

MV 3500 Internetwork Communications and Simulation (3-2)

July-September 2021

Synopsis	An introduction to network communications in simulation applications. Topics include an introduction to the TCP/IP protocol stack; TCP/IP socket communications, including TCP, UDP, and multicast; and protocol design issues, with emphasis on Distributed Interactive Simulation (DIS) Protocol and High Level Architecture (HLA). Course emphasis is on creation and testing of network programming network code and web-browser applications. Prerequisites: CS2173 or equivalent (Java programming).		
Instructor	Don Brutzman Office Watkins 270	brutzman@nps.edu http://faculty.nps.edu/brutzman	1.831.656.2149 work 1.831.402.____ cell
	Office hours are anytime available, as indicated by my online schedule in Outlook. Make an appointment if you want to be sure to see me. If needed please call me directly.		
Schedule	Meeting times are based on NPS student availability. Dates for your assignments and project demos are announced in advance. Savage Lab in Watkins 267 is available for use.		
Course Sites	NPS version control: https://gitlab.nps.edu/Savage/NetworkedGraphicsMV3500 Prior site CLE Sakai: MV3500 Internetwork Communications and Simulation MS Teams references: MV3500 Networked Simulation > Files > MV3500 Wikipedia References		
Software	Savage Developers Guide: Apache NetBeans 12.4, Java Development Kit (OpenJDK) 16 Wireshark, https://wireshark.org Open-DIS Project, https://github.com/open-dis open-dis7-java, open-dis7-source-generator X3D-Edit Authoring Tool, https://savage.nps.edu/X3D-Edit		
Optional Textbooks	<ol style="list-style-type: none"> 1. Networked Graphics: Building Networked Games and Virtual Environments, Anthony Steed and Manuel Fradinho Oliveira, Elsevier, 536 pages, 2009. Chapter 7 discusses X3D graphics, X3D-Edit authoring tool, and IEEE DIS networking protocol. 2. Networked Virtual Environments, Sandeep Singhal and Michael Zyda, ACM Press SIGGRAPH Series, Addison Wesley, 2009. 3. Modeling and Simulation: Linking Entertainment and Defense, National Research Council (NRC), 1997, Michael Zyda chair and editor. 3. X3D for Web Authors, Don Brutzman and Leonard Daly, Morgan Kaufmann, 2007. 		
Guidelines	<ol style="list-style-type: none"> 1. You must devote time to reading and coding in order to succeed in this course. 2. Students are encouraged to work together. However all submitted assignment must include your own work. Group solutions to project assignments are acceptable when so credited. As in any endeavor, individual integrity is essential. If in doubt, ask. 3. This course has potential to significantly help you in your thesis and other courses. Thus your comments, questions and suggestions are worthwhile and always welcome. 4. Most course communications are in our Team channel. For course email, please include "MV3500" in the subject line so that email filters can locate it satisfactorily. 		

Course Objectives

1. Understand network programming strengths and limitations for distributed simulation.
2. Improve your programming skills, becoming sufficiently competent to knowledgeably supervise programming teams in future assignments.
3. Support your **thesis work** and projects in other classes.
4. Use tools, techniques and a repeatable methodology that can help you later in your career.

Class Policy and Study Recommendations

1. You are learning new ideas and a new vocabulary. Thinking and writing in a new language requires fluency. Don't be reluctant to think new thoughts or work hard. Persistence pays.
2. Discussion and dialog will make class material a lot more immediate.
3. Projects make up your entire grade, just like real world. Exams are *boring* (harsh label).
4. Grading is based on merit and performance. I expect everyone to work hard and get an A.
5. You learn how to program solutions to problems by doing. Thus we do lots of coding and projects. The key for me is to see that you are progressing steadily through hands-on experience. Your final project should pass the “quantitatively cool” test among all of us.
6. Students are expected to check in projects on time. It is your responsibility to contact me in advance for assistance if you are unable to meet an assignment date. If your work is unfinished at deadline, check it in and keep working. If needed: finalize and fix something even if late, since spiral improvement is much better than waterfall crashing.
7. The highest levels of honesty and professional integrity are expected of military leaders. See [NPS Academic Honor Code NPSINST 5370.4D](#) for a detailed list of your responsibilities.

Candidate Projects

- As the course progresses, we consider a variety of projects that can contribute to shared live virtual constructive (LVC) scenarios that utilize distributed virtual environments.
- What are your challenges of interest, for your thesis and for your career? Let's discuss it, there are many opportunities available and I am always happy to help.

MV 3500 Internetwork Communications and Simulation

Week	Dates	Topics	Notes
1	2021 6-9 JUL	1. Introduction and getting started, TCP/IP networking	Network interoperability across programming languages. Netbeans installation.
2	12-16 JUL	2. Simulation Networking Standards and the Web	Assignment 1 due
3	19-23 JUL	3. TCP/IP Sockets in Java	ITACS tour planned, future once masking relaxed. Alternate: secure networking VPN.
4	26-30 JUL	4. TCP Sockets in Java Curt Blais: Rich Semantic Track (RST)	Install/run Wireshark, watch video
5	2-6 AUG	5. Java Input/Output (I/O) “Cheat Sheet” presentation 6. User Datagram Protocol (UDP)	Assignment 2 due (unicast) Testing argon VPN (maybe next quarter)
6	9-13 AUG	7. Network Scalability, Open-DIS Overview 8. Distributed Interactive Simulation (DIS) Protocol	Final project discussions Install X3D-Edit (pending rebuild)
7	16-20 AUG	Considering DIS applications 9. High-Level Architecture (HLA)	Assignment 3 started: DIS comms by adapting OpenDis7Examples - ExampleSimulationProgram
8	23-27 AUG	11. Network Security, Norbraten: certificates, PKI, etc. Cloud computing, Simulation as a Service (SAAS) 27 AUG Loren Peitso: defeating lag in networked VEs	Assignment 3 due: example program using DIS Assignment 4: OpenDIS Servlet , DIS assessment
9	30 AUG – 3 SEP	9. High-Level Architecture (HLA) continued 10. Test and Training Enabling Architecture (TENA)	Project design reviews by each student/team. Using DIS PDU recorder, playback.
10	6-10 SEP	Project testing, finish Network Security lectures	Final project demos
11	13-17 SEP	Exam period this week: project demos	Overflow period if needed

Homework Assignments and Grading Criteria

Percent	Task	Summary
15%	Assignment 1	Update unicast sockets sender/receiver, modifying provided code. Demonstrate ability to build, run and document Java programs that perform networking tasks.
15%	Assignment 2	Modify provided code for unicast/multicast sockets sender/receiver. Demonstrate proficiency to build, run and document software that performs networking tasks.
15%	Assignment 3	Shared communications examples using NPS campus network (or possibly trusted argon VPN). Use ExampleSimulationProgram OpenDIS PDU example, modifying provided code, then recorded/saved/replayed using X3D-Edit or Wireshark.
15%	Assignment 4	HTTP servlet with DIS protocol . Class efforts, DIS assessment.
20%	Project: Source Code and Data Content	Write a DIS program that has a simple but meaningful behavior. Include DIS enumerations for specific player platforms of interest. Document the behavior and save the stream for possible future use. Example functional traits for your project are found on next page.
20%	Project Documentation and Class Presentation	<i>Project Attributes</i> are listed on next page. You are undergoing peer review with your current and future classmates. We expect that your presented, recorded, repeatable work is quantitatively cool!
100%	gitlab version control	Durable outcomes: this work helps you to perform future duties as a MOVES graduate, helps your efforts in the follow-on practicum course, and also helps future students “look over your shoulder” to observe good practices and continue building lessons learned.

Candidate Course Projects

Project attributes:

- Project preparation plans and progress reports are given in class, as announced. Be ready!
- Source code checked into course gitlab version control. Netbeans project or standalone.
- README.md description and directions, includes Javadoc for classes and methods.
- Simple slideset presentation including key points and generous use of screenshots.
- Class demonstration, 15-20 minutes. Include list of accomplishments plus future TODO items.
- Repeatable and useful, for your own theses and for future students. Quantitatively cool!

Direct continuation of course homework capabilities:

- 1. Multicast DIS bridge between LANS (2 people)**
 - a. Connect two LANs via unicast socket.
 - b. Read multicast DIS PDUs from each side, send to other side.
 - c. Maintain list of passed PDUs, filter duplicates to prevent infinite loops.
 - d. Create a recording of the stream, sorted and filtered. Report simple statistics.
- 2. Improve test cases to provide unit tests (2 people)**
 - a. Extend homework3/OpenDisPduSender to set example values for each available PDU method.
 - b. Confirm visually using X3D-Edit, WireShark, Java that values are passed (round-trip testing).
 - c. Create recording files for each test set, again using X3D-Edit PDU Player-Recorder.
 - d. Establish DIS “unit test” by checking results into git so that variations are detectable.
- 3. Improve OpenDIS documentation “Missing Manual” (potentially for all students)**
 - a. <https://github.com/open-dis/DISTutorial> in Markdown format.
 - b. Editorial corrections, note problems and list suggestions for future work.
 - c. Build script automated download provide the latest IEEE DIS specification as a reference.
 - d. Upgrade Wireshark configuration to support all IEEE DIS PDUs, just like new Open-DIS build
- 4. More elaborate, tactically interesting scenario for generating PDU packets**
 - a. Prepare for Chris Fitzpatrick practicum, Live Virtual constructive (LVC) network modeling.
 - b. Areas of interest: Monterey Bay, Camp Pendleton, Rota Spain, etc.

Advanced projects extending repertoire shown in course:

5. Assess use cases, add PDUs to John Furr’s Forward Observer (FO) thesis Java source code
6. Adapt a Unity or other program to emit PDUs using Open-DIS C# binding. Caveat programmer!
7. Read a GPS data file and generate ESPDU stream
8. Read a different data file (in a commonly used format) and generate ESPDU stream.
9. Refresh, test demonstration programs using GoogleMaps, OpenStreetMap using HTML/Javascript.
10. Refresh, update X3DOM OpenDIS example by Byron Harder using HTML/X3D/JavaScript.
 - a. Note separate research progress on this capability is leading to SPIDERS3D updates.
11. Support future practicum efforts: AIS streams to DIS, exemplar for SIMC2 NATO work, etc.
12. X3D Position/Orientation Interpolator smoothing/distillation/compression for stream playback.
13. “Your project here”... work supporting your thesis or professional interests is especially welcome.