

# Teaching Network Protocol Concepts in an Open-Source Simulation Environment

J. Mark Pullen

Department of Computer Science

George Mason University

4400 University Drive, Fairfax VA 22030 USA

[mpullen@cs.gmu.edu](mailto:mpullen@cs.gmu.edu)

# Overview

- Philosophy behind Network Workbench
- Teaching network protocols using simulation
- Java Network Workbench 2
  - Features
  - Structure
  - Available exercises
- JNW2 and constructivism
- Grade outcomes/conclusion

# Philosophy behind Network Workbench

- We've been using simulation as a teaching tool for network protocols since 1993
- Students learn more by solving problems
  - Specifically how network protocols work
  - We provide scaffolding for all the “busy work” of programming
  - They program the core algorithms
  - Simulation provides reproducible results
- Java Network Workbench 2 (JNW2) replaces Network Workbench (which was C++ based)
  - Open source software for teaching

# JNW2 and Constructivism

JNW2 is rooted in constructivist philosophy as laid out by Savery and Duffy (1995):

- *Anchor all learning activities to a larger task or problem*
- *Support the learner in developing ownership for the overall problem or task*
- *Design an authentic task*
- *Design the task and the learning environment to reflect the complexity of the environment they should be able to function in at the end of learning*
- *Give the learner ownership of the process used to develop the solution*
- *Design the learning environment to support and challenge the learner's thinking*
- *Encourage testing ideas against alternative views and alternative contexts*
- *Provide opportunity for and support reflection on both the content learned and the learning process*

# What is Provided in JNW2

- Overall environment packaged for NetBeans
  - Also usable in Eclipse
  - Java offers lower complexity and student familiarity
- At the heart, a Discrete Event Simulation (DES) engine
  - All activity has known time (or distribution)
  - Each event can trigger another event
- Java classes to support all functions
  - Good quality programming
  - Stack patterned after the Internet Protocol Suite
  - In the middle of key stack layers, place for student to add code
  - With algorithm to be implemented, in comments
  - Must interoperate with open-source code provided
- Example data “email” and network structure

# Java Network Workbench 2 Features

- Like the Internet, node addresses have a local and global part
  - And there is a Data Link layer underneath
- DES steps through the simulation visibly, in “ticks”
  - *nextEvent()* pulls next scheduled action from list
- Stochastic option: an event can have predictable time or last for time drawn at random from a distribution
  - But the random number generator seed always follows the same pattern so the outcome is predictable
- Network can be viewed graphically

JNW2

File

Simulation Stopped   Add Subnets   Add Link   Clear Text   Clear All

Run, Draw and Save Network

Network Topology

Network Parameters

number of routers/subnets:7  
 number of hosts:4  
 number of links:16  
 WAN diameter:3  
 reliable links:false  
 serial link bit error rate:1.0E-6  
 LAN bit error rate:0.0  
 reliable transport:true

links matrix:

0	384	384	0	0	0	0
384	0	0	1544	0	1544	0
384	0	0	0	384	0	0
0	1544	0	0	0	0	128
0	0	384	0	0	384	128
0	1544	0	0	384	0	0
0	0	0	128	128	0	0

exitInterfaces matrix:

0	1	2	0	0	0	0
3	0	0	4	0	5	0
6	0	0	0	7	0	0
0	8	0	0	0	0	9
0	0	10	0	0	11	12
0	13	0	0	14	0	0
0	0	0	15	16	0	0

subnet	lanType	hostCount	dataRate
1	1	1	
2	1	0	
3	1	0	
4	1	0	

Print at

Lan Collision    Application Layer    DLC Layer

CRC/FCS    Transport Layer    Physical Layer

Bit Stuffing    Network Layer    Slow Start

Data Files  
stream-Test1.txt

Output Parameters

Optional

Reliable Links

Reliable Transport

Lan Bit Error R... 0.0

Network Characteristics   Routing Matrix

# Java Network Workbench 2 Exercises

- Data Link Layer
  - *bit stuffing/unstuffing*
  - *error detection*
  - *Carrier Sense Multiple Access with Collision Detection*
- Network Layer
  - *topology matrices*
  - *network layer routing*
- Transport Layer
  - *reliable transport layer*
  - *slow-start sending end window*



# More Exercises Planned

- *token passing local area networks ?*
- *network layer routing information distribution*
- *multicast networking*
- *network security*
- *application layer message handling*
- *mixture of traffic types*

# Grade Outcomes and Conclusion

- ABET requirements stimulated collecting outcomes; students score a letter grade higher on JNW2-related exam questions

Semester	Number of Students	JNW2-related	Not JNW2-related
Fall 2009	26	64.6	61.3
Spring 2014	30	67.7	56.2
Fall 2014	22	78.0	72.0
Spring 2016	35	78.1	63.5
Overall	113	72.2	62.7

- Students enjoy the projects and learn more
- JNW2 is available <http://netlab.gmu.edu/JNW2>
  - Solutions available by faculty request on department letterhead