

MV 3500 Internetwork Communications and Simulation (3-2)

July-September 2018

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: CS2971 and CS2173.		
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. Exams and project demo dates are announced in advance. Savage Lab in Watkins 267 is always available for your use.		
Course Sites	NPS CLE Sakai: MV3500 Internetwork Communications and Simulation NPS version control: https://gitlab.nps.edu/Savage/NetworkedGraphicsMV3500		
Software	Netbeans 8, http://netbeans.org Wireshark, https://wireshark.org Open-DIS Project, https://github.com/open-dis X3D-Edit Authoring Tool, https://savage.nps.edu/X3D-Edit		
Textbooks	<ol style="list-style-type: none"> 1. <i>Networked Graphics: Building Networked Games and Virtual Environments</i>, 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. <i>Networked Virtual Environments</i>, Sandeep Singhal and Michael Zyda, ACM Press SIGGRAPH Series, Addison Wesley, 2009. 3. <i>X3D for Web Authors</i>, Don Brutzman and Leonard Daly, Morgan Kaufmann, 2007, http://x3dgraphics.com 		
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 every assignment submitted must be 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 may significantly help you in your thesis and other courses. Your comments, questions and suggestions are always welcome. 4. 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 a lot more immediate.
3. Projects make up your entire grade, just like the real world. Exams are boring.
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.
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 5370.4C](#) for a detailed list of your responsibilities.

Candidate Projects

- As the course progresses, we will consider various projects that can contribute to a shared military scenario utilizing a distributed virtual environment.
- What is your challenge of interest? Let's discuss it.

MV 3500 Internetwork Communications and Simulation

Week	Dates	Topics	Notes
1	2-6 JUL	Introduction and getting started, TCP/IP networking	Install Netbeans, discuss programming languages
2	9-13 JUL	TCP/IP Sockets in Java	Assignment 1
3	16-20 JUL	Java Input/Output (I/O) User Datagram Protocol (UDP)	Install Wireshark
4	23-27 JUL	Network Scalability	Assignment 2
5	30 JUL - 3 AUG	Distributed Interactive Simulation (DIS) Protocol Open-DIS Overview, Don McGregor NPS	Install X3D-Edit Final project discussions
6	6-10 AUG	NATO C2SIM Project, Dr. J. Mark Pullen GMU	Guest lecture, instructor travel to ISO X3D meeting, France
7	13-17 AUG	SPIDERS3D collaborative VE, Alex Viana NAVFAC	Guest lecture, instructor travel to ACM SIGGRAPH, Vancouver
8	20-24 AUG	DIS applications High-Level Architecture (HLA)	Assignment 3
9	27-31 AUG	High-Level Architecture (HLA) continued Test and Training Enabling Architecture (TENA)	Project design reviews
10	3-7 SEP	Network security considerations Cloud computing, Simulation as a Service (SAAS)	ITACS tour, Project testing
11	10-14 SEP	Special-interest topics, class project demonstrations	

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. 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 that values are passed (i.e. round-trip testing).
- c. Create recording files for each test set, again using X3D-Edit PDU Player-Recorder.
- d. Establish “unit test” by checking results into git so that variations are detectable.

3. Improve OpenDIS documentation “Missing Manual”

- a. <https://github.com/open-dis/DISTutorial> in Markdown format.
- b. Editorial corrections, note problems and list suggestions for future work.
- c. Will provide the latest IEEE DIS specification as a reference.

4. More elaborate, tactically interesting scenario for generating PDU packets

- a. Preparation for practicum modeling
- b. Areas of interest: Monterey Bay, Camp Pendleton, Rota Spain, etc.

Advanced projects extending repertoire shown in course

5. Adapt a Unity or other program to emit PDUs using Open-DIS C# binding.
6. Read a GPS data file and generate ESPDU stream
7. Read a different data file (in a commonly used format) and generate ESPDU stream
8. Refresh, test demonstration programs using GoogleMaps, OpenStreetMap using HTML/Javascript.
9. Refresh, test X3DOM OpenDIS example by Byron Harder using HTML/X3D/JavaScript.
 - a. Support by Brutzman and McGregor. Note progress will later lead to SPIDERS3D updates.
10. Support future practicum efforts: AIS stream to DIS, exemplar for SIMC2 NATO work
11. “Your project here”... work supporting your thesis or professional interests is especially welcome.

Grading Criteria

Percent	Task	Summary
20%	Assignment 1	Unicast sockets sender/receiver, modifying provided code
20%	Assignment 2	Multicast sockets sender/receiver, modifying provided code
20%	Assignment 3	OpenDIS PDU Track Sender, modifying provided code, then recorded/saved/replayed using X3D-Edit or Wireshark
20%	Project: Source Code and Data Content	Deliverables for your project's functionality, listed on page 4
20%	Project Documentation and Class Presentation	<i>Project Attributes</i> are also listed on page 4
100%	gitlab version control	This work helps you to perform future duties as a MOVES graduate, your efforts in the follow-on practicum course, and future students.