6 March 2018

MV 3500 Internetwork Communications and Simulation (3-2)

January-March 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). The emphasis is on network programming network code and web-browser applications.
Prerequisites: CS2971 and CS2173.

Instructor

Don Brutzman	brutzman@nps.edu	1.831.656.2149 work
Office Watkins 270	http://faculty.nps.edu/brutzman	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 demos dates are announced in advance. Savage Lab in Watkins 267 is typically 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, http://netbeans.org
Wireshark, https://wireshark.org
Open-DIS Project, https://github.com/open-dis
X3D-Edit Authoring Tool, available at https://savage.nps.edu/X3D-Edit

Textbooks

1. Networked Graphics: Building Networked Games and Virtual Environments by Anthony Steed and Manuel Fradinho Oliveira, Elsevier, 536 pages, 2009. Chapter 7 discusses X3D, X3D-Edit and DIS.
2. *Networked Virtual Environments*, Sandeep Singhal and Michael Zyda, ACM Press SIGGRAPH Series, Addison Wesley, 2009.
3. Don Brutzman and Len Daly, *X3D for Web Authors*, Morgan Kaufmann, 2007, http://x3dgraphics.com

Guidelines

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 knowledgably 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. I prefer that you hand in something late which is correct and high quality, rather than something on time which is broken.

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	8-12 JAN	Introduction and getting started, TCP/IP networking	Install Netbeans, discuss programming languages
2	15-19 JAN	C2SIM NATO Project, Dr. J. Mark Pullen GMU SPIDERS3D Introduction, Alex Viana NAVFAC Open-DIS Overview, Don McGregor NPS	Instructor travel; guest lectures
3	22-26 JAN	TCP/IP continued	Install Wireshark
4	29 JAN - 2 FEB	TCP/IP Sockets in Java, Java Input/Output (I/O)	Install X3D-Edit
5	5-9 FEB	User Datagram Protocol (UDP), Network Scalability	
6	12-16 FEB	Distributed Interactive Simulation (DIS) Protocol	Midterm exam, date TBD
7	19-23 FEB	DIS applications	Final project discussions
8	26 FEB - 2 MAR	High-Level Architecture (HLA)	Project design
9	5-9 MAR	Test and Training Enabling Architecture (TENA)	Project design
10	12-16 MAR	Network security highlights, PKI quicklook	Project testing
11	19-23 MAR	Finals and class project demonstrations	Possibly sooner if all agree

Candidate Course Projects

Project attributes

- Project preparation plan/progress report in class, after first week.
- Source code checked into course gitlab version control. Netbeans project or standalone.
- README description and directions, includes Javadoc for classes and methods.
- Simple powerpoint presentation including key points and screenshots.
- Demonstration to class, 10-20 minutes. Include list of accomplishments plus TODO items.
- Repeatable and useful, 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 that values are being 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, or Rota Spain

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. Progress will later lead to SPIDERS3D updates.

~~10. Support future practicum efforts: AIS stream to DIS, exemplar for SIMC2 NATO work: deferred~~

11. "Your project here"… special work supporting your thesis or professional interests is 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	*Project Attributes* 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.