification (ID)	Entity State PDU Entity State Update PDU	Represents a physical object in the exercise, which is usually capable of interacting with other entities. May also include cultural features and environmental objects without specific dimensions. Examples: tanks, aircraft, ships, soldiers, munitions, satellites, building, and bridge.
Point Object Object ID	Point Object State PDU	Transmits specific subset of information about environment objects described by a point. Examples: vehicle defilades, trenches, log cribs, abatis, craters, ribbon bridges, rubble, armored vehicle launched bridges, stationary bridges, destructible buildings, and smoke.

Table D.1—Objects *(continued)*

***Object type/*identifier**	**PDUs**	**Description**
Linear Object Object ID	Linear Object State PDU	An object that is geometrically anchored to the terrain with one point and has a segment size and orientation represented by lines, which may be dynamically changed during an event. Transmits specific subset of information about environment objects described by lines. Examples: ditches, wire obstacles, minefield lane markers, roads, train tracks, and canals.
Areal Object Object ID	Areal Object State PDU	Transmits specific subset of information about environment objects described by areas. Examples: lakes and city boundaries.
Minefield Minefield ID	Minefield State PDU	Minefield State PDU: Defines area and mine characteristics without specifying the location of individual mines.
Mine Entity Mine Entity ID	Minefield Data PDU	Defines area and mine characteristics including the location of individual mines.
Aggregate Aggregate ID	Aggregate State PDU	Defines an aggregate, subaggregate and lists any entities for which Entity State PDUs are being output that are part of the aggregation.
Environmental Process Environmental Process ID	Environmental Process PDU	Communicates simple environment variables, small-scale environmental updates, and embedded processes. An embedded process is an environmental effect instigated by the action of an entity that may continue after that entity has left the battlespace. (Alternative to using Linear, Point and Areal objects.) Examples: contrails, smoke, obscurants, chaff, dust clouds, and toxic chemicals.
Group Group ID	IsGroupOf PDU	Contains a list of entities. Used to conserve bandwidth.
Gridded Data Gridded Data ID	Gridded Data PDU	Transmits information about large-scale or high-fidelity spatially and temporally varying ambient fields and about natural and man-made environmental effects and features. Examples: land and subsurface terrain.

D.2.2.3 Live objects

Live objects represent real-world entities such as live soldiers on a training range, live aircraft, and live operational equipment and systems such as a fielded military system like PATRIOT or a Navy ship in port. These live objects may be represented in the synthetic battlespace using PDUs from the DIS Live Entity Interaction/Protocol Family and other applicable PDUs or may be represented using the Entity State and other PDUs normally used for virtual and constructive simulated objects. When a PDU is used to portray a live object or associated independent attributes or interactions and the PDU is not from the Live Entity Interaction/Protocol Family, the PDU Status record Live-Virtual-Constructive (LVC) Indicator field (bits 1 to 2) in the PDU header shall be set to Live (1).

D.2.2.4 Virtual objects

Virtual objects emulate real-world entities such as live soldiers on a training range, live aircraft, and live operational equipment and systems such as a fielded military system like PATRIOT or a Navy ship in port. They are usually referred to as virtual simulators. These virtual simulators include physical characteristics of the real-world entity, as well as equipment and computer functionality. In some cases, actual hardware and software from the real-world system is used. Instead of including the complete system and the physical environment it is in, it may replicate just the crew stations and compartment the stations are in. All virtual simulators have one or more humans-in-the loop to operate them. Examples of virtual simulators include aircraft cockpit simulators, Navy mock-ups of ship Combat Information Centers that are constructed in buildings on land, tank simulators, and Command, Control, Communications, Computers, Intelligence, Surveillance and Reconnaissance (C4ISR) system workstations arranged as in the real-world but in a room in a facility. Virtual simulators are presented in the synthetic battlespace as entities, and Entity State PDUs as well as other PDUs to represent their sensor and weapons capabilities are issued. When a PDU is used to portray a virtual simulator or associated independent attributes or interactions, the PDU Status record LVC Indicator field (bits 1 to 2) in the PDU header shall be set to Virtual (2).

D.2.2.5 Constructive objects

Constructive objects simulate real-world entities such as live soldiers on a training range, live aircraft, and live operational equipment and systems such as a fielded military system like PATRIOT or a Navy ship in port. They also simulate atmospheric, terrain, and water environments. They are usually referred to as constructive simulations. These constructive simulations do not replicate actual displays, workstations, and physical characteristics of the live entity they are simulating, but they provide the functionality of the real system through automatic behavior and manual operator intervention. Constructive simulations may have preprogrammed movements and dynamic automatic interactive behaviors. Most constructive simulations also have a human-in-the loop that can override automatic behavior and initiate and control constructive simulation actions. Constructive simulations allow a large number of entities to be simulated with simulation operators monitoring and controlling a number of the entities. Examples included simulated pilots acting as real pilots for a number of aircraft and responding to operational unit voice and digital data link commands, and ground unit simulated personnel acting as platoons, battalion, and high-echelon movements of simulated personnel and equipment at simulation workstations. The entities and their interactions portrayed by constructive simulations in the synthetic battlespace issue Entity State PDUs as well as other PDUs to represent their sensor and weapons capabilities. When a PDU is used to portray a constructive simulation or associated independent attributes or interactions, the PDU Status record LVC Indicator field (bits 1 to 2) in the PDU header shall be set to Constructive (3).

D.2.3 Primary and secondary object IDs

A PDU may include one or more fields that contain entity or other object identifiers, or it may have no fields that can contain an entity or other object identifiers. So that requirements related to the setting and processing of object ID fields in a PDU are able to be clearly stated, the terms *primary* and *secondary* are used to differentiate between the two categories of object IDs. See 4.2.5.2 for the general requirements associated with object identifiers. Not all PDUs are related to conveying information about a specific object so some PDUs may not include a field that contains a primary or secondary object identifier:

a) *Primary Object ID*. A *primary* object identifier is an object identifier (e.g., Entity ID and Object ID) that identifies the main object (e.g., entity or point) that is the subject of the information contained in the PDU. As an example, the Firing Entity ID is the *primary* object ID in the Fire PDU. The *primary* object ID, when present in a PDU, is a fixed field that is always present. It uses the 48-bit identifier format (site, application and reference number).

b) *Secondary Object ID*. A *secondary* object identifier is an object identifier (e.g., Entity ID and Object ID) that identifies one or more other objects in addition to the *primary* object ID contained in the PDU. A *secondary* object ID field may be included as a fixed field in a PDU (e.g., the Target Object

ID of the Fire PDU) or may only appear in a record that may or may not always be present (e.g., an Identification Friend or Foe (IFF) Data record in the IFF PDU).

c) *Simulation Management PDUs*. There are Simulation Management (SIMAN) PDUs and Simulation Management with Reliability (SIMAN-R) PDUs. All SIMAN and SIMAN-R PDUs have Originating ID and Receiving ID fields, either of which can contain an entity ID (see 5.6.2.3 and 5.6.2.4). They are not considered to be either primary or secondary object IDs, even if they identify objects.

A simulation shall discard a PDU that contains an invalid object identifier for a primary object identifier. It may ignore an invalid object identifier for a secondary object identifier if it can still process the PDU without adversely affecting its simulation; otherwise, it shall discard the PDU. See 4.2.5.8 for the definition of valid object identifiers.

Table D.2 lists for each PDU whether any *primary* or *secondary* object IDs are associated with it. The PDUs are sorted alphabetically. This table is not intended to convey information about simulations as a simulation ID or simulation address does not represent an entity or other object. Originating and Receiving ID fields, for example, may contain an entity ID, a Simulation ID, or some other ID.

Table D.2—Primary and secondary object IDs

PDU	Primary object ID field	Secondary object ID fields
Acknowledge	None	None
Acknowledge-R	None	None
Action Request	None	*Entity/Object IDs may be included in Datum records.*
Action Request-R	None	*Entity/Object IDs may be included in Datum records.*
Action Response	None	*Entity/Object IDs may be included in Datum records.*
Action Response-R	None	*Entity/Object IDs may be included in Datum records.*
Aggregate State	None	Entity IDs—*Entity IDs*
Appearance	Live Entity ID	None
Areal Object State	Object ID	Referenced Object ID
Articulated Parts	Live Entity ID	None
Attribute	Entity/Object ID	*Entity/Object IDs may be included in Attribute records.*
Collision	Issuing Entity ID	Colliding Entity ID
Collision-Elastic	Issuing Entity ID	Colliding Entity ID
Comment	None	*Entity/Object IDs may be included in Datum records.*
Comment-R	None	*Entity/Object IDs may be included in Datum records.*
Create Entity	Non	None

Table D.2—Primary and secondary object IDs *(continued)*

PDU	Primary object ID field	Secondary object ID fields
Create Entity-R	None	None
Data	None	*Entity/Object IDs may be included in Datum records.*
Data Query	None	None
Data Query-R	None	None
Data-R	None	*Entity/Object IDs may be included in Datum records.*
Designator	Designating Entity ID	Designated Object ID
Detonation	Source Entity ID	Target Entity ID Exploding Entity ID *Entity and Object IDs may also be contained in Variable Parameter records*
Directed Energy (DE) Fire	Firing Entity ID	*Entity and Object IDs may be contained in directed energy records.*
Electromagnetic Emission (EE)	Emitting Entity ID	Track/Jam Data—*Entity ID*
Entity Damage Status	Damaged Entity ID	*Entity and Object IDs may be contained in Entity Damage records.*
Entity State	Entity ID	*Entity and Object IDs may be contained in the Variable Parameter records.*
Entity State Update	Entity ID	*Entity and Object IDs may be contained in the Variable Parameter records.*
Environmental Process	Environmental Process ID	*Entity and Object IDs may be contained in the Environment State records.*
Event Report	None	None
Event Report-R	None	None
Fire	Firing Entity ID	Target Entity ID Munition/Expendable Entity ID
Gridded Data	None	None
IFF	Emitting Entity ID	*Entity and Object IDs may be contained in IFF Data records.*
Information Operations (IO) Action	IO Attacker Entity ID	IO Primary Target Entity ID *Entity and Object IDs may be contained in IO Data records.*
IO Report	IO Attacker Entity ID	Primary Target Entity ID *Entity and Object IDs may be contained in IO Data records.*
Intercom Control	Source Entity ID	Master Entity ID
Intercom Signal	Entity ID	None
IsGroupOf	None	Group Entity IDs
IsPartOf	None	Receiving Entity ID

Table D.2—Primary and secondary object IDs *(continued)*

PDU	Primary object ID field	Secondary object ID fields
Live Entity (LE) Detonation	Firing Live Entity ID	Target Live Entity ID Munition Live Entity ID
LE Fire	Firing Live Entity ID	Target Live Entity ID Munition Live Entity ID
Linear Object State	Object ID	Referenced Object ID
Minefield Data	Minefield ID	Mine Entity ID
Minefield Query	Minefield ID	None
Minefield Response Negative Acknowledgment (NACK)	Minefield ID	None
Minefield State	Minefield ID	None
Point Object State	Object ID	Referenced Object ID
Receiver	Entity ID	Transmitter Radio Reference ID—*Entity ID*
Record Query-R	None	None
Record-R	None	None
Remove Entity	None	None
Remove Entity-R	None	None
Repair Complete	Repairing Entity ID	Receiving Entity ID
Repair Response	Receiving Entity ID	Repairing Entity ID
Resupply Canceled	Supplying Entity ID or Receiving Entity ID	Supplying Entity ID or Receiving Entity ID
Resupply Offer	Supplying Entity ID	Receiving Entity ID
Resupply Received	Receiving Entity ID	Supplying Entity ID
Supplemental Emission/Enitity State (SEES)	Originating Entity ID	None
Service Request	Requesting Entity ID	Servicing Entity ID
Set Data	None	*Entity/Object IDs may be included in Datum records.*
Set Data-R	None	*Entity/Object IDs may be included in Datum records.*
Set Record-R	None	*Entity/Object IDs may be included in Datum records.*
Signal	Radio Reference ID—*Entity ID or Object ID*	None
Start/Resume	None	None
Start/Resume-R	None	None
Stop/Freeze	None	None
Stop/Freeze-R	None	None

Table D.2—Primary and secondary object IDs *(continued)*

PDU	Primary object ID field	Secondary object ID fields
Transfer Ownership (TO)	Transfer Entity ID	*Entity and Object IDs may be contained in Datum records.*
Transmitter	Radio Reference ID—*Entity ID or Object ID*	*Entity and Object IDs may be contained in VTP records.*
Time Space Position Information (TSPI)	Live Entity ID	None
Underwater Acoustic (UA)	Emitting Entity ID	None

D.2.4 Object interactions

Some objects may interact with each other, and some may not. The degree and conditions under which objects may interact as reflected in their presence in secondary object ID fields of PDUs shall be as specified in Table D.3.

Table D.3—Object interactions

PDU	PDU field	Object IDs eligible to be contained in this field
Fire	Target ID	All Object IDs Minefield Entity ID Aggregate Entity ID Group Entity ID
	Location in Object's Coordinates	All Target IDs except a Linear or Gridded Data Object. The *Location in Object's Coordinates* field for these two objects shall be set to 0.0, 0.0, 0.0 (all bits set to zero).
Detonation	Target ID	All Object IDs Minefield Entity ID Aggregate Entity ID Group Entity ID
	Location in Object's Coordinates	All Target IDs except a Linear or Gridded Data Object. The *Location in Object's Coordinates* field for these two objects shall be set to 0.0, 0.0, 0.0 (all bits set to zero).
Designator	Designated ID	All Object IDs Minefield Entity ID Aggregate Entity ID Group Entity ID
	Designator Spot with Respect to Designated Entity	All Designated IDs except a Linear or Gridded Data Object. The *Location in Object's Coordinates* field for these two objects shall be set to 0.0, 0.0, 0.0 (all bits set to zero).

Table D.3—Object interactions *(continued)*

EE	Track/Jam Data	Entity ID, Aggregate Entity ID and Group Entity ID only.
Collision-Elastic	Colliding Object ID	Entity ID, Mine Entity ID, Aggregate Entity ID and Group Entity ID only.
Collision	Colliding Object ID	Entity ID, Mine Entity ID, Aggregate Entity ID and Group Entity ID only.

Annex E

(normative)

Dead reckoning definitions and algorithms

E.1 Scope

This annex defines terms associated with dead reckoning and a naming methodology to distinguish between dead reckoning algorithms. This annex also includes the specific algorithms required by this standard. In this annex, any reference to an Entity State Protocol Data Unit (PDU) may also be applied to an Entity State Update PDU as appropriate.

E.2 Description of dead reckoning

A method of position/orientation estimation called dead reckoning shall be employed to limit the rate at which Entity State PDUs are issued. By estimating the position/orientation of other entities, it is not necessary to receive a report about every change in position and orientation that occurs in the entity's trajectory. Only when a change in position and orientation differs by a prespecified amount (threshold) from the dead reckoned position and orientation is a new update required.

E.3 Dead reckoning and the issuing entity

Each simulation application shall maintain an internal model of itself (representing its actual position and orientation) and a dead reckoned model of itself. Thresholds for position and orientation shall be established as criteria for determining whether the entity's actual position/orientation has varied from the dead reckoned model. When the entity's actual position and orientation have varied from the dead reckoned position and orientation by more than the threshold value, the entity shall issue an Entity State PDU to communicate to the other simulation applications the actual position and orientation. The entity shall also use this same information communicated to other simulation applications to update the dead reckoned model of itself.

E.4 Dead reckoning and the receiving entity

Each simulation application shall also maintain a dead reckoned model of the position of entities that are of interest (within sight or range). An entity shall also dead reckon the orientation of other entities when specified by the dead reckoning model in use. The dead reckoned position/orientation of other entities shall be used to display their position/orientation in an entity's visual or sensor displays. When an entity receives a new update from one of the entities that it is dead reckoning, then it shall correct its dead reckoning model and base its estimation on the most recent position, velocity, and acceleration information. Smoothing techniques may be used to eliminate jumps that may occur in a visual display when the dead reckoned position/ orientation of an entity is corrected to the most recently communicated position/orientation.

E.5 Dead Reckoning Model (DRM) notation

A DRM notation shall consist of three elements. The first element shall indicate whether the model specifies rotation as either fixed (F) or rotating (R). The second element shall specify dead reckoning rates to be held constant as either rate of position (P) or rate of velocity (V). The third element shall specify the coordinate system to be used with the dead reckoning algorithm as either world coordinates (W) or body axis coordinates (B). The second element identifies the algorithm as being first (P) or second (V) order with respect to position. If the first element specifies rotation (R), the algorithm is first order with respect to orientation.

This notation may be summarized as follows:

DRM (F or R, P or V, W or B)

For example, a DRM that considers orientation fixed and a constant rate of position in world coordinates would be expressed as:

DRM (FPW)

E.6 Dead reckoning formulas

A dead reckoning formula is a formula that represents a forward extrapolation in time. This is equivalent to forward integration of time derivatives. Two examples are given as follows:

Example 1

$$V(t) = \int_{t_0}^{t} A(\tau)d\tau$$

Example 2

$$r(t) = \int_{t_0}^{t} V(\tau)d\tau$$

Implementation of the sender’s dead reckoning algorithm by the receiver helps to limit the receiver’s error.

Table E.1 contains nine standard algorithms for dead reckoning. It provides formulas associated with each field value. The first field value assumes a static entity; four employ world-centered coordinates, whereas four employ body-centered coordinates. Four algorithms are first order (algorithms 2, 3, 6, and 7) and four second order (algorithms 4, 5, 8, and 9) with respect to position. Those algorithms specifying rotation (3, 4, 7, and 8) will contain two formulas, the first being for translational dead reckoning and the second for rotational dead reckoning. All the algorithms that specify rotation are first order with respect to orientation.

The most general equations are those for algorithms 4 DRM (RVW) for the world referenced algorithms and 8 DRM (RVB) for the body referenced algorithms. The other dead reckoning algorithms can be viewed as special cases of these two by setting some combination of velocity, acceleration, and angular velocity equal to zero. A detailed description on the use of algorithms 4 and 8 is contained in E.7. The choice of the algorithm will depend on the circumstances during the life of an entity. For example, it may be more appropriate to use body-centered coordinates to describe trajectory until the final stages when world-centered coordinates may be required to predict terminal interaction with other entities in the simulation. Details on these methods can be found in the documents referenced in Annex J.

Table E.1—Dead reckoning formulas

Field	Model	Formula	Examples
1	STATIC	N/A	Static entities
2	DRM (FPW)	$P = P_0 + V_0\Delta t$	Constant velocity (or low acceleration) linear motion
3	DRM (RPW)	1) $P = P_0 + V_0\Delta t$ 2) $[R]_{w \to b} = [DR][R_0]_{w \to b}$	Similar to DRM 2 but where orientation is required (e.g., visual simulation)
4	DRM (RVW)	1) $P = P_0 + V_0\Delta t + \frac{1}{2}A_0\Delta t^2$ 2) $[R]_{w \to b} = [DR][R_0]_{w \to b}$	Similar to DRM 5 but where orientation is required (e.g., visual simulation)
5	DRM (FVW)	1) $P = P_0 + V_0\Delta t + \frac{1}{2}A_0\Delta t^2$	High speed (e.g., missile) or maneuvering at any speed
6	DRM (FPB)	1) $P = P_0 + [R_0]^{-1}_{w \to b}([R1]V_b)$	Similar to DRM 2 but when body-centered calculation is preferred
7	DRM (RPB)	1) $P = P_0 + [R_0]^{-1}_{w \to b}([R1]V_b)$ 2) $[R]_{w \to b} = [DR][R_0]_{w \to b}$	Similar to DRM 3 but when body-centered calculation is preferred
8	DRM (RVB)	1) $P = P_0 + [R_0]^{-1}_{w \to b}([R1]V_b + [R2]A_b)$ 2) $[R]_{w \to b} = [DR][R_0]_{w \to b}$	Similar to DRM 4 but when body-centered calculation is preferred
9	DRM (FVB)	1) $P = P_0 + [R_0]^{-1}_{w \to b}([R1]V_b + [R2]A_b)$	Similar to DRM 5 but when body-centered calculation is preferred

E.7 Dead reckoning mathematics

E.7.1 General

This subclause describes the position and orientation dead reckoning computations associated with algorithms 4 and 8 and is based on the work done in Towers and Hines [B17].

E.7.2 Equations for algorithm 4 DRM (RVW)

E.7.2.1 General

Two equations for algorithm 4 DRM are detailed in E.7.2.2 and E.7.2.3. The other world referenced algorithms (2, 3 and 5) can be viewed as special cases by setting some combination of acceleration and angular velocity (as appropriate) equal to zero.

E.7.2.2 Position dead reckoning for algorithm 4 DRM (RVW)

The position in world coordinates after time Δt is:

$$P = P_0 + V_0 \Delta t + \frac{1}{2} A_0 \Delta t^2$$

where

P_o is the position vector in world coordinates at initial simulation time

V_o is the velocity vector in world coordinates at initial simulation time

A_o is the acceleration vector in world coordinates at initial simulation time

Δt is the time increment for dead reckoning step

E.7.2.3 Orientation dead reckoning for algorithm 4 DRM (RVW)

The orientation matrix that takes world coordinates into body coordinates at time $t_o + \Delta t$ is:

$$[R]_{w \to b} = [DR][R_0]_{w \to b}$$

where

$[DR]$ is the dead reckoning matrix

$[R_0]_{w \to b}$ is the entity's initial world to body orientation matrix.

These terms are computed in sections E.7.2.4 and E.7.2.5.

E.7.2.4 Computing the dead reckoning matrix [DR]

The matrix that takes body coordinates at time t_o into body coordinates at time $t_o + \Delta t$ is:

$$[DR] = e^{-\Delta t \Omega}$$

or

$$[DR] = \frac{(1 - \cos(|\omega| \Delta t))}{|\omega|^2} \omega\omega^T + \cos(|\omega| \Delta t) I - \frac{\sin(|\omega| \Delta t)}{|\omega|} \Omega$$

where

ω is the angular velocity in body coordinates (ω_x, ω_y, and ω_z being the body axis angular velocities contained in the Entity State PDU). The magnitude of the angular velocity is computed by:

$$|\omega| = (\omega_x^2 + \omega_y^2 + \omega_z^2)^{\frac{1}{2}}$$

NOTE 1—Using this method of dead reckoning, ω will remain constant for every frame (Δt) in the dead reckoning process until a new PDU update.

The 3×3 identity matrix, I, is defined as having 1s on the main diagonal and 0s elsewhere so that:

$$I = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The skew matrix is given by:

$$\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

$\omega\omega^T$ is computed by:

$$\omega\omega^T = \begin{bmatrix} \omega_x\omega_x & \omega_x\omega_y & \omega_x\omega_z \\ \omega_y\omega_x & \omega_y\omega_y & \omega_y\omega_z \\ \omega_z\omega_x & \omega_z\omega_y & \omega_z\omega_z \end{bmatrix}$$

NOTE 2—It was mentioned above that the angular velocities are contained in the Entity State PDU as body axis velocities. However, if the angular velocities are in terms of the Euler angles, then a transformation to body axis angular velocities is needed. Thus, the following transformation formulas are given.

Body axis angular velocities to world (Euler) angular velocities:

$$d\theta / dt = \omega_y \cos\phi - \omega_z \sin\phi$$

$$d\psi / dt = \frac{\omega_y \sin\phi + \omega_z \cos\phi}{\cos\theta}$$

$$d\phi / dt = \omega_x + (d\psi / dt)\sin\theta$$

World (Euler) angular velocities to body axis angular velocities:

$$\omega_x = d\phi / dt - (d\psi / dt)\sin\theta$$

$$\omega_y = (d\theta / dt)\cos\phi + (d\psi / dt)\sin\phi\cos\theta$$

$$\omega_z = -(d\theta / dt)\sin\phi + (d\psi / dt)\cos\phi\cos\theta$$

E.7.2.5 Computing the initial orientation $[R_o]_{w\rightarrow b}$

The matrix that defines the initial orientation at simulation time t_0 can be computed from the initial Euler angles that are available in the Entity State PDU bearing the timestamp reference time corresponding to the simulation time t_0. This matrix is a world to body axis transformation and is defined as:

$$[R_0]_{w\rightarrow b} = \begin{bmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{bmatrix}$$

E.7.2.6 Recovering the Euler angles from $[R]_{w\rightarrow b}$

After $[R]_{w\rightarrow b}$ is determined, the Euler angles at $t_o + \Delta t$ can be computed as follows with $[R]_{w\rightarrow b}$ of the form:

$$\begin{bmatrix} A11 & A12 & A13 \\ A21 & A22 & A23 \\ A31 & A32 & A33 \end{bmatrix}$$

for θ:

$$\theta = \sin^{-1}(-A13)$$

where by definition, θ is the elevation angle of the x-axis above the horizontal plane and will lie in the range:

$$-\frac{\pi}{2} \leq \theta \leq \frac{\pi}{2}$$

for ψ:

$$\cos\theta\cos\psi = A11$$

and because cosθ is always positive for the defined range of θ, then:

$$\cos\theta\sin\psi = A12$$

yields:

$$\operatorname{sgn}(\sin\psi) = \operatorname{sgn}(A12)$$

which then gives:

$$\psi = \cos^{-1}(A11/\cos\theta) \times \operatorname{sgn}(A12)$$

similarly for ϕ:

$$\phi = \cos^{-1}(A33/\cos\theta) \times \operatorname{sgn}(A23)$$

Note that exceptions have to be made for ϕ and ψ when $\theta = +\pi/2$ or $-\pi/2$.

E.7.3 Equations for algorithm 8 DRM (RVB)

E.7.3.1 General

Equations for algorithm 8 DRM are detailed in E.7.3.2 and E.7.3.3. The other body referenced algorithms (6, 7, and 9) can be viewed as special cases by using a subset of algorithm 8.

NOTE—Even though orientation is not dead reckoned for algorithms 6 and 9, angular velocity is still required for position dead reckoning. This is in contrast to the world referenced algorithms 2 and 5, which do not require angular velocity. Acceleration is not required for algorithms 6 and 7, the same as for world referenced algorithms 2 and 3.

E.7.3.2 Position dead reckoning for algorithm 8 DRM (RVB)

As in this axis system the rotation affects the orientation of V and A, then an integral form has to be used. The position in world coordinates after time Δt from body referenced V and A is computed by:

$$P = P_0 + \left[R_0\right]^{-1}_{w \to b}(\left[R1\right]V_b + \left[R2\right]A_b)$$

where

P_o is the position vector in world coordinates at initial simulation time

V_b is the velocity vector in body coordinates at initial simulation time

A_b is the time derivative of V_b

The quantity A_b, which represents the time derivative of V_b, is generally not the same as the entity acceleration expressed in body coordinates. A_b is equivalent to the body acceleration a_b minus the centripetal acceleration experienced in the turn. So the relationship between A_b and a_b is:

$$A_b = a_b - \Omega V_b$$

The remaining terms in this equation are computed as follows:

The matrix $\left[R_0\right]^{-1}_{w \to b}$ is simply the transpose of the initial orientation matrix $\left[R_0\right]_{w \to b}$ at t_o computed in E.7.2.5.

The matrix $[R1]$ is defined as:

$$[R1] = \int_0^{\Delta t} e^{\tau\Omega} d\tau$$

or expanded out becomes:

$$[R1] = \frac{|\omega|\Delta t - \sin(|\omega|\Delta t)}{|\omega|^3}\omega\omega^T + \frac{\sin(|\omega|\Delta t)}{|\omega|}I + \frac{1 - \cos(|\omega|\Delta t)}{|\omega|^2}\Omega$$

The matrix [$R2$] is defined as:

$$[R2] = \int_0^{\Delta t} \tau e^{\tau\Omega} d\tau$$

where [$R2$] expanded out becomes:

$$[R2] = \frac{\frac{1}{2}|\omega|^2\Delta t^2 - \cos(|\omega|\Delta t) - |\omega|\Delta t\sin(|\omega|\Delta t) + 1}{|\omega|^4}\omega\omega^T + \frac{\cos(|\omega|\Delta t) + |\omega|\Delta t\sin(|\omega|\Delta t) - 1}{|\omega|^2}I$$
$$+ \frac{\sin(|\omega|\Delta t) - |\omega|\Delta t\cos(|\omega|\Delta t)}{|\omega|^3}\Omega$$

E.7.3.3 Orientation dead reckoning for algorithm 8 DRM (RVB)

The equation for dead reckoning orientation for algorithm 8 is the same as shown for algorithm 4; thus:

$$[R]_{w \to b} = [DR][R_0]_{w \to b}$$

E.7.4 Geometric interpretation of orientation dead reckoning

E.7.4.1 General

E.7.4.1.1 Introduction

Euler angle representations of rotation have the property that under certain orientations, a large change in at least one Euler angle may result from a small change in object orientation. At the singularity points, one angle can change discontinuously with an infinitesimal change in orientation. This instability in the Euler angles leads to some simulations using other approaches to processing rotations internally, such as using a geometric (quaternion) representation. An introduction to the concepts and mathematics of quaternions and 3-dimensional rotation is given in Kuipers [B10].

For simulations using a geometric (or quaternion) rotation scheme internally, there is a simpler method of calculating the dead reckoning rotation matrix [DR] and an alternative method of determining whether the orientation threshold has been exceeded. The existing dead reckoning algorithms are used; only the method of constructing the [DR] rotation matrix and threshold calculation differs.

The angular velocity ω in the Entity State PDU is actually a vector that describes an axis of rotation. Orientation dead reckoning is performed by successive rotations of the entity about this axis. The angle of rotation (β) about this axis is $|\omega|\Delta t$, that is, the magnitude of the vector multiplied by the time increment for dead reckoning. The equations below show how to form a quaternion from the axis and angle of rotation and how to then convert that quaternion into the DR rotation matrix.

If either the issuing or the receiving entity does not use geometric (quaternion) rotation representations, the standard method of constructing the [DR] rotation matrix (see E.7.2.4) is used. Either method gives identical results.

E.7.4.1.2 Geometric definition of a quaternion

When using a quaternion to specify the orientation of a body, a single, four-component, real, normalized object defines the transformation between world and body coordinates. This quaternion, referred to as q, is traditionally thought of as a scalar plus a vector written as (q_0, $\boldsymbol{q}$) or (q_0, q_x, q_y, q_z).

Three-dimensional rotation can be uniquely described as a rotation of angle β about an axis defined by a unit vector **u**. The rotation quaternion can be constructed from these parameters as follows:

$$q_0 = \cos(\beta/2)$$

$$q_x = u_x \sin(\beta/2)$$

$$q_y = u_y \sin(\beta/2)$$

$$q_z = u_z \sin(\beta/2)$$

Quaternions used to describe rotation are unit quaternions, a subset of general quaternions. Unit quaternions are normalized, having a magnitude of 1:

$$\sqrt{q_0{}^2 + q_x{}^2 + q_y{}^2 + q_z{}^2} = 1$$

E.7.4.1.3 Quaternion multiplication

Successive rotations can be combined by multiplying the quaternions that describe the rotations. Multiplication of two quaternions q_A and q_B, resulting in product q_C, is performed as follows:

$$q_{C0} = q_{A0}q_{B0} - q_{Ax}q_{Bx} - q_{Ay}q_{By} - q_{Az}q_{Bz}$$

$$q_{Cx} = q_{A0}q_{Bx} + q_{Ax}q_{B0} + q_{Ay}q_{Bz} - q_{Az}q_{By}$$

$$q_{Cy} = q_{A0}q_{By} - q_{Ax}q_{Bz} + q_{Ay}q_{B0} + q_{Az}q_{Bx}$$

$$q_{Cz} = q_{A0}q_{Bz} + q_{Ax}q_{By} - q_{Ay}q_{Bx} + q_{Az}q_{B0}$$

Like matrix multiplication, quaternion multiplication is not commutative; that is, in general, $q_Aq_B \neq q_Bq_A$.

E.7.4.1.4 Quaternion dot product

The dot product of two quaternions q_A and q_B is useful for finding the geometric rotation angle (β) between two orientations. It is defined as follows:

$$q_A \bullet q_B = q_{A0}q_{B0} + \boldsymbol{q}_A \bullet \boldsymbol{q}_B = q_{A0}q_{B0} + q_{Ax}q_{Bx} + q_{Ay}q_{By} + q_{Az}q_{Bz} = \cos\frac{\beta}{2}$$

where β is the angle of rotation between the two orientations defined by q_A and q_B.

E.7.4.2 Construction of the dead reckoning quaternion q_{DR}

Using the geometric definition of a quaternion (see E.7.4.1.2), the quaternion q_{DR} that rotates body coordinates at time t_0 into body coordinates at time $t_0 + \Delta t$ can be calculated with:

$$q_{DR0} = \cos(\beta/2)$$

$$q_{DRx} = u_x \sin(\beta/2)$$

$$q_{DRy} = u_y \sin(\beta/2)$$

$$q_{DRz} = u_z \sin(\beta/2)$$

where the angle of rotation β is:

$$\beta = |\omega|\Delta t$$

and the unit vector **u** describing the axis of rotation is:

$$\mathbf{u} = \omega / |\omega|$$

ω and $|\omega|$ are the same as in E.7.2.4, the angular velocity vector from the Entity State PDU and its magnitude.

E.7.4.3 Construction of the dead reckoning rotation matrix [DR] from the DR quaternion

The matrix that rotates body coordinates at time t_0 into body coordinates at time $t_0 + \Delta t$ can be constructed from the dead reckoning quaternion q_{DR} defined in E.7.4.2 as follows:

$$[DR] = \begin{bmatrix} {q_{DR0}}^2 + {q_{DRx}}^2 - {q_{DRy}}^2 - {q_{DRz}}^2 & 2(q_{DRx}q_{DRy} + q_{DR0}q_{DRz}) & 2(q_{DRx}q_{DRz} - q_{DR0}q_{DRy}) \\ 2(q_{DRx}q_{DRy} - q_{DR0}q_{DRz}) & {q_{DR0}}^2 - {q_{DRx}}^2 + {q_{DRy}}^2 - {q_{DRz}}^2 & 2(q_{DRy}q_{DRz} + q_{DR0}q_{DRx}) \\ 2(q_{DRx}q_{DRz} + q_{DR0}q_{DRy}) & 2(q_{DRy}q_{DRz} - q_{DR0}q_{DRx}) & {q_{DR0}}^2 - {q_{DRx}}^2 - {q_{DRy}}^2 + {q_{DRz}}^2 \end{bmatrix}$$

This matrix is identical to the one calculated in E.7.2.4 and can be used for dead reckoning orientation as shown in E.7.2.3.

NOTE—No rotation matrices are required for the pure quaternion method of rotational dead reckoning described in E.8.2.3.

E.7.5 Geometric threshold calculation

E.7.5.1 General

Two methods of calculating orientation threshold are defined, one based on rotation matrices and one based on quaternions. Before these methods were defined, most implementations computed the Euler angles of a rotating body and triggered an Entity State update whenever any of the three Euler angles changed by more than the threshold value. This approach results in more frequent updates than are necessary when near an Euler angle singularity, swelling network traffic and producing extra computational burden.

To ensure that orientation updates take place at a consistent and correct rate, a common method for setting a threshold is defined for both the rotation matrix and quaternion dead reckoning approaches. The method is based on the geometry of rotations, so it is not dependent on the specific representation or implementation technique employed. This resolves the issue of too-frequent state updates triggered by the Euler-angle approach when the orientation of the body *x*-axis is closely aligned with the world coordinate *Z*-axis (e.g., when traveling northbound or southbound at latitudes near the equator).

The following subclauses describe the threshold calculation for matrix-based and quaternion-based dead reckoning independently, but consistently, using the same geometric definition of the true angle of rotation between the actual (current) and dead reckoned orientations. This allows implementers to use either dead reckoning technique defined by the standard and to trigger Entity State updates at the lowest possible rate.

E.7.5.2 Matrix-based orientation dead reckoning threshold computation

When using a rotation matrix to specify the orientation of a body, a single 3 × 3 real, orthonormal array defines the transformation between world and body coordinates. This matrix, referred to as $[R]_{w \to b}$ in the standard dead reckoning algorithms, has two key properties in common with all rotation matrices. First, the transpose of the matrix is the inverse, making inversion of rotation matrices a mathematically fast operation. And second, the trace of the matrix is related to the rotation angle, β, as follows:

$$Tr[R] = 1 + 2\cos\beta$$

where the trace is the sum of the diagonal elements ($Tr[R] = R_{11} + R_{22} + R_{33}$) and β is the geometric angle of rotation, independent of the coordinate system chosen. These properties lead to the definition of the threshold algorithm for rotation matrix dead reckoning.

As a simulated object rotates, the simulation has to maintain two separate rotation matrices, one that describes the actual (true, current) body orientation and one that describes the result of executing the dead reckoning from the last entity state update. Both of these are world-to-body matrices, so to make them easier to distinguish, they are labeled with different subscripts. The actual orientation is labeled $[R]_A$, and the dead reckoned orientation is labeled $[R]_D$. This is a notation change from Table E.1 and E.7.2 necessary to better explain threshold calculations. $[R]_{w \to b}$ is renamed $[R]_D$, representing dead reckoned orientation after extrapolation, and $[R_0]_{w \to b}$ is renamed $[R]_U$, representing the orientation given in the last PDU update. Repeating the orientation dead reckoning formula of E.7.2.3 in this notation gives:

$$[R]_D = [DR][R]_U$$

The rotation error between $[R]_A$ and $[R]_D$ must be determined to see whether they differ by more than the DRA_ORIENT_THRESH value.

It is desirable to compute a tolerance value from DRA_ORIENT_THRESH that will be identically zero if the two rotations match, and that will grow monotonically as they differ by ever-greater angles. To do that, let:

$$\delta = 2 - 2\cos(\mathrm{DRA\_ORIENT\_THRESH})$$

which is selected to satisfy these properties and to make subsequent calculations efficient. The value of δ can be precomputed and stored so that the computationally expensive cosine function does not need to be called again.

The rotation matrix "difference" between $[R]_A$ and $[R]_D$ will be labeled $[R]_E$ and is defined as:

$$[R]_E = [R]_D^T[R]_A$$

where the superscript T refers to the matrix transpose. $[R]_E$ is the matrix that transforms from the dead-reckoned orientation to the actual orientation. The trace of this matrix is a function of the rotation angle, β, between the actual and dead reckoned orientations as defined above, so if:

$$3 - Tr[R]_E > \delta$$

then the rotation angle exceeds the DRA_ORIENT_THRESH. Note that this calculation does not involve trigonometric functions if the δ value is precalculated and stored, and only the diagonal values of $[R]_E$ are relevant, which means that only those three cells need to be computed. The function $3 - Tr[R]_E$ is chosen so that the left-hand side is zero if the orientations do not differ (that is if $[R]_E$ is the identity matrix).

E.7.5.3 Quaternion-based orientation dead reckoning threshold computation

As a simulated object rotates, the simulation has to maintain two separate quaternions, one that describes the actual (true, current) body orientation (q_A) and one that describes the result of executing the dead reckoning (q_D) from the last entity state update (q_U). q_D can be found with a quaternion multiplication (see E.7.4.1.3) of q_U by the dead reckoning quaternion, q_{DR}, found from the entity's angular velocity ω and Δt as shown in E.7.4.2:

$$q_D = q_U q_{DR}$$

The key property used here is that the dot product of two quaternions q_A and q_D is related to the geometric rotation angle between them (see E.7.4.1.4). That is:

$$q_A \bullet q_D = \cos\frac{\beta}{2}$$

where β is the angle of rotation between the two orientations defined by q_A and q_D.

The rotation error between these orientations must be determined to see whether they differ by more than the DRA_ORIENT_THRESH value.

It is desirable to compute a tolerance value from DRA_ORIENT_THRESH that will be identically zero if the two rotations match, and that will grow monotonically as they differ by ever-greater angles. To do that, let:

$$\varepsilon = 1 - \cos(\text{DRA\_ORIENT\_THRESH}/2)$$

which is selected to satisfy these properties and to make subsequent calculations efficient. The value of ε can be precomputed and stored so that the computationally expensive cosine function does not need to be called again.

The dot product of the current and dead-reckoned quaternions is formed:

$$S = q_A \bullet q_D = q_{A0}q_{D0} + q_{Ax}q_{Dx} + q_{Ay}q_{Dy} + q_{Az}q_{Dz}$$

and if:

$$1 - S > \varepsilon$$

then the angle between the two quaternions exceeds the threshold and a state update is required. As for matrix-based dead reckoning, this calculation requires no trigonometric functions if the ε value is precomputed and stored. The function $1 - S$ is chosen so that the left-hand side is zero if the orientations do not differ.

When the orientation threshold (or any Entity State PDU threshold) is broken, a PDU update occurs and the actual orientation becomes the update orientation in the PDU:

$$q_U \leftarrow q_A$$

This is also the case when a heartbeat period expires, causing an update.

E.8 Use of the Other Parameters field in Dead Reckoning Parameters

E.8.1 Use of Other Parameters for additional dead reckoning algorithms

The Entity State PDU contains a 120-bit Other Parameters subfield within the Dead Reckoning Parameters field that can be used to define additional parameters (beyond the standard set of position, orientation, velocity, angular velocity, and acceleration) required for more complex dead reckoning algorithms. For example, a ship slew rate parameter could be defined if an algorithm required slew rate to calculate dead reckoning. Additional algorithms are identified with a Dead Reckoning Algorithm value greater than 9. The Dead Reckoning Algorithm value determines the format of the Other Parameters field, if used.

E.8.2 Use of Other Parameters for standard algorithms 1 through 9

E.8.2.1 General

If the standard dead reckoning algorithms 1 through 9 defined in this annex are used, the first 8 bits of the 120-bit field shall be an enumeration named DR Parameters Type (see [UID 296]) that describes the format of the remaining 112 bits. One of the following formats should be used if no other use of the Other Parameters field is defined for the exercise. If the Other Parameters field is not used, the DR Parameters Type shall be set to None (0) and the remaining 112 bits shall be set to zero.

Much of the expensive computational calculation can be avoided in the receiving simulations by use of extra information in the Other Parameters field of the Dead Reckoning Parameters in the Entity State PDU. For fixed DRM entities (no rotational dead reckoning), computation is saved by the issuing simulation providing orientation in local coordinates directly in the PDU to avoid coordinate conversion by each receiving simulation. For rotating DRM entities, the issuing simulation provides the entity orientation as a quaternion, which is computationally simpler to dead reckon than Euler angles.

If the Other Parameters field is not being used for additional dead reckoning formulas as specified in E.8.1, one of the following formats should be used when issuing Entity State PDUs.

E.8.2.2 Fixed DRM entities

Most simulations require entity orientation in local coordinates (i.e., the familiar Yaw-Pitch-Roll Euler angles) for processing and visualization. This requires conversion of the world coordinate Euler angles upon receipt of each Entity State PDU to local coordinate Euler angles. Although this is a relatively straightforward conversion, the amount of computation required can be significant for large exercises with many entities.

Entities using fixed dead reckoning algorithms, STATIC (1), FPW (2), FVW (5), FPB (6), or FVB (9) should use the form of the Dead Reckoning Parameters shown in Table E.2. The local coordinate Yaw-Pitch-Roll Euler angles are put directly in Other Parameters by the issuing simulation, eliminating orientation conversion for receiving simulations.

The local coordinate system used here is defined by North, East, and Down axes with their origin at the entity's center of bounding volume. The axes move with the entity's position but do not rotate with it. The Down axis is vertical, i.e., perpendicular to the surface of the Earth ellipsoid, and the North and East axes are perpendicular to the Down axis (i.e., they are locally horizontal) in true North and East directions from the entity.

Local coordinate Yaw-Pitch-Roll Euler angles are defined identically to the Psi-Theta-Phi Euler angles in world coordinates (see 1.6.3), except that they are angles about the local North, East, and Down axes, respectively, instead of the world *X*-, *Y*-, and *Z*-axes.

The fields of Other Parameters are as follows:

a) *DR Parameters Type*. This field shall be set to Local Euler Angles (1) and shall be represented by an 8-bit enumeration.

b) *Local Yaw*. Heading from true North, positive to the right, in radians. This field shall be represented by a 32-bit floating point number.

c) *Local Pitch*. Elevation angle above or below the local horizon, positive up, in radians. This field shall be represented by a 32-bit floating point number, and its value shall be limited to between $-\pi/2$ and $+\pi/2$, inclusive.

d) *Local Roll*. Bank angle from the local horizontal, positive tilt to the right (also called right wing down), in radians. This field shall be represented by a 32-bit floating point number.

Table E.2—Local Euler Angles Dead Reckoning Parameters

Field Size (bits)	Field Name	Data Type
8	Dead Reckoning Algorithm	8-bit enumeration
120	Other Parameters	DR Parameters Type—8-bit enumeration = 1
		Padding—16 bits unused
		Local Yaw—32-bit floating point
		Local Pitch—32-bit floating point
		Local Roll—32-bit floating point
96	Entity Linear Acceleration	3 × 32-bit floating point
96	Entity Angular Velocity	3 × 32-bit floating point
Total Local Euler Angles Dead Reckoning Parameters size = 320 bits		

E.8.2.3 Rotating DRM entities

E.8.2.3.1 General

Entities using rotating dead reckoning algorithms RPW (3), RVW (4), RPB (7), or RVB (8) should use the form of the Dead Reckoning Parameters shown in Table E.3. The quaternion can be used by receiving simulations to eliminate some of the expensive trigonometric computation required by the traditional method of using Euler angles and rotation matrices for dead reckoning the orientation of each entity.

The fields of Other Parameters are as follows:

a) *DR Parameters Type*. This field shall be set to World Orientation Quaternion (2) and shall be represented by an 8-bit enumeration.

b) q_U. Four-valued unit quaternion that represents the orientation of the entity. The first value is approximated as a 16-bit unsigned integer. The remaining three values are 32-bit floating point numbers.

Table E.3—World Orientation Quaternion Dead Reckoning Parameters

Field Size (bits)	Field Name	Data Type
8	Dead Reckoning Algorithm	8-bit enumeration
120	Other Parameters	DR Parameters Type—8-bit enumeration = 2
		$\hat{q}_{U0}$ —16-bit unsigned integer
		q_{Ux}—32-bit floating point
		q_{Uy}—32-bit floating point
		q_{Uz}—32-bit floating point
96	Entity Linear Acceleration	3 × 32-bit floating point
96	Entity Angular Velocity	3 × 32-bit floating point
Total World Orientation Quaternion Dead Reckoning Parameters size = 320 bits		

Normally, a quaternion would require four floating point values. To fit a quaternion into 112 bits, the first of the four values, q_{U0}, is converted to a 16-bit scaled integer $\hat{q}_{U0}$ by the issuing simulation. Receiving simulations can convert the integer back to a floating point value and recover the lost resolution as described here.

E.8.2.3.2 Issuance of orientation quaternion

The update orientation quaternion, q_U, is calculated by the issuing simulation using the world coordinate Euler angles ψ, θ, and ϕ (see 1.6.3) as follows:

$$q_{U0} = \cos\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2}$$

$$q_{Ux} = \cos\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2} - \sin\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2}$$

$$q_{Uy} = \cos\frac{\psi}{2}\sin\frac{\theta}{2}\cos\frac{\phi}{2} + \sin\frac{\psi}{2}\cos\frac{\theta}{2}\sin\frac{\phi}{2}$$

$$q_{Uz} = \sin\frac{\psi}{2}\cos\frac{\theta}{2}\cos\frac{\phi}{2} - \cos\frac{\psi}{2}\sin\frac{\theta}{2}\sin\frac{\phi}{2}$$

If the first term q_{U0} is negative, all four values are negated to make the first term positive. This does not change the entity orientation represented by the quaternion.

If $q_{U0} < 0$:

$$q_{U0} \leftarrow -q_{U0}$$

$$q_{Ux} \leftarrow -q_{Ux}$$

$$q_{Uy} \leftarrow -q_{Uy}$$

$$q_{Uz} \leftarrow -q_{Uz}$$

The first term is converted to $\hat{q}_{U0}$ by multiplying it by 65 536 and casting as a 16-bit unsigned integer (i.e., truncate down to the nearest integer, and do not round up):

$$\hat{q}_{U0} = \text{truncate}(q_{U0} \times 65\ 536)$$

If $q_{U0} \geq 1$, set $\hat{q}_{U0}$ to 65 535 (q_{U0} could be greater than 1 due to rounding error).

The four terms $\hat{q}_{U0}$, q_{Ux}, q_{Uy}, and q_{Uz} are now ready to be put in Other Parameters by the issuing simulation.

E.8.2.3.3 Receipt of orientation quaternion

The receiving simulation recovers the first term as a floating point value with:

$$q_{U0} = \sqrt{1 - (q_{Ux}^2 + q_{Uy}^2 + q_{Uz}^2)}$$

A faster calculation of the square root can be performed with $\hat{q}_{U0}$ as follows:

$$q_{U0} \approx \frac{\hat{q}_{U0}}{65\ 536} + \frac{1}{131\ 072}$$

$$q_{U0} \leftarrow \frac{1}{2}\left(q_{U0} + \frac{1 - (q_{Ux}^2 + q_{Uy}^2 + q_{Uz}^2)}{q_{U0}}\right)$$

This approximation can be improved by iterating, but the calculation done once as shown without further iteration is sufficient for a single-precision floating point result because the original guess of q_{U0} is close to the correct value.

The entity can now be dead reckoned efficiently with a quaternion multiplication (see E.7.4.1.3) of q_{U} by the dead reckoning quaternion, q_{DR}, found from the entity's angular velocity ω and Δt as shown in E.7.4.3:

$$q_D = q_U q_{DR}$$

The Psi-Theta-Phi Euler angles for the dead-reckoned orientation are derived from q_{D} as follows:

$$\psi = \tan^{-1}(2(q_{Dx}q_{Dy} + q_{D0}q_{Dz})/(q_{D0}^2 + q_{Dx}^2 - q_{Dy}^2 - q_{Dz}^2))$$

$$\theta = \sin^{-1}(-2(q_{Dx}q_{Dz} - q_{D0}q_{Dy}))$$

$$\varphi = \tan^{-1}(2(q_{Dy}q_{Dz} + q_{D0}q_{Dx})/(q_{D0}^2 - q_{Dx}^2 - q_{Dy}^2 + q_{Dz}^2))$$

The $\tan^{-1}$ operations are properly implemented with the ATAN2 function. As with recovery of Euler angles from a rotation matrix, exceptions have to be made when $\theta = \pm\pi/2$.

E.9 Smoothing

When a new update of position is received from another entity, a correction in position is usually required so that the entity may be depicted in the simulation as accurately as possible. If the new position is put into the image display system immediately, it can cause jumps in the displayed image. Hence, the preferred method is to gradually correct the position of the entity. This is called smoothing. A simple example of smoothing follows. To perform the smoothing over a time period of *n* iterations, a new update is extrapolated ahead by that time period to obtain P_f, the position at the end of the smoothing period. Each iteration linearly approaches that position by the following equation. This is repeated for all three position axes and the three Euler angles for orientation:

$$xP_i = P_o + (P_f - P_o)i/n$$

where

i is an integer, from 1 to n
n is an integer, the number of smoothing points
P_0 is the starting position of smoothing (i.e., the position before update)
P_i is the ith smoothing position
P_f is the position at the final nth smoothing point

Annex F

(informative)

Heartbeats, timeouts, and thresholds

F.1 Scope

This annex defines additional information related to heartbeats (periodic updates) of information using Distributed Interactive Simulation (DIS) Protocol Data Units (PDUs), the timeout of information, and implementing threshold parameters. An informative annex represents guidance related to implementing Clause 4 and Clause 5 fundamental requirements so that they are properly understood and implemented. Heartbeat requirements are contained in 4.2.6. Heartbeats apply to entities, other objects, and supplemental PDUs. Timeouts apply to remote data. Timeout requirements are defined in 4.2.7 and threshold requirements in 4.2.8.

F.2 Heartbeats

F.2.1 General

The original DIS concept had one heartbeat for all entities, regardless of kind, domain, or whether the entity was stationary or moving. Experience in conducting DIS exercises has led to establishing different heartbeat rates for different types of entities. The most common is to separate land domain entities and to give them a separate heartbeat from all other entities. This has been done due to the significant increase in land entities that are present in modern simulation exercises. Having different heartbeat rates can reduce exercise network loading. An analysis was conducted, and it was determined that independent heartbeat rates should be established for each kind and domain for an entity and that there should be one heartbeat rate for all stationary entities regardless of their kind or domain.

It is recognized that there will be situations where all DIS simulations participating in a DIS exercise will not have been upgraded to implement multiple entity and object heartbeats based on kind, domain, and whether an entity is moving or stationary.

F.2.2 Guidelines

The following guidelines and information are provided:

a) Heartbeats apply to local entities and other local objects and supplemental data. Heartbeat timeouts only apply to remote entities, other remote objects, and remote supplemental data. Local data never times out or is removed from a simulation because it did not meet its heartbeat requirement for transmission.
b) When a simulation joins or rejoins an exercise and has a considerable quantity of data to initially send, it should initially consider metering the data so as not to overwhelm the exercise network or individual simulation applications. This is also true for subsequent heartbeats.
c) A simulation assigns a separate heartbeat timer to each object or supplemental PDU for each local object for which it transmits a PDU.
d) Each heartbeat timer is restarted whenever a full set of data is sent in PDU(s) for the following cases:
    1) The first PDU is initially sent for the entity, other object, or supplemental data.

2) A PDU is sent because of a change in data.
3) A PDU is sent because a threshold is broken.
4) A PDU is sent to meet a heartbeat requirement.
5) A PDU is sent to meet other issuance requirements.

F.2.3 Mixed heartbeat environments

A simulation either meets all the heartbeat and timeout requirements or only meets some of them. These are defined as follows:

a) *Minimum Heartbeat Compliance (MHC) Simulation*. A simulation that (1) does not implement a separate heartbeat parameter for each local entity kind and domain, for local stationary entities, and for all other local objects and supplemental PDUs that it is able to transmit or (2) does not implement a separate timeout value for each remote entity kind and domain, for stationary remote entities, and for other remote object and supplemental PDUs that it is able to receive and process.
b) *Full Heartbeat Compliance (FHC) Simulation*. A simulation that (1) implements a separate heartbeat parameter for each local entity kind and domain, for local stationary entities, and for all other local objects and supplemental PDUs that it is able to transmit and (2) implements a separate timeout value for each remote entity kind and domain, for stationary remote entities, and for other remote object and supplemental PDUs that it is able to receive and process.

Table F.1 and Table F.2 show how simulations may achieve heartbeat and timeout interoperability when there is a mixture of simulations some of which meet the minimum heartbeat compliance requirements and some of which are fully compliant with the standard.

Table F.1—Heartbeat and timeout interoperability—option 1

MHC simulations: Set all heartbeat and timeout parameters to a single value that matches the single value used by FHC simulations.

Example:

	Normal	Mixed HC
HRT_BEAT_TIMER	5 s	10 s

NOTE—The HRT_BEAT_TIMER symbolic name is no longer used. It was originally defined as the single heart beat timer symbolic name in IEEE Std 1278.1™-1995.

FHC simulations. Set all heartbeat and timeout values to the same single value. Example: An FHC simulation transmits air and ground moving and stationary entities. It normally gives them different heartbeat values:

	Normal	Mixed HC
HBT_ESPDU_PLATFORM_AIR	5 s	10 s
HBT_ESPDU_PLATFORM_LAND	55 s	10 s
HBT_STATIONARY	60 s	10 s

Table F.2—Heartbeat and timeout interoperability—option 2

MHC simulations: Set all heartbeat parameters to a single value that is equal to the lowest FHC heartbeat rate for a type of entity that is received and processed.

Heartbeat Example:

MHC Simulation A only transmits air entities and normally at a 15 s heartbeat rate.

FHC heartbeats

HBT_ESPDU_PLATFORM_AIR	5 s	lowest
HBT_ESPDU_PLATFORM_LAND	55 s	
HBT_STATIONARY	60 s	highest

MHC Simulation A sets its heartbeat rate to 5 s instead of its normal 15 s rate.

Timeout Example:

MHC Simulation A only receives remote air entities, and because the normal local air entity heartbeat is 15 s, and the multiplier value is 3, the normal remote entity timeout is 45 s (15 × 3).

FHC heartbeats

HBT_ESPDU_PLATFORM_AIR	5 s	lowest
HBT_ESPDU_PLATFORM_LAND	55 s	
HBT_STATIONARY	60 s	highest

MHC Simulation A sets its timeout rate to 180 s instead of its normal 45 s rate. This way, it will not timeout stationary air entities prematurely.

F.3 Timeouts

Timeouts are used to clear out remote data if a PDU is not received for the data within a specified time interval. The original DIS concept had one timeout parameter for all remote entities and other remote objects and a separate timeout parameter for each type of remote supplemental PDU. There was one multiplier value that was used to calculate the timeout parameters for all heartbeated PDUs. Experience in conducting DIS exercises has led to establishing different timeout rates for different types of entities. The most common is to have a separate timeout for land domain entities from all other entities. An analysis was conducted, and it was determined that independent timeout rates should be established for each corresponding heartbeat parameter.

See Table F.1 for techniques for achieving interoperability when all the simulations in an exercise are not fully compliant with the timeout requirements in this standard.

F.4 Thresholds

Thresholds are designed so that differences in position and other information for objects, such as entities, between the local simulation that is sending the objects and at remote simulations that are receiving the objects are reduced to an acceptable level. Thresholds are also used to cause different logic to be invoked if a certain value is reached.

General threshold requirements and a list of thresholds found in the standard are contained in 4.2.8. Thresholds requirements for a specific PDU type are defined in Clause 5 as part of the issuance rules. The default values and other information related to a threshold value are contained in the Symbolic Name table in 6.1.8.

Threshold requirements related to dead reckoning an object are contained in Annex E.

An example of the use of different thresholds among simulation and making dynamic changes to threshold values during an exercise is the case of a virtual air refueling tanker simulator whose boom operator is getting trained to refuel aircraft. As this is primarily a visual task, it is important that both the aircraft approaching the boom be updated at a higher rate than normal, as well as the tanker aircraft entity so (1) the refueling aircraft appears to move realistically to the boom operator and (2) the boom extending from the tanker simulator, and the tanker aircraft itself, appear to have smooth, realistic motion to a pilot that is in a virtual cockpit being refueled. (This is true even though both simulations will normally have smoothing algorithms in their image generators to reduce jerkiness.) However, this high-update rate need not necessarily be sustained for the entire life of the tanker aircraft or aircraft being refueled. It depends on whether any virtual aircraft can see the tanker visually and whether the boom operator can see any other aircraft in its view. If neither is true, then the aircraft entity update rates can be reduced.

Another example of the need for higher thresholds for some types of entities is the case of Tactical Ballistic Missiles being intercepted by other missiles or directed energy weapons.

Annex G

(informative)

Time calculations and uses

G.1 Scope

This annex provides information on the use of time in Distributed Interactive Simulation (DIS) and time calculations associated with synchronization when operating in a real-time, distributed simulation environment. This annex is based on Saunders [B15].

G.2 General

Distributed Interactive Simulation (DIS) is one of two commonly used standards representing the simulation industry's best understanding of how to build realistic, complex, virtual worlds. The High Level Architecture (HLA) is the other standard being used, and typically a joint training exercise will be a hybrid event with both DIS simulations and HLA federates participating through the use of HLA gateways. Although we have accomplished much, and demonstrated much, the technology is still in its infancy in many ways. Some aspects of simulation have become very real such as the visualization of the environment and entities at a participant's own location. Although much of this problem can be traced to the different levels of fidelity represented by many simulations participating in a distributed event, the misunderstanding of time has led to the use of only real-world Coordinated Universal Time (UTC) time when conducting an exercise.

The use of time in distributed simulation continues to be misunderstood and misapplied. Many simulations are not following the standard with respect to time, and this has resulted in reduced flexibility and interoperability for the warfighter. This annex will help people to understand time and how to implement it properly.

G.3 Why time is important

The DIS community operates under a pair of conflicting objectives: (1) make DIS widely usable and (2) make DIS widely effective. Wide use forces us toward simple protocols with requirements that are easy to implement and execute. Wide effectiveness forces us toward simulation techniques that produce the correct answers. The DIS standards represent the community's best effort to balance these objectives, a policy commonly referred to as the "80/20 Solution." This should represent the solution that addresses 80% of the problem spectrum. Unusual special cases may require extra effort, but they are 20% by definition. Training exercises, tests, and experimentation involving fielded units are always conducted in real time. That is, the exercise time ticks away, second by second, just like happens in the real world. One second of exercise time equals one second of real-world time. This is called the Real-Time Protocol. DIS also supports a Non-Real-Time protocol where simulation time can run faster or slower than real time. This will be discussed later in this annex.

Time is needed for a variety of reasons:

a) To keep the synthetic representation of the world operating on the same time so participants can coordinate time-sensitive activities just like they would in the real world.
b) To support operational and mission analysis and mission playback. You cannot do efficient analysis unless everyone's data is time-tagged correctly. If a participant says that its Detonation Protocol

Data Unit (PDU) blew up a vehicle at 23:40:30 and the vehicle owner reports the vehicle was destroyed at 23:40:25, 5 s before the Detonation PDU, analysis becomes almost impossible. All analysis depends on the proper time-ordering of data.

c) To support exercise troubleshooting. Again, examining PDUs is a common technique for isolating a problem, and the order in which they occurred can become an important clue to determine what exactly is the problem.

d) To support dead reckoning of remote entities and other objects received from other simulations. Time is one of the most misunderstood aspects of dead reckoning, especially the erroneous assumption that simulation time cannot be synchronized and used for dead reckoning unless it is Absolute simulation time.

e) To support network loading analysis and to identify contributors to peak loading conditions.

f) To correlate scripted operational events that are supposed to occur simultaneously.

g) To correlate truth data with operational perceived data.

h) To support mission rehearsal for some date and time in the future. It is in this area that the present misunderstanding and implementation of time has led to the inability to conduct operations at a future date and time except by manual manipulation of the environment (e.g., setting the moon phase to a quarter moon for 3-h night instead of allowing the simulation to use simulation time to determine the correct date and time at a given geographic location; this may work for a training event that only lasts a few hours, but it becomes more problematic the longer the exercise runs).

i) To support historical reconstruction of an actual real-world operation using simulation where the date and times are in the past.

j) To automatically support high-fidelity environmental effects such as tides, moonlight, night and day, solar eclipses, and so on instead of manually attempting to set those parameters. These environmental effects are based on dates and times that are typically in a database. Without the ability to set simulation time separate from absolute or relative time, the data cannot be used electronically unless software is built to allow an offset to be put in to compensate for the use of absolute or relative time. If the exercise is based on real-world UTC time and the exercise is delayed by several hours, then all the scripts and offsets have to be adjusted.

k) To support radio propagation effects based on the time of day or season of the year at a given geographical location.

l) To support highly coupled events (e.g., enemy missile versus friend interceptor missile engagement).

DIS exercises use the coordination of time to make simulated events unfold in causal order. A simulator can independently say that it launched a munition at time t and it detonated later at time $t + x$. As the munition and detonation were modeled in the same simulator with the same clock, receivers can determine the order in which these events occurred. However, many cause-and-effect situations cannot be modeled in one simulator. Receivers have to reconstruct the proper time ordering and time between events based on the timestamps in the events' PDUs. For this to work, timestamps are based on the common reference time.

Reference time allows the coordination of time between simulators, for both simulation time and wall clock time. Reference time is a real-world time, meaning it elapses at the same rate as UTC and never pauses. Any instant of simulation time corresponds to reference time by the rules of the exercise agreement. The relationship between simulation and reference time may change dynamically, such as during simulation freeze and restart. Non-real-time simulations have to manage a more complex relationship between reference time and simulation time.

All PDUs contain a timestamp representing reference time, either absolute or relative. Because the relationship of simulation time to reference time is known by all simulators, any simulation can derive the simulation time from its reference clock using the exercise rules of reference time.

For example, when two simulators need to represent each other's entities with high fidelity, they also need highly coordinated time. It is not enough to know that your wingman is moving up; you need to know when the move started. Reference time allows the correlation of time between your own aircraft and your wingman to the level of fidelity necessary to meet the requirements of your simulation exercise.

In addition to this coordination of the time being simulated (simulation time), the computers that execute the exercise must perform some operations simultaneously, such as freeze or restart. Not only must these events occur at the same simulation time, they must occur at the same wall clock time. The rules of reference time allow this.

Absolute reference time provides the highest fidelity. Often reference time is UTC, the international standard for time in the real world, and the use of UTC is widely interoperable. A Global Positioning System (GPS) can provide an excellent source of UTC, as can other clock systems. Occasionally, neither GPS nor a reference clock traceable to UTC is available, for example, due to facility and security considerations. Such exercises can easily establish their own reference time, with a precision clock whose time is distributed through specialized hardware or computer time server distributing time through a local area network with Network Time Protocol. The reference time selection is an exercise agreement, and it is essential that the clock providing the reference time not be stopped or reset during the exercise. The degree of synchronization between the clocks in each simulator and the actual reference time for the exercise depends on the temporal fidelity required to achieve exercise objectives. Like other exercise parameters, it is an engineering decision to be made during the design phase.

G.4 Time terminology

The following terms are associated with time and are briefly described here. More complete descriptions may be found in 3.1 and 3.2. Specific requirements related to Simulation, Absolute and Relative Time, timestamps, and synchronization are specified in 4.6:

a) *Absolute Time*. Absolute time is time based on the present reference time synchronized across the exercise as defined in item b1iii) in 4.6.3. Absolute time may or may not be UTC time depending on exercise agreements.
b) *Clock Time Record*. This is a 64-bit DIS record consisting of a 32-bit Hour field representing the hours since 0000 h 1 January 1970 UTC and a 32-bit Time Past The Hour field representing the time past the present hour. See 6.2.88.2.3 for the scale factor. The Time Past The Hour field is used in the Timestamp and other fields. The Hour field is used in the Simulation Time field of the Start/Stop PDU. See 6.2.14.
c) *Clock Skew*. Clock skew is a phenomenon in synchronous circuits in which the clock signal (sent from the clock circuit) arrives at different components at different times. This is typically due to two causes. The first is a material flaw, which causes a signal to travel faster or slower than expected. The second is distance: if the signal has to travel the entire length of a circuit, it will likely (depending on the circuit's size) arrive at different parts of the circuit at different times.[11]
d) *DIS Time Format*. The time format used by DIS as specified by the Clock Time record.
e) *Elapsed Time*. This is the same as scenario elapsed time.
f) *Exercise Time*. This is the same as simulation time.
g) *GPS Time*. Global Positioning System (GPS) time is the atomic time scale implemented by the atomic clocks in the GPS ground control stations and the GPS satellites themselves. GPS time was zero at 0 h on 6 January 1980, and because it is not perturbed by leap seconds, GPS is now ahead of UTC time. However, the GPS time has a fixed "week" counter, it rolled over to 0 on 21 August 1999 at 23:59:47Z, and it will continue to do this every 20 years. Computers that are interfaced to a GPS receiver, such as a network time server, have software that adjusts the GPS time to the present UTC

[11] *Source:* See Wikipedia (http://en.wikipedia.org/wiki/Clock_skew).

time. Other sources of accurate time are available on the Internet such as the UTC time available from national time standards organizations.

h) *Local Time*. The local time in a given world time zone based on a 24-h cycle.

i) *Network Time Server*. A computer server on a network that uses the Network Time Protocol (NTP) to synchronize computer clock times in a network of computers.

j) *Non-Real-Time Protocol*. This is the protocol when 1 s of real time does not equal 1 s of absolute, relative, or simulation time. Instead it may represent less or more than 1 s of real-world time. This protocol is typically used for campaign-level simulation exercises where a day's worth of battle is run in a few hours.

k) *Real Time Protocol*. This is the protocol that is based on 1 s of real-world time equaling 1 s of absolute, relative, or simulation time. This is the protocol that is typically used in an exercise.

l) *Real-World Time*. Same as Wall Clock Time.

m) *Reference Time*. A real-world time defined by exercise agreement for the coordination of timestamps. Any point in simulation time corresponds to at least one point in reference time. If an exercise is reset to an earlier simulation time, simulation time in the interval between the freeze instant and the earlier restart instant will correspond to two different reference times. If an exercise is frozen and resumed, reference time in the interval between the freeze and the resume will correspond to the freeze instant of simulation time. Reference time, like any real-world time, does not stop when simulation time is paused.

n) *Relative Time*. Relative time is a reference time that need not be synchronized. Simulation applications using relative time can still be synchronized to a time server. The initial reference time for a simulation application using relative time can be set to zero, to the local time value, or to some other time value.

o) *Scenario Elapsed Time*. This is the time in reference to a scenario that is reading in simulated data. It is usually expressed as the time from the start of reading the scenario file, which usually begins at time zero and increments based on real-world time.

p) *Scenario Time*. There is no universally accepted definition of scenario time. It may mean either the equivalent of exercise or simulation time or the same as scenario elapsed time.

q) *Simulation Host Clock*. This is the internal computer clock, which is normally set to the exercise agreed reference time.

r) *Simulation Time*. Simulation time is the common time being simulated used by participants related to activities and events as they occur in a simulated exercise. It is also referred to as exercise time. See 3.1.

   Simulation time is independent of reference time when initially set and then increments in step with the current reference time. As a simulation is running, each instant of simulation time corresponds to an instant of reference time. If the simulation uses fixed frame steps, then simulation time advances discretely from one frame to the next. To maintain real-time execution, when the simulation completes a frame, it may need to wait until reference time advances to correspond with the simulation time of the next frame. Simulations without fixed frames may advance simulation time by different amounts, waiting as necessary to align the next instant of simulation time with the corresponding instant of reference time. Whenever simulation time is advanced, the interval between the new instant of simulation time and the prior instant of simulation time must equal the interval of between the reference time corresponding to the prior instant of simulation and the reference time corresponding to the new instant of simulation time. This assures 1 s of simulation time passes in 1 s of reference time for real-time operation.

   For example, a mission rehearsal may be conducted for a military operation to take place six months from now. The military operation war plan calls for an assault time of 1600Z on 21 December 2010. The actual mission rehearsal will be carried out with the exercise starting at the assault time on 15 July 2008 at 0800Z. If the Start/Resume PDU is used, the Real-World Time field would be set to the equivalent of 15 July 2008 at 0800Z and the Simulation Time field would be set to the equivalent of

1600Z on 21 December 2010. The timestamp in the PDU Header would be set to absolute time (i.e., the present UTC time at which the Start/Resume PDU is sent).

s) *Time Source*. A service that provides a source for synchronizing time. This is usually in the form of a service that can be reached via electronic means such as a network time server.

t) *Timestamp*. The timestamp is a field in the PDU Header that contains time past the current hour. If absolute time is specified in the timestamp, it may or may not represent UTC time depending on exercise agreements. If relative time is specified in the timestamp, it represents some preselected time past the hour other than the present real-world UTC time. The Timestamp field is 32 bits with 1 bit used to indicate either absolute or relative Timestamp format. See 6.2.88.

u) *UTC Time*. Coordinated Universal Time (abbreviated as UTC as a compromise between the English CUT and the French TUC for temps universel coordonné) is the standard time common to every place in the world. Formerly and still widely called Greenwich Mean Time (GMT), Zulu Time, and also World Time, UTC nominally reflects the mean solar time along the Earth's prime meridian.

v) *Wall Clock Time*. Wall-clock time, also referred to as wall time or real-world time, refers to elapsed time as determined by a chronometer such as a wristwatch, clock on the wall, or computer time display.

G.5 Time management guidance

Time-sensitive synthetic events such as the presence and degree of moonlight, which can affect night visual goggles, can be manually entered into a table so an hour simulated mission or exercise may not need precise date and time information or the simulation may not have been built to maintain simulation time that includes date and time information.

All clocks drift because economic factors limit their accuracy, resulting in the need to re-synchronize clocks. Synchronization takes place in two steps: (1) measurement of the error between the clock and the time source and (2) adjustment of the clock to minimize the error. When the adjustment is made, simulation values extrapolated by dead reckoning will be affected. Two techniques should be employed to minimize these anomalies: (1) frequent synchronization, to minimize the error that has to be adjusted and (2) incremental adjustment, where the change is made over a period of time rather than in one big jump.

But if the exercise runs a week and the simulators need to run 24 h a day, the simulation manual entry of time-sensitive data like the amount of moonlight may not be able to be reset without stopping the simulation.

The fact that some fielded units can only use real-world UTC time synced to GPS time has been touted as the reason why all exercises have to use real-world UTC time. Research has shown that many fielded units have the ability to set their system time to a UTC time manually. More forethought in designing combat systems would have led to a capability to sync a combat system to a UTC time source that represented a past or present UTC time where the entire combat system was only being used to support an exercise and no nonexercise live operations were being supported. Of course, if the combat system is running live operations simultaneously with exercise activity, then this becomes a problem.

G.6 Managing time errors

When time is discussed, it is usually in terms of what will be observed as sluggishness by the virtual simulator's human subjects (hereafter, subjects). Good studies of human perception of time delay usually come around to numbers in the 100 ms to 200 ms neighborhood as upper limits of tolerable delay. If time was only used for turning lights on and off, this would be a reasonable but rough approximation. Unfortunately, DIS simulators also use time for a computation called "dead reckoning" that is used to forward extrapolate the position of entities, such as vehicles, that have motion. Human perception of position can be a significantly more important effect than the temporal disruption of network delays.

The DIS standards have two explicit techniques to control the error in position. The dead reckoning procedures utilize a transmission limit, and new parameters are transmitted whenever the dead reckoning extrapolation exceed a given threshold limit. These parameters, in combination with the precision of the parameters transmitted, allow accurate control of position error bounds.

The other control technique is specifically oriented to managing time error. Each PDU contains a timestamp field indicating the time at which the data contained in the PDU is valid. This is normally the time of transmission but may be an earlier time depending on when the timestamp value is applied. Timestamps are included in each PDU and can be used to support the time part of the dead reckoning equations. Two methods of timestamp correction are provided: (1) absolute timestamps, where time is referenced to a standard; and (2) relative timestamps, where time is not referenced. Both timestamp methods are effective in controlling positional error, if used correctly. The widely held notion that relative timestamps are worthless and that introduction of kooky errors is therefore acceptable is totally false, as the following clause will show.

G.7 Real-world time errors

Many distributed exercises operate on a nationwide computer network, shared with for e-mail and other Internet activities. This network may be implemented with leased circuits that cross the country, it may or may not be encrypted, and the technology used may not be developed to perform DIS simulations. It represents the kinds of Wide Area Network (WAN) technology that is commercially available at economical rates. A number of network latency experiments were conducted on such a network, during normal daily operations, to assess its utility for DIS experiments. Figure G.1 shows a typical latency histogram for a transcontinental path. This information is easy to determine, and network operators collect this information on a routine basis for management reasons related to purchase of additional bandwidth, routing changes, and so on. For any WAN exercise where significant funds are being spent or significant decisions are being made, this information should be assumed to be available.

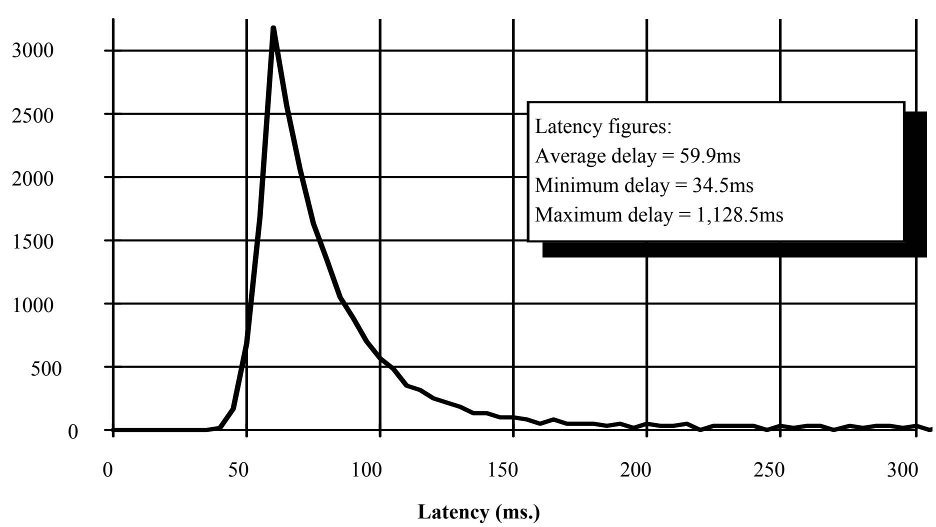


Figure G.1—Typical latency data for a wide area network path from California to Texas

The latency shows a basically exponential distribution, albeit with rather large variance because there are several hops in the path and routers along the path have memory queues that present most of the delay. Remember that no routing priority or bandwidth reservation was used to optimize this traffic; it represents "normal" service levels.

Two interesting effects can be seen from the chart in Figure G.1. First, the average latency is close to the median, and most samples are relatively near the average. In this case, the 66% case represents samples from 35 ms to 85 ms, centered on the average.

Let us see how this might be converted to distance. Table G.1 shows the positional error caused by variance in latency.

Table G.1—Position errors available via time management

Delay (ms)	Common technique(s)	Tank (100 km/h)	Aircraft (1000 km/h)	Missile (4000 km/h)
1000	Voice mark	27.78 m	277.78 m	1111.11 m
85		2.36 m	23.61 m	94.44 m
60	PDU Time per 8.6.2.4	1.67 m	16.67 m	66.67 m
35	Network source	0.97 m	9.72 m	38.89 m
25	NTP on separate Local Area Network	0.69 m	6.94 m	27.78 m
5	Jitter compensation	0.14 m	1.39 m	5.56 m
1		0.03 m	0.28 m	1.11 m
100µs	GPS, IRIG-B	0.003 m	0.03 m	0.11 m

The selection of error thresholds has to take these time errors into account. For example, error thresholds of 5 m across this sample WAN will produce positional errors that, on average, exceed 6.6 m for a tank speed vehicle if latency is ignored. If 5 m accuracy is needed, then the thresholds need to be set to less than 3.3 m. If accuracy of less than 1 m is needed for tank speed objects, setting the threshold to 0 will not provide adequate results and time management techniques have to be used. Similarly if the high end of the 66% solution for missile position errors is less than 100 m, time management had better be considered. It is hard to imagine a missile simulation situation where 100 m errors were acceptable.

G.8 How to process timestamps to reduce errors

The simplest thing to do with timestamps is to ignore them. When a packet is received, just overwrite the sender time stored in the packet with the receiver time of receipt. Across this sample WAN, this produces errors between 35 ms and 1000 ms with an average error of 60 ms. If you can live with the errors from this line of Table G.1, you are done. This is equivalent to the assumption that the clock skew between simulations is zero.

If you want better solutions, you can use the average latency value for your network to offset the packets as you arrive. In this example, you overwrite the sender time with your clock—60 ms. This makes the average error 0 and 66% of the time errors are less than 25 ms. Consult Table G.1, you have less than 50% of the errors of the previous approach.

If the idea of 7 m errors in aircraft position troubles you, then it will be time to process timestamps in the way they were intended. All you need is a clock in your simulator that is regulated relatively close to the reference time used by the clock in the sending simulator. Most modern computers have time-of-day clocks that are accurate to better than 1 part in 10 000 (10 s/day). Such a clock can provide 100 µs accuracy. However, you need to get synchronized in order to use the errors on that line of Table G.1.

The two types of timestamps presume different synchronization methods. The use of absolute timestamps typically uses one of two techniques: (1) GPS synchronization (Katz [B7], [B8]) or (2) NTP synchronization. These papers cover the topic quite thoroughly. Despite these detailed papers, most simulators cannot use absolute time because their owners have not bought them a good $2000 clock.

Relative timestamps provide a poor mans version of absolute timestamps. To coordinate relative time, you get everyone in the exercise to implement some scheme for setting their time-of-day clocks. Many schemes have been used, from shouting "Mark" into a walkie talkie to sending a special PDU across the network before other traffic is started. The smaller the error from Table G.1 you can tolerate, the more accurate the coordination technique you need. However, you need an exercise-specific action to get the clocks aligned, and you need it implemented in every simulator in the exercise. You may require everyone that is using relative timestamps to be synchronized to a time source, including a UTC time source, as far as setting their simulation host clocks. But you cannot assume that everyone has done this as there is no bit in a PDU indicating whether a relative timestamp has been synchronized to a time source.

To make sure that relative timestamps are synchronized, you need to compare information about a received PDU that contains a relative timestamp with the time you are maintaining in your simulation application. This is done using software and without any special hardware. It does require that a few packets be observed before time is well synchronized. As packets are received, the difference between their relative timestamps and the receiver's clock is averaged. This average will correspond to the average latency, and the difference represents clock skew. After a few dozen packets, the difference between the observed average and the real average latency is around 5 ms. After several hundred, the difference is in the 1 ms neighborhood. This is not as long a time to wait as might be imagined. A dozen ES PDUs per minute are emitted by simulators for entities that are stopped. Thus, a few minutes of idle time before the exercise starts provides data for 5 ms accuracy, and at 1 to 2 PDUs per second while moving, 1 ms accuracy can be had in a matter of minutes. Referring to Table G.1, the 1 ms neighborhood is a nice neighborhood. Exercises with stricter needs really should use GPS-based time.

The algorithm for processing relative timestamps has the following steps:

a) Before the simulation begins, build a table giving the average latency of the network link to each host computer on the network. Host computers are known by the Site Number and Application Number they use in identifiers (see 4.2.5.2). Some exercises may publish this data, perhaps assuming that the network latency is only a function of the Site Number. In some networks, this data has to be collected dynamically using PING messages or other means. Although network latency may increase during the exercise, as traffic increases, these stored values should be selected to represent the latency during the next step. These values are called L(SN,AN) to reflect the latency in the link to a host known by its Site Number and Application Number.
b) When the simulation begins, simulators begin to transmit PDUs. The best results are achieved when network latency remains roughly constant during this initial period.
c) When each PDU is received, the Site Number, Application Number, and relative timestamp are examined. It may be desirable to examine only Entity State PDUs, if other PDUs are being sent with relative timestamps in the future.

d) The difference between [the relative timestamp plus L(SN,AN)] and the current local time is calculated. This value, called C(SN,AN), reflects the difference between the clocks. The error in this estimate is the difference between the average network latency and the actual network latency experienced by the PDU.

e) The difference is accumulated on a per (Site Number and Application Number) basis. This may be a simple running average, or a more stable technique, such as Kalman filtering, can be used.

f) Until the exercise is ready to start, as signified by a Start PDU or just a simple timeout, return to step b).

g) Future PDUs are processed by taking the received relative timestamp minus the calculated C(SN,AN) to produce a timestamp usable with the local time.

Annex H

(normative)

Transfer Ownership function

H.1 Scope

This annex provides the detailed requirements related to Protocol Data Units (PDUs) that directly support the Transfer Ownership function. This information is normative in nature and levies a requirement to adhere to the practices described herein.

H.2 Related PDUs

H.2.1 General

The following PDUs are used as part of the Transfer Ownership function. Only the Transfer Ownership (TO) PDU is unique to transfer ownership. All other PDUs used to support transfer ownership are also used to support other Distributed Interactive Simulation (DIS) functions. These will be referred to here as nonunique PDUs. The general and detailed requirements for a nonunique PDU specified in Clause 4 and Clause 5 of this standard apply in addition to the requirements specified in this annex when used in support of transfer ownership. If a conflict exists between the PDU requirements contained outside of 5.9.4 and those contained in this annex, the requirements specified herein shall be considered superseding. If a conflict exists between the requirements in 5.9.4 and those contained in this annex, the requirements in 5.9.4 shall be considered superseding.

H.2.2 Transfer Ownership (TO) PDU

The TO PDU shall be used to request the transfer of an entity and to cancel such a transfer. This PDU is only used in conjunction with the Transfer Ownership function:

a) *Information Content.* The fields shall be set as specified in Table H.1. See also 7.8.4 for the format and definition of the fields for the TO PDU.

Table H.1—TO PDU

Field	Subfield	Value
PDU Header	Standard	As appropriate
Originating Identification (ID)	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0

Table H.1—TO PDU *(continued)*

Field	Subfield	Value
Receiving ID	Site Number Application Number Entity Number	The Receiving ID shall be set based on the Transfer Type field value as follows: If Transfer Type = 1, 4 *Push Transfer* Receiving Site Number Receiving Application Number NO_ENTITY (0) If Transfer Type = 2, 5, 8, 9 *Pull Transfer* Receiving Site Number Receiving Application Number NO_ENTITY (0) If Transfer Type = 7 *Cancel Transfer* The other simulation involved in this transfer transaction shall be specified: Receiving Site Number Receiving Application Number NO_ENTITY (0) If Transfer Type = 10 *Remove Entity* A specific simulation shall be specified: Receiving Site Number Receiving Application Number NO_ENTITY (0)
Request ID		A unique number assigned by a simulation that identifies this transfer transaction. Any subsequent PDU containing a Request ID field from a simulation that is involved in this transfer transaction shall use this value.
Required Reliability Service		1 = Unacknowledged
Transfer Type		As appropriate
Transfer Entity ID		The Entity ID of the entity being transferred. Site Number= nonzero value Application Number = nonzero value Entity Number = nonzero value
Number of Record Sets		0 = No records are associated with this TO PDU. In this case, no Record Set fields are included. n = Number of record sets in this specific PDU. If the total number of record sets is more than can be contained in this PDU, then the PDU shall include a Total Record Sets record, and one or more Set Record-R PDUs are used to send the remaining records.
Record Set #1	See 7.8.4	As appropriate.
Record Set #R	See 7.8.4	As appropriate.

b) *Issuance of the TO PDU*

The TO PDU shall be issued as follows:

1) When a simulation desires to initiate the transfer of an entity or environmental process. In this case, the appropriate timer shall also be set.

2) When a simulation desires to terminate a transfer transaction.

3) When a simulation desires to send a *Remove Entity* TO PDU to resolve an ownership conflict. In this case, the *Remove Entity* TO PDU may be originated either automatically or by operator action.

4) If this is a Push or Pull Transfer request, the appropriate timer shall be set, if applicable, upon sending this PDU.

c) *Receipt of the TO PDU*

A received TO PDU shall be processed by the addressed simulation as follows:

1) If it is a Pull Transfer request, the receiving simulation shall determine whether the entity in the Transfer Entity ID field is one that it owns. If it is the owner, then rule 2) and rule 3) below shall apply. If not, the Acknowledge-R PDU shall be output with the *Unable to Comply* response.

2) If this is a transfer request, and it has passed initial validity checks, an Acknowledge-R PDU shall be output to the simulation indicated in the Originating ID field of the TO PDU with one of the following responses:

   i) An *Unable to Comply* response shall be issued if, for whatever reason, the simulation determines that the request cannot be automatically complied with.

   ii) An *Able to Comply* response shall be issued if the simulation determines that the request can be automatically complied with.

   iii) A *Pending Operator Action* response shall be issued if the simulation is in the Manual Transfer Mode.

3) If this is a *Cancel Transfer Request* that has passed validity checks, the receiving simulation shall cancel the transaction. No Acknowledge-R PDU shall be output in this case.

4) If this is a Remove Entity TO PDU, the simulation contained in the Receiving ID field shall determine whether it has a local active entity that matches the Entity ID in the Transferred Entity field. If it does, it shall proceed as follows:

   i) The receiving simulation shall stop simulating and issuing updates for the local entity. A *final* Entity State PDU shall not be issued.

   ii) Send an *ownership* Event Report with Ownership Status = Local Entity Canceled – Remove Entity TO Received (6).

H.2.3 Acknowledge-R PDU

The Acknowledge-R PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.12.4.6 and 7.11.6. It is used in support of the transfer ownership process to acknowledge the TO PDU when it contains a request to transfer an entity and when an operator has manually determined whether it can accept a transfer (i.e., Able to Comply and Unable to Comply). Only the requirements related to its use in support of transfer ownership are specified here:

a) *Information Content*

The Acknowledge-R PDU when used in support of transfer ownership shall be set as specified in Table H.2.

Table H.2—Acknowledge-R PDU

Field	Subfield	Value
PDU Header	Standard	As appropriate
Originating ID	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0
Receiving ID	Site Number Application Number Entity Number	Site Number of originator of the transfer request. Application Number of originator of the transfer request. 0
Acknowledge Flag	None	5 (Transfer Ownership)
Response Flag	None	One of the following, as applicable: 1 = Able to comply. 2 = Unable to comply. 3 = Pending Operator Action.
Request ID	None	Set to the Request ID value in the TO PDU to which this Acknowledge-R PDU is responding.

b) *Issuance of Acknowledge-R PDU*

The Acknowledge-R PDU shall be issued in support of transfer ownership when any of the following conditions occurs:

1) A TO PDU is received with a transfer type of 1, 2, 4, 5, 8, or 9. In this case, the acquiring simulation shall send an Acknowledge-R PDU with the Response Flag set to Able to Comply (1), Unable to Comply (2), or Pending Operator Action (3). If the TO PDU is accompanied by one or more Set Record-R PDUs, the Acknowledge-R PDU shall not be sent until all expected Set Record-R PDUs have been received. (See 5.9.4.3 for the when not all expected Set Record-R PDUs are received.)
2) An Acknowledge-R PDU was previously sent with the Response Flag set to Pending Operator Action (3), and the acquiring simulation has now determined whether or not it can comply with the request. In this case, the acquiring simulation shall send an Acknowledge-R PDU with the Response Flag set to Able to Comply (1) or Unable to Comply (2).
3) If the Response Flag is set to Unable to Comply (2), the transfer transaction shall be automatically terminated.

c) *Receipt of Acknowledge-R PDU*

When an Acknowledge-R PDU is received in support of transfer ownership, it shall be processed as follows by a simulation that has implemented transfer ownership:

1) If the Response Flag = Able to Comply (1), the receiving simulation shall proceed to the next phase of the transfer transaction as specified in 5.9.4.2.
2) If the Response Flag = Unable to Comply (2), the receiving simulation shall automatically terminate this transfer transaction.

3) If the Response Flag = Pending Operator Action (3), the receiving simulation shall initiate the TO_MAN_RESPONSE_TIMER and await another Acknowledge-R PDU with the Response Flag set to Able to Comply (1) or Unable to Comply (2).

H.2.4 Set Record-R PDU

The Set Record-R PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.12.4.15 and 7.11.15. The Set Record-R PDU is used in support of transfer ownership to pass internal state data. Only the requirements related to its use in support of transfer ownership are specified here:

a) *Information Content*

The Set Record-R PDU fields shall be set as shown in Table H.3.

Table H.3—Set Record-R PDU

Fields	Subfield	Value
PDU Header	Standard	As appropriate
Originating ID	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0
Receiving ID	Site Number Application Number Entity Number	Acquiring Site Number Acquiring Application Number NO_ENTITY
Request ID	None	Set to the Request ID from the original TO PDU.
Required Reliability Service	None	1 (Unacknowledged)
Padding	None	0
Number of Record Sets	None	R = Number of Record Sets in this PDU.
Record Set #1	See 7.8.4	As appropriate.
Record Set #R	See 7.8.4	As appropriate.

b) *Issuance of Set Record-R PDU*

The Set Record-R PDU shall be issued in support of transfer ownership to transmit internal state data as follows:

1) By the divesting simulation that initiated a Push Transfer request when all the necessary records cannot be contained in the TO PDU. Multiple Set Record-R PDUs may be required to send all the necessary data.
2) By the divesting simulation that has sent a *Can Comply* Acknowledge-R PDU in response to a Pull Transfer request. If no internal state data is required to be sent, a single Set Record-R PDU shall be output with the Number of Record Sets field set to zero.

c) *Receipt of Set Record-R PDU*

A received Set Record-R PDU in support of transfer ownership shall be processed by the simulation to whom it is addressed as follows:

1) For a Push Transfer request:

   i) If all Set Record-R PDU(s) have been received, mandatory record data shall be processed as required by this standard. Other record data shall be processed as agreed to in an exercise agreement.

   ii) Optionally, Record-R PDU(s) may be output in accordance with 5.9.4.3.

   iii) If processing the data contained in a Set Record-R PDU results in terminating the transfer, the acquiring simulation shall reply with an *Unable To Comply* Acknowledge R-PDU to the originating simulation.

2) For a Pull Transfer request, the acquiring simulation shall perform either of the following:

   i) Complete the transfer in accordance with 5.9.4.2.3, and optionally, send one or more Record-R PDUs as specified in 5.9.4.3.

   ii) If processing the data contained in a Set Record-R PDU results in terminating the transfer, the acquiring simulation shall reply with a *Cancel Transfer* TO PDU and automatically terminate the transaction.

H.2.5 Record-R PDU

The Record-R PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.12.4.16 and 7.11.16. The Record-R PDU is used by the Transfer Ownership function to provide feedback on internal state data received from the divesting simulation during a transfer transaction. It may also be used to transmit internal state data for an entity in response to a Record Query-R PDU received when no transfer transaction is in progress. Only the first use involving transfer ownership is covered in this paragraph (use of the Record-R PDU to respond to a Record Query-R PDU is specified in 5.12.4.16):

a) *Information Content*

   When the Record-R PDU is used to return a copy of the internal state data received from the divesting simulation during a transfer transaction, the fields shall be set as shown in Table H.4.

Table H.4—Record-R PDU

Field	Subfield	Value
PDU Header	Standard	As appropriate.
Originating ID	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0
Receiving ID	Site Number Application Number Entity Number	Site Number of the divesting simulation Application Number of the divesting simulation NO_ENTITY
Request ID	None	The Request ID that was contained in the original TO PDU, or the original Record Query-R PDU, whichever applies.
Required Reliability Service	None	1 (Unacknowledged).
Padding	None	0
Event Type	None	0 (Other)

Table H.4—Record-R PDU *(continued)*

Field	Subfield	Value
Response Serial Number	None	A sequential number beginning with zero (0) that indicates the Record-R PDU(s) that make up this response. If more than one Record-R PDU is part of the response, each one shall have the same number.
Number of Record Sets (R)	None	Number of Record Sets in this Record-R PDU.
Record Set #1	See 7.8.4	As appropriate.
Record Set #R	See 7.8.4	As appropriate.

b) *Issuance of the Record-R PDU*

The Record-R PDU when used as part of a transfer transaction shall be optional. It may be issued in response to a TO PDU containing records or in response to one or more Set Record-R PDUs received by a simulation as part of a transfer transaction. The issuance requirements of 5.12.4.16.3 shall apply.

c) *Receipt of the Record-R PDU*

Receipt of the Record-R PDU when used as part of a transfer transaction shall be optional. If implemented, the receipt requirements of 5.12.4.16.4 shall apply.

H.2.6 Event Report PDU

The Event Report PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.6.5.12 and 7.5.12. When used in support of transfer ownership, the Event Report PDU shall be used to report ownership of an entity by a simulation that has implemented transfer ownership. When used for this purpose, it shall be referred to as the *ownership* Event Report PDU. Only the requirements related to its use in support of transfer ownership are specified here. Note that the *ownership* Data PDU is used to respond to a an *ownership* Data Query PDU:

a) *Information Content*

When the Event Report PDU is used to support the Transfer Ownership function, the fields shall be set as shown in Table H.5.

Table H.5—*ownership* Event Report PDU

Field	Subfield	Value
PDU Header	Standard	As appropriate.
Originating ID	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0
Receiving ID	Site Number Application Number Entity Number	ALL_SITES ALL_APPLIC 0
Event Type		17 = Ownership Report
Padding		0

Table H.5—*ownership* Event Report PDU *(continued)*

Field	Subfield	Value
Number of Fixed Datum Records		0
Number of Variable Datum Records		1
Variable Datum record #1	Variable Datum ID	15 800 = Ownership Status
	Variable Datum Length	64
	Variable Datum Value = Ownership Status record (see 6.2.65)	Entity ID of the transferred entity
		Ownership Status value [see item b) in H.2.6]
		0 (8 bits unused padding)

b) *Issuance of the ownership Event Report PDU*

The *ownership* Event Report PDU shall be transmitted with the Ownership Status value set as follows:

1) New Owner (1). An acquiring simulation assumes ownership and has transmitted the *initial* Entity State PDU. It shall transmit the *ownership* Event Report PDU immediately following the *initial* Entity State PDU. The *ownership* Event Report PDU shall be output regardless of the PDU Status record Transferred Entity Indicator field value.

2) Ownership Query Response (2). A transfer-capable simulation receives an *ownership* Data Query PDU, and it believes it is the owner of the referenced entity.

3) Ownership Conflict (3). A transfer-capable simulation determines that an ownership conflict exists and it believes that it is the rightful owner.

4) Local Entity Canceled—Auto Resolve Conflict (4). A simulation automatically cancels a local entity that it has determined is involved in an ownership conflict and it is not the rightful owner.

5) Local Entity Canceled—Manual Resolve Conflict (5). A simulation manually cancels a local entity by operator action after it has determined that it was involved in an ownership conflict and it was not the rightful owner.

6) Local Entity Canceled—Remove Entity TO Received (6). A simulation automatically cancels a local entity upon receiving a Remove Entity TO PDU addressed to it.

c) *Receipt of the ownership Event Report PDU*

When an *ownership* Event Report PDU is received, it shall be processed as follows:

1) If the divesting simulation receives an *ownership* Event Report before the transfer is complete [see item b6) in 5.9.4.2.2], the divesting simulation may ignore the report.

2) If no transfer transaction is active for the entity and the Ownership Status = New Owner (1) or Ownership Conflict (3), and the entity is a local active entity, and the simulation receiving the PDU has an indication that it terminated a transfer transaction with the simulation shown in the Originating ID field, it shall process the PDU as specified in item g2) in 5.9.4.2.1.

3) If the Ownership Status = Ownership Conflict (3) and the simulation receiving the PDU believes it is the rightful owner, it shall send an *ownership* Event Report PDU with Ownership Status = Ownership Conflict (3).

4) All other reason codes not specified in item c) above shall be ignored and the PDU may be discarded without further processing.

H.2.7 Entity State PDU

The Entity State PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.3.2 and 7.2.2. When used in support of transfer ownership, the Entity State PDU is used by a new owner to report the entity upon assuming ownership and thereafter in accordance with the rules for reporting an entity except for requirements related to entity state data fields whose values have to be maintained that are contained herein. Only the requirements related to its use in support of transfer ownership are specified here:

a) *Information Content*

The *initial* Entity State PDU sent by the acquiring simulation shall have the fields and subfields set in accordance with Table H.6. Initial values that are subsequently changeable are also specified in the table.

Table H.6—*initial* Entity State PDU

Field	Subfield	Value (see NOTE)
PDU Header	All	Per standard
Entity ID	All	Not changeable
Force ID	Force ID	Per standard
Number of Variable Parameter Records (*N*)	Number	Per standard
Entity Type	All	Not changeable
Alternate Entity Type	All	Not changeable
Entity Linear Velocity	All	Per standard
Entity Location	All	Per standard
Entity Orientation	All	Per standard
Entity Appearance	All	Per standard
Dead Reckoning Parameters	All	Per standard
Entity Markings	All	Not changeable
Capabilities	All	Per standard
Variable Parameter records	All	Per standard

NOTE—The Values shown in Table H.6 are defined as follows:

Per Standard. As specified by IEEE Std 1278.1.

Not changeable. The values in this field and all its subfields, if any, shall remain the same as was contained in the last Entity State PDU received from the divesting owner prior to creation of this entity by the acquiring simulation in its local database.

b) *Issuance of the Entity State PDU*

When the Entity State PDU is generated by a simulation that has accepted ownership of an entity or environmental process due to a transfer transaction, the following requirements shall be met. Any reference to Entity State PDU shall also apply to the Environmental Process PDU unless otherwise specified. If there is a conflict between the requirements here and those specified elsewhere in 5.9.4, this paragraph shall be considered superseding:

1) The fields in the *initial* Entity State PDU and subsequent Entity State or Entity State Update PDUs shall be set as specified in Table H.6.
2) The *initial* Entity State PDU shall be output immediately after the entity has been converted or instantiated as a local entity.
3) Further remote data received on the entity from the divesting simulation after the entity has been converted or instantiated as a local entity shall be ignored.
4) Issuance of the *initial* Entity State PDU ends the transfer transaction at the Acquiring Simulation when the PDU is sent and at the Divesting Simulation when the PDU is received. However, the Acquiring Simulation still has a requirement to send the *ownership* Event Report PDU as described in item e) in 5.9.4.2.1.

c) *Receipt of the Entity State PDU*

When an Entity State PDU is received by a simulation involved in a transfer transaction, the following requirements shall be met. Any reference to Entity State PDU shall also apply to the Environmental Process PDU unless otherwise specified:

1) If the simulation receiving the Entity State PDU is the acquiring simulation, it shall continue to process remote entity state data until it has established the entity as a local entity. Once it is established as a local entity, received entity information shall be ignored.
2) If the simulation receiving the Entity State PDU is the divesting simulation, it shall cease transmission of the Entity State (or Entity State Update) PDU and any other PDUs [e.g., Identification Friend or Foe (IFF) PDU] that have local data for the entity.

H.2.8 Data Query PDU

The Data Query PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.6.5.9 and 7.5.9. When used in support of transfer ownership, the Data Query PDU shall be used by a simulation to request ownership information for an entity. When used for this purpose, it is referred to as an *ownership* Data Query PDU. Only the requirements related to its use in support of transfer ownership are specified here:

a) *Information Content*

The information contained in an *ownership* Data Query PDU shall be as shown in Table H.7.

Table H.7—*ownership* Data Query PDU

Field	Subfield	Value
PDU Header	Standard	As appropriate.
Originating ID	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0
Receiving ID	Site Number Application Number Entity Number	The Entity ID of the entity being queried.
Request ID		Identifies this specific Data Query PDU request. Starts at 0 and incremented by one for each succeeding Data Query PDU sent after a simulation is initialized.
Time Interval		0
Number of Fixed Datums		0
Number of Variable Datums		1
Variable Datum ID #1	Variable Datum ID	15 800 = Ownership Status

b) *Issuance*

Any simulation may issue an *ownership* Data Query PDU whether or not it has implemented the Transfer Ownership function. This request may be automatically generated or initiated by manual operator action.

The Receiving ID field identifies the specific entity for which ownership information is being requested.

c) *Receipt*

Each simulation that has implemented the transfer ownership function shall process the *ownership* Data Query PDU and respond with an *ownership* Data PDU. If a simulation has not implemented the transfer ownership function, this PDU shall be ignored.

H.2.9 Data PDU

The Data PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.6.5.11 and 7.5.11. When used in support of transfer ownership, the Data PDU shall be used by the present owner of an entity that has implemented transfer ownership to respond to an *ownership* Data Query PDU. Only the requirements related to its use in support of transfer ownership are specified here:

a) *Information Content*

The information contained in an *ownership* Data PDU shall be as shown in Table H.8.

Table H.8—*ownership* Data PDU

Field	Subfield	Value
PDU Header		As applicable
Originating ID	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0
Receiving ID	Site Number Application Number Entity Number	Data Query Site Number Data Query Application Number 0
Request ID		Set to the Request ID contained in the *ownership* Data Query PDU to which this Data PDU is responding.
Padding		0
Number of Fixed Datum Records		0
Number of Variable Datum Records		1
Variable Datum record #1	Variable Datum ID	15 800 = Ownership Status
	Variable Datum Length	64
	Variable Datum Value = Ownership Status record (see 6.2.65)	Entity ID of the transferred entity
		2 = Ownership Query Response
		0 (8 bits unused padding)

b) *Issuance of the Data PDU*

The *ownership* Data PDU shall be issued in response to the receipt of an *ownership* Data Query PDU.

c) *Receipt of the Data PDU*

A simulation that has sent an *ownership* Data Query PDU shall process any Data PDU where it is the Receiving ID and the Request ID equals the Request ID contained in the original Data Query PDU. A simulation may also process a Data PDU not addressed to it where the Variable Datum ID = 15 800.

H.2.10 Record Query-R PDU

The Record Query-R PDU is a nonunique PDU used by transfer ownership and other DIS functions. It is described in 5.12.4.14 and 7.11.14. When used in support of transfer ownership, the Record Query-R PDU shall be used by a simulation when requesting internal state data on an entity in preparation for deciding whether to initiate or respond to a transfer request. A simulation that implements transfer ownership is not required to either initiate or respond to a Record Query-R PDU if it has no requirement to request internal state data outside of a transfer transaction. Any request for internal state data not related to a transfer transaction is beyond the scope of this standard:

a) *Information Content*

The information contained in a Record Query-R PDU shall be as shown in Table H.9.

Table H.9—Record Query-R PDU

Field	Subfield	Value
PDU Header	Standard	As appropriate
Originating ID	Site Number Application Number Entity Number	Originating Site Number Originating Application Number 0
Receiving ID	Site Number Application Number Entity Number	Queried Site Number Queried Application Number Queried Entity Number
Request ID		Identifies this specific Record Query-R PDU request. Starts at 0 and incremented by one for each succeeding Record Query-R PDU sent after a simulation is initialized.
Required Reliability Service		1 = Unacknowledged
Padding		0
Event Type	16-bit enumeration	1 = Internal Entity State Data
Number of Records	32-bit unsigned integer	
Record ID #1	32-bit enumeration	
Record ID #n	32-bit enumeration	

b) *Issuance of the Record Query-R PDU*

When the Record Query-R PDU is used to request internal state data related to a (potential) transfer transaction, it shall be for a specific entity, be addressed to the entity, and contain the Record IDs for the data it is requesting. It shall be sent once. If no Record-R PDUs are received in response to the query within a simulation's timeout parameter, it shall be assumed that the simulation to which it is addressed does not have the capability to provide internal state data related to a pending transfer transaction.

c) *Receipt of the Record Query-R PDU*

If the simulation to which the request is directed implements the receipt of this PDU for the purpose of providing internal state data for an entity it owns, it shall respond with the requested data, if available, in one or more Record-R PDUs. If data is not available for a requested Record ID, the Record ID shall be returned with zero values in the data fields for that record.

Annex I

(normative)

Articulated and attached parts

I.1 Scope

This annex defines the mechanism by which the state of articulated and attached parts shall be communicated. The particular parts that are to be represented as articulated rather than as fixed are not specified by this standard. The Articulated Part VP and Attached Part VP records are Variable Parameter records.

I.2 Articulated parts and attached parts

I.2.1 General

The articulated parts of an entity are represented by Articulated Part VP records as defined in 6.2.94.2 and as described in I.2.2. Attached parts of an entity are represented by Attached Part VP records as defined in 6.2.94.3 and as described in I.2.3. A part that is both attached and articulated is considered an articulated part as described in I.2.4.

I.2.2 Articulated parts

I.2.2.1 General

The Articulated Part VP record is used to represent the state of the movable parts of an entity. Examples of movable parts include the turret on a tank and the periscope on a submarine. An Articulated Part VP record shall represent the value of only one parameter of a movable, or articulated, part. Thus, it may require multiple Articulated Part VP records to describe the state of a single articulated part. The number of Articulated Part VP records used to represent a given articulated part on a given entity shall be determined at exercise initialization. This number shall not change for the duration of the simulation exercise. The total number and type of articulated parts associated with an entity shall not change during the execution of the exercise, except as noted in I.2.4. The Parameter Type field of the Articulated Part VP record shall identify both the Type Class (a particular movable part) and the Type Metric (state parameter) whose value is contained in the Parameter Value field (see I.2.2.4). The 16 currently defined type metrics (state parameters) are described in I.2.2.5.

I.2.2.2 Numbering of articulated parts

Each of an entity's articulated parts shall be sequentially assigned a part identification (ID). The entity itself shall be assigned a part ID of zero. An articulated part shall have a part ID greater than that of the articulated part to which it is attached. The part IDs associated with an entity shall be a continuous sequence starting with zero. The part identification is used in the ID—Part Attached To field of the Articulated Part VP record (see 6.2.94.2).

The part ID associated with an Articulated Part VP record is expressed by its order within an Entity State Protocol Data Unit (PDU) or Detonation PDU. The first Articulated Part VP record in a PDU is associated with part ID 1. For subsequent Articulated Part VP records, the associated part ID is incremented whenever the type class (see I.2.2.6) in the record changes. See I.2.2.9 for an example.

I.2.2.3 Reference and part coordinate systems

Each articulated part has an associated part coordinate system. This coordinate system is fixed with respect to the part. Each attached articulated part also has an associated reference coordinate system for an articulated part. This coordinate system is fixed with respect to the part that the articulated part is attached to. Both the part and reference coordinate systems are right-handed Cartesian coordinate systems. Both coordinate systems are defined when the entity type is defined.

The collection of Parameter Type and Parameter Value fields for a part in a PDU define the transformation from the reference coordinate system to the part coordinate system for an articulated part. In general, the reference and part coordinate systems should be defined such that a null transformation corresponds to the part in its neutral position.

I.2.2.4 Parameter Type field

The Parameter Type field of an Articulated Part VP record is composed by adding the Type Metric enumeration to the Type Class enumeration (see I.2.2.9 for an example):

Parameter Type = Type Class + Type Metric

The Type Metric enumeration determines which of the transformations described in I.2.2.5 are specified by this parameter type. The Type Class enumeration specifies which particular articulating part of a model is referenced by this parameter type. The values of the Type Class enumerations are defined in [UID 59], and Type Metric enumerations are defined in [UID 58].

The maximum value of the Type Metric shall be 31. The Type Class value shall be a multiple of 32. Upon receipt, these values can be recovered from the Parameter Type field as follows:

Type Metric = Parameter Type modulo 32

Type Class = Parameter Type - Type Metric

I.2.2.5 Type metrics

I.2.2.5.1 General

The Type Metric enumeration uniquely identifies the transformation to be applied to the articulated part.

I.2.2.5.2 Position (1)

Position shall specify the location of an articulated part along a particular path to which its movement is constrained. The path may be any three-space curve. The position shall be expressed in 32-bit floating point numbers. The value zero shall represent fully retracted, and one shall represent fully extended. Intermediate positions are represented as a fraction of the path traveled. One path may be associated with each articulated part on each entity type.

I.2.2.5.3 Position rate (2)

Position rate shall specify the rate of change of position in units of fraction of entire path per second. For example, a position rate of one indicates that the articulated part has traversed the entire path in 1 s. Position rate shall be expressed in 32-bit floating point format.

I.2.2.5.4 Extension (3)

Extension shall specify the linear extension of the part in one direction in meters. The value zero shall represent fully retracted. Extension shall be expressed in 32-bit floating point format.

I.2.2.5.5 Extension rate (4)

Extension rate shall specify the rate of change of extension in units of meters per second. The extension rate shall be expressed in 32-bit floating point format.

I.2.2.5.6 *x* (5), *y* (7), and *z* (9)

The *x*, *y*, and *z* shall specify the translation from the articulated parts reference coordinate system to the current location of the articulated parts coordinate system. The *x*, *y*, and *z* shall be expressed in 32-bit floating point format.

I.2.2.5.7 *x* rate (6), *y* rate (8), and *z* rate (10)

The *x*, *y*, and *z* rates shall specify the rate of change of the position of the articulated coordinate system expressed in meters per second. The *x*, *y*, and *z* rates shall be expressed in 32-bit floating point format.

I.2.2.5.8 Azimuth (11)

Azimuth shall specify the rotation of an articulated part with respect to its reference *z*-axis. Azimuth shall be measured in radians and shall be expressed in 32-bit floating point format.

I.2.2.5.9 Elevation (13)

Elevation shall specify the rotation of an articulated part with respect to its reference *y*-axis. Elevation shall be measured in radians and shall be expressed in 32-bit floating point format.

NOTE—It is not necessary to constrain the elevation angle to $\pm\pi/2$.

I.2.2.5.10 Rotation (15)

Rotation shall specify the rotation of an articulated part with respect to its reference *x*-axis. Rotation shall be measured in radians and shall be expressed in 32-bit floating point format.

I.2.2.5.11 Order of transformation

If more than one of azimuth, elevation, or rotation exists for the same articulated part, the order of the transformation shall be azimuth, then elevation, and then rotation.

I.2.2.5.12 Azimuth rate (12), elevation rate (14), and rotation rate (16)

These rates shall specify the rate at which the angle is changing. Angular rates shall be measured in radians per second and shall be expressed in 32-bit floating point format. These rates represent instantaneous angular velocity.

I.2.2.6 Type class

The Type Class enumeration uniquely identifies a particular articulated part on a given entity type. For an example of uniquely assigning type classes to an entity's articulated parts, see I.2.2.9.

I.2.2.7 Dead reckoning of articulated parts

The issuing simulation application may specify the dead reckoning of an articulated part by including position, extension, *x*, *y*, *z*, azimuth, elevation, and/or rotation rate type metrics in an Articulated Part VP record of an Entity State PDU (see 5.3.2). The same dead reckoning thresholds (see Table 5) that apply to an entity apply to its articulated parts. Note that angular movement of an articulated part may result in a linear threshold being exceeded. Absolute linear error for each articulated part in a chain of connected parts shall be used to determine whether a threshold has been exceeded.

Dead reckoning of articulated parts is done with a first-order extrapolation as follows:

$$P = P_0 + R\Delta t$$

where P is the dead-reckoned linear or angular value of the articulated part type metric, P_0 is the original value in the Articulated Part VP record of the Entity State PDU, and R is the associated rate in the PDU.

I.2.2.8 Guidelines for selecting type metrics

Although it is possible to define part and reference coordinate systems so that almost any type metric could apply to almost any articulated part, Table I.1 provides a guideline for which type of metrics should be used with some common articulated parts. Rate parameters shall be used when dead reckoning of articulated parts is required.

Table I.1—Guidelines for selecting type metrics

Part	Recommended type of metric
Horizontal control surfaces	Elevation
Vertical control surfaces	Azimuth
Extendible items	Extension
Fixed path items	Position
Turrets	Azimuth
Guns	Elevation
Movable missile launcher	Azimuth and elevation

I.2.2.9 Articulated part example—tank main gun

To represent the articulation of the main (primary) gun on a tank (or howitzer), the azimuth of the turret with respect to the tank is required as well as the elevation of the gun with respect to the turret. This represents two articulated parts (Type Classes), each with two parameters (Type Metrics). A more detailed example would also represent one or more additional secondary guns, but for the sake of simplicity, this example does not include secondary weapons or other articulated or attached parts. Table I.2 illustrates the actual parts.

Table I.2—Articulated parts of a tank

Part number	Part	Part Attached to	Notes
1	Turret	Chassis (0)	Attached directly to the tank
2	Main Gun	Turret (1)	Attached to the turret, inherits turret rotation

In this example, we wish to dead reckon the position of the turret, so we require two articulation parameters (Type Metrics): azimuth and azimuth rate. We also wish to dead reckon the position of the gun, so we require two articulation parameters for it as well: elevation and elevation rate. These four articulation parameters therefore correspond to four separate Articulated Part VP records as illustrated below in Table I.3. This table provides exemplary values for the Change Indicator and Parameter Value fields and identifies the correct Parameter Types for each of the records and how those fields were computed. Note specifically how each Change Indicator for the different VP records are independent and may be different. Note also how the Turret Azimuth Rate is not a separate ID—Part Attached to from Turret Azimuth. Thus, the correct values (as shown) for the ID—Part Attached to for the Primary Gun Elevation and Primary Gun Elevation Rate is 1.

Table I.3—Articulated Part VP records of a tank

Record	Field name	Value	Description
Turret Azimuth	Record Type	0	Articulated Part VP record
	Change Indicator	213	Incremented by one for each change
	ID—Part Attached to	0	Tank chassis
	Parameter Type	4107	4096 (primary turret) + 11 (azimuth)
	Parameter Value	–0.305	Angle in radians
Turret Azimuth Rate	Record Type	0	Articulated Part VP record
	Change Indicator	45	Incremented by one for each change
	ID—Part Attached to	0	Tank chassis
	Parameter Type	4108	4096 (primary turret) + 12 (azimuth rate)
	Parameter Value	–0.058	Rate in radians/s

Table I.3—Articulated Part VP records of a tank *(continued)*

Record	Field name	Value	Description
Primary Gun Elevation	Record Type	0	Articulated Part VP record
	Change Indicator	187	Incremented by one for each change
	ID—Part Attached to	1	Turret
	Parameter Type	4429	4416 (primary gun) + 13 (elevation)
	Parameter Value	0.267	Angle in radians
Primary Gun Elevation Rate	Record Type	0	Articulated Part VP record
	Change Indicator	34	Incremented by one for each change
	ID—Part Attached to	1	Turret
	Parameter Type	4430	4416 (primary gun) + 14 (elevation rate)
	Parameter Value	0.384	Rate in radians/s

Continuing this example, let us suppose that for the next Entity State PDU update, the main gun elevation has reached its final position (as shown in Table I.3). Thus, in this next update, the elevation has not changed and the rate has dropped to zero. Additionally, the azimuth has updated smoothly during a 2 s interval, but the rate has not changed. This exemplary set of VP records is therefore shown in Table I.4. Note which Change Indicator fields change and which ones do not.

Table I.4—Articulated Part VP records of a tank (stationary gun)

Record	Field name	Value	Description
Turret Azimuth	Record Type	0	Articulated Part VP record
	Change Indicator	214	Incremented by one for each change
	ID—Part Attached to	0	Tank chassis
	Parameter Type	4107	4096 (primary turret) + 11 (azimuth)
	Parameter Value	-0.421	Angle in radians
Turret Azimuth Rate	Record Type	0	Articulated Part VP record
	Change Indicator	45	Incremented by one for each change
	ID—Part Attached to	0	Tank chassis
	Parameter Type	4108	4096 (primary turret) + 12 (azimuth rate)
	Parameter Value	–0.058	Rate in radians/s

Table I.4—Articulated Part VP records of a tank (stationary gun) *(continued)*

Record	Field name	Value	Description
Primary Gun Elevation	Record Type	0	Articulated Part VP record
	Change Indicator	187	Incremented by one for each change
	ID—Part Attached to	1	Turret
	Parameter Type	4429	4416 (primary gun) + 13 (elevation)
	Parameter Value	0.267	Angle in radians
Primary Gun Elevation Rate	Record Type	0	Articulated Part VP record
	Change Indicator	35	Incremented by one for each change
	ID—Part Attached to	1	Turret
	Parameter Type	4430	4416 (primary gun) + 14 (elevation rate)
	Parameter Value	0.0	Rate in radians/s

I.2.3 Attached parts

I.2.3.1 General

The Attached Part VP record represents removable parts that may be attached to an entity. Examples of such attached parts include missiles and other external stores on an aircraft. The following requirements shall be met:

a) The Parameter Type field shall identify the location (or station) to which the part is attached. Stations shall be assigned to each entity type as described in I.2.3.2.

b) The Attached Part Type field shall contain the entity type of the attached part. Entity types are defined in [UID 30].

c) When an attached part becomes detached, the following shall occur:

   1) An Entity State PDU shall be immediately issued with the Detached Indicator set to Detached (1).

   2) If an entity is created for the detached part, an *initial* Entity State PDU for it shall be issued simultaneously with the issuance of the update of the Entity State PDU to reflect the detached status.

   3) When the next Entity State PDU is issued for the entity that had a detached part, the Attached Part VP record shall no longer be included in the PDU.

I.2.3.2 Stations

The station to which an attached part is attached is identified by the Parameter Type enumeration of the Attached Part VP record. Stations shall be numbered sequentially beginning with one and incrementing by one. The order of numbering shall be from top to bottom, then back to front, and then left to right. The only exception shall be aircraft wing stations. The fuselage stations, left wing stations, and right wing stations shall be separated into different categories. Wing stations shall be numbered from inboard to outboard. The Attached Part Type field of the Attached Part VP record shall contain the Entity Type record representing the type of store located at the specified station. The attached part Parameter Type enumerations (stations) are defined in [UID 57].

I.2.3.3 Attached part example—air-to-surface missiles on aircraft

To represent air-to-surface missiles on an aircraft, it is necessary to specify the station (Parameter Type) to which each of the missiles is attached and the type of each missile. In this simple example, we have two similar missiles, one on the first station of each wing.To represent the attached missiles, an initial Entity State PDU would be sent with Attached Part VP records as shown in Table I.5. The Detached Indicator shows that both missiles are attached and the ID—Part Attached to fields show that they are both attached to the aircraft chassis. The Parameter Type indicates the station, and the table shows the range of stations for the left and right wings. The Attached Part Type field identifies the specific type of munition for the attached part.

Table I.5—Attached Part VP records of air-to-surface missiles on an aircraft

Record	Field name	Value	Description
AGM-158A JASSM Missile	Record Type	1	Attached Part VP record
	Detached Indicator	0	Attached
	ID—Part Attached to	0	Aircraft chassis
	Parameter Type	640	First left-wing station (640–767)
	Attached Part Type	2.9.225.1.26.1.0	Type of missile as a 64-bit Entity Type
AGM-158A JASSM Missile	Record Type	1	Attached Part VP record
	Detached Indicator	0	Attached
	ID—Part Attached to	0	Aircraft chassis
	Parameter Type	768	First right-wing station (768–895)
	Attached Part Type	2.9.225.1.26.1.0	Type of missile as a 64-bit Entity Type

When the missile on the left-wing is fired and therefore becomes detached, a subsequent Entity State PDU would be sent with an Attached Part VP record as shown in Table I.6. Note that before the VP record is removed from the PDU, the final record must contain an appropriately changed Detached Indicator.

Table I.6—Attached Part VP records of a launched air-to-surface missile on an aircraft

Record	Field name	Value	Description
AGM-158A JASSM Missile	Record Type	1	Attached Part VP record
	Detached Indicator	1	Detached
	ID—Part Attached to	0	Aircraft chassis
	Parameter Type	640	First left-wing station (640–767)
	Attached Part Type	2.9.225.1.26.1.0	Type of missile as a 64-bit Entity Type
AGM-158A JASSM Missile	Record Type	1	Attached Part VP record
	Detached Indicator	0	Attached
	ID—Part Attached to	0	Aircraft chassis
	Parameter Type	768	First right-wing station (768–895)
	Attached Part Type	2.9.225.1.26.1.0	Type of missile as a 64-bit Entity Type

The very next Entity State PDU would then have then the Attached Part VP record simply as shown in Table I.7.

Table I.7—Attached Part VP records after launched air-to-surface missile on an aircraft

Record	Field name	Value	Description
AGM-158A JASSM Missile	Record Type	1	Attached Part VP record
	Detached Indicator	0	Attached
	ID—Part Attached to	0	Aircraft chassis
	Parameter Type	768	First right-wing station (768–895)
	Attached Part Type	2.9.225.1.26.1.0	Type of missile as a 64-bit Entity Type

I.2.4 Attached articulated parts

I.2.4.1 General

A part that is both attachable and articulated shall be represented as an articulated part (see I.2.2). Examples of such parts include an attachable mine plow on a tank and the hinged, deployable bridge on the U.S. Armored Vehicle Launched Bridge (AVLB). The part ID (see I.2.2.2) of an attached articulated part shall exceed that of any ordinary articulated part. In other words, all attached articulated parts shall always be at the end of the list of articulated parts. As a result, the addition or removal of one or more attached articulated parts shall not change the part ID of any other articulated part.

The total number of articulated parts may change during an exercise only due to the addition or removal of attached articulated parts.

I.2.4.2 Attached articulated part example—M60A1 AVLB

To represent the articulation of the bridge on an M60A1 AVLB, the elevation of the bridge launcher with respect to the chassis is required as well as the elevation of the two bridge segments. This represents three articulated parts (Type Classes), each with two parameters (Type Metrics). Table I.8 illustrates the actual parts. The bridge launcher is an articulated part on the M60A1 AVLB that does not detach. Its default position is on top of the chassis, and it rotates forward and down to deploy the bridge. The first bridge section (detachable) also rotates forward and down. The second bridge section is not detachable from the first bridge section, and it rotates forward and up. The ranges of rotations are provided in Table I.9.

Table I.8—Articulated parts of the M60A1 AVLB

Part number	Part	Part Attached to	Notes
1	Bridge launcher	Chassis (0)	Attached directly to the chassis of the vehicle
2	Bridge section 1	Bridge launcher (1)	Attached to the bridge launcher, inherits the bridge launcher rotation
3	Bridge section 2	Bridge section 1 (2)	Attached to bridge section 1, inherits bridge section 1 rotation

In this example, we wish to dead reckon the position of the bridge launcher and each bridge section, so we require two articulation parameters (Type Metrics) each: elevation and elevation rate. These six articulation parameters therefore correspond to six separate Articulated Part VP records as illustrated below in Table I.9. This table provides exemplary values for the Change Indicator and Parameter Value fields and identifies the correct Parameter Types for each of the records and how those fields were computed. Note specifically how each Change Indicator for the different VP records is independent and may be different. In this example, the bridge launcher is fully extended and thus no longer moving. Therefore, the Parameter Value for the bridge launcher elevation rate is 0. This represents the actual model because the bridge launcher must be fully extended (in place) before the bridge sections are articulated.

Table I.9—Articulated Part VP records of the M60A1 AVLB

Record	Field name	Value	Description
Bridge Launcher Elevation	Record Type	0	Articulated Part VP record
	Change Indicator	87	Incremented by one for each change
	ID—Part Attached to	0	AVLB chassis
	Parameter Type	3309	3296 (bridge launcher) + 13 (elevation)
	Parameter Value	–1.57 (–π/2)	Angle in radians (range is 0 to –π/2)
Bridge Launcher Elevation Rate	Record Type	0	Articulated Part VP record
	Change Indicator	45	Incremented by one for each change
	ID—Part Attached to	0	AVLB chassis
	Parameter Type	3310	3296 (bridge launcher) + 14 (elevation rate)
	Parameter Value	0.0	Rate in radians/s (not moving)
Bridge Section 1 Elevation	Record Type	0	Articulated Part VP record
	Change Indicator	67	Incremented by one for each change
	ID—Part Attached to	1	Bridge launcher
	Parameter Type	3341	3328 (bridge section 1) + 13 (elevation)
	Parameter Value	–1.119	Angle in radians (range is 0 to –π/2)
Bridge Section 1 Elevation Rate	Record Type	0	Articulated Part VP record
	Change Indicator	34	Incremented by one for each change
	ID—Part Attached to	1	Bridge launcher
	Parameter Type	3342	3328 (bridge section 1) + 14 (elevation rate)
	Parameter Value	–0.010	Rate in radians/s
Bridge Section 2 Elevation	Record Type	0	Articulated Part VP record
	Change Indicator	38	Incremented by one for each change
	ID—Part Attached to	2	Bridge section 1
	Parameter Type	3373	3360 (bridge section 2) + 13 (elevation)
	Parameter Value	2.238	Angle in radians (range is 0 to –π/2)

Table I.9—Articulated Part VP records of the M60A1 AVLB *(continued)*

Record	Field name	Value	Description
Bridge Section 2 Elevation Rate	Record Type	0	Articulated Part VP record
	Change Indicator	12	Incremented by one for each change
	ID—Part Attached to	2	Bridge section 1
	Parameter Type	3374	3360 (bridge section 2) + 14 (elevation rate)
	Parameter Value	0.021	Rate in radians/s

Because articulated parts and attached articulated parts are all represented with the Articulated Part VP record, all of these articulation parameters in Table I.9 are given using a Record Type of 0. What may not be clear in this example is that the first bridge segment may detach from the bridge launcher, but the bridge launcher does not detach from the chassis and neither does the second bridge section detach from the first. If, for example, an articulated gun that is not an attached part were available on this entity (not on the bridge), then the articulated parameters for that gun would need to be placed before the bridge segments 1 and 2, although not necessarily before the bridge launcher. The bridge launcher is therefore an articulated part and bridge section 1 is an attached articulated part. Bridge section 2 is an articulated part with respect to bridge section 1, but it is an attached articulated part with respect to the AVLB chassis. When the bridge detaches, only the first two Articulated Part VP records would be present in the Entity State PDU for the M60A1 AVLB. The bridge itself would then become a separate object in the battlespace and no longer be represented in the M60A1 AVLB Entity State PDU.

Annex J

(informative)

Bibliography

Bibliographical references are resources that provide additional or helpful material but do not need to be understood or used to implement this standard. Reference to these resources is made for informational use only.

These references, unless prohibited by copyright, are also available on the Simulation Interoperability Standards Organization (SISO) website.[12]

[B1] Arkfen, G. B., *Mathematical Methods for Physicists*. New York: Academic Press, 1966.

[B2] Burchfiel, J., "The advantage of using quaternions instead of Euler angles for representing orientation," *Third Workshop on Standards for the Interoperability of Defense Simulations*, vol. 7, Orlando, FL, pp. 66–84, Aug. 1990.[13]

[B3] Goel, S., and Morris, K. D., "Dead reckoning for aircraft in distributed interactive simulation," *AIAA/AHS Flight Simulation Technologies Conference*, Hilton Head, SC, pp. 277–284, Aug. 1992.[14]

[B4] IEEE Std 1278.3™-1996, IEEE Recommended Practice for Distributed Interactive Simulation—Exercise Management and Feedback.[15, 16]

[B5] IEEE Std 1730™-2010, IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP).

[B6] Joint Staff J-8, Deputy Director C4, Assistant Deputy Director for Command and Control Integration (ADD C2I), Joint Fires Division (JFD), "(U) Mark XIIA Mode 5 and Mode Select (Mode S) Joint Concept of Operations (Joint CONOPS)," 4 Nov. 2011.

[B7] Katz, A., "Synchronization of networked simulators," *Eleventh Workshop on Standards for the Interoperability of Defense Simulations*, Orlando, FL, pp. 81–88, Sept. 1994.[17]

[B8] Katz, A., "The absolute clock in the DIS scheme," *Tenth Workshop on Standards for the Interoperability of Defense Simulations*, Orlando, FL, pp. 1–4, Mar. 1994.

[B9] Katz, A., and Graham, K., "Dead reckoning for airplanes in coordinated flight," *Tenth Workshop on Standards for the Interoperability of Defense Simulations*, Orlando, FL, pp. 5-13, Mar. 1994.

[12]The Simulation Interoperability Standards Organization hosts a repository of papers, guidance documents, and standards related to distributed interactive simulation and simulation interoperability. The files of the Distributed Interactive Simulation (DIS) Product Support Group (PSG) (http://www.sisostds.org/DigitalLibrary.aspx?EntryId=29286) are the best starting point. The Simulation Interoperability Standards Organization (SISO) website is located at http://www.sisostds.org. In the Digital Library, look at folders Support Groups > DIS PSG > IEEE 1278 Bibliography Material.

[13]This document is available at http://www.sisostds.org/DigitalLibrary.aspx?EntryId=31599.

[14]This document is available at http://www.aiaa.org/content.cfm?pageid=298. Search by Author Name and "dead-reckoning."

[15]IEEE publications are available from The Institute of Electrical and Electronics Engineers (http://standards.ieee.org/).

[16]The IEEE standards or products referred to in this clause are trademarks of The Institute of Electrical and Electronics Engineers, Inc.

[17]The documents in [B7] through [B9] are available at http://www.sisostds.org/DigitalLibrary.aspx?EntryId=31599.

[B10] Kuipers, J. B., *Quaternions and Rotation Sequences: A Primer with Applications to Orbits, Aerospace and Virtual Reality*. Princeton, NJ: Princeton University Press, 2002.

[B11] Lin, K. C., and Schab, D., "Study on the network load in distributed interactive simulation," *1994 AIAA Flight Simulation Technologies Conference*, Scottsdale, AZ, pp. 202–209, Aug. 1994.[18]

[B12] Murray, R. E., "Proposed PDU bundling specifications for IEEE 1278.2," *2009 Fall Simulation Interoperability Workshop*, Orlando, FL, Paper 09F-SIW-139, Sept. 2009.

[B13] Saunders, R., "Basic Mathematics for DIS applications," *Tenth Workshop on Standards for the Interoperability of Defense Simulations*, Orlando, FL, pp. 463–470, Mar. 1994.

[B14] Saunders, R., "Formal expression of dead reckoning: Mathematical and representation recommendations," *The Fifth Workshop on Standards for the Interoperability of Defense Simulations*, Orlando FL, pp. A-133–A-138, Sept. 1991.

[B15] Saunders, R., "It's about time, it's about space—time and space in DIS," *Twelfth Workshop on Standards for the Interoperability of Defense Simulations*, vol. 1, Orlando, FL, pp. 63–66, Mar. 1995.

[B16] Schaffer, R., and Waters, R., "Dead reckoning algorithms and the simulation of high performance aircraft," *Fourth Workshop on Standards for the Interoperability of Defense Simulations*, Orlando, FL, pp. 233–244, Mar. 1991.

[B17] Towers, J., and Hines, J., "Equations of motion of the DIS 2.0.3 dead reckoning algorithms," *Tenth Workshop on Standards for the Interoperability of Defense Simulations*, Orlando, FL, pp. 431–462, Mar. 1994.

[18]The documents in [B11] through [B17] are available at http://www.sisostds.org/DigitalLibrary.aspx?EntryId=31599.