

Introduction to the Wireless Emulator

Haowen Chan

Carnegie Mellon University

Outline

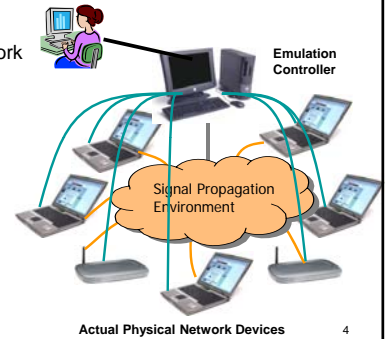
- **CMU wireless emulation testbed overview**
- **Setting up your account**
- **Programming the emulator**

Emulator Documentation

- <http://www.cs.cmu.edu/~emulator/>
- This recitation is mostly **examples**
- Comprehensive interface/function reference is in the documentation
- It is brief; so **read the whole documentation** (except the javadoc)
- Some parts are outdated due to switch to Emulab interface but most of it is still valid
- Files for this recitation are available on course website

Typical Usage

- **ssh** access to emulation controller
- **ssh** access to network devices via **wired control network**
 - Configure wireless settings, run applications, etc
- **emuRun** program controls **wireless Signal Propagation Environment**
 - GUI, XML scripting, Java interfaces to change signal environment



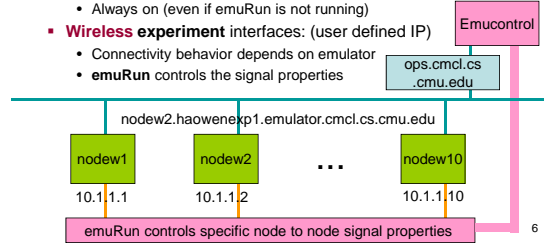
Actual Physical Network Devices

Experiment Definition

- Defined in NS script file
 - Typically a template will be given to you
- Number of nodes
- Type of network (AP-based, adhoc)
- Node particulars
 - IP addresses
 - User-defined names
- Quite self-explanatory, see examples

Emulator SSH Login

- Main emulator control machine: `emucontrol-1.ece.cmu.edu`
- Two networks connecting experimental nodes:
 - **Wired control** interface: `<name>.<EID>.<PID>.cmcl.cs.cmu.edu`
 - Always on (even if emuRun is not running)
 - **Wireless experiment** interfaces: (user defined IP)
 - Connectivity behavior depends on emulator
 - **emuRun** controls the signal properties



Node Control

- From emucontrol-1, first ssh to ops.cmcl.cs.cmu.edu, then ssh to experiment nodes
 - e.g. ssh nodew2.haowenexp1.emulator.cmcl.cs.cmu.edu
- Once logged into a node, can control its operation
 - Change wireless stack parameters, run protocols and applications
 - sudo access to some wireless config commands
 - E.g.: sudo iwconfig ath0 channel 6
 - Network diagnostics
 - E.g.: ping, iperf, etc
 - For projects: run your own application/protocol
- Execute emuRun on emucontrol-1 to start wireless experiment

7

Outline

- CMU wireless emulation testbed overview
- **Setting up your account**
- Programming the emulator

8

Slide Pace Increase Alert!

- Due to time constraints, these slides will be skimmed over
- Just simple, direct instructions
 - Type this, copy that file
- This presentation will be available online and a more detailed list of instructions will be posted on the course webpage.

9

Getting Started

- CMULab accounts:
- <https://boss.cmcl.cs.cmu.edu/>
- →Request Account
- **Important:** use ANDREW ID for userID
- Project name: emulator
- Check your email and verify it by clicking on the link
- Wait for Emulator staff to approve your membership
- emucontrol-1 accounts will be created for you by emulator staff

10

Emucontrol-1 Account

- Login to emucontrol-1.ece.cmu.edu will be your andrew ID (password will be provided by staff)
- Most of your program code will be on your home directory here
- Please read-protect your directory
 - e.g. chmod 700 /home/<youruserid>

11

SSH Access

- Use CMULab -> My Profile -> SSH Keys to upload your SSH public keys for SSH access to ops.cmcl.cs.cmu.edu
 - E.g. to allow ssh from emucontrol-1 to ops, upload the file ~/.ssh/id_rsa.pub to the website
 - To enable stuff like winSCP from your personal computer, generate a keypair (e.g. with puttygen or ssh-keygen) and upload the public key using the same interface

12

SSL Setup

- My emulab --> profile --> Generate SSL cert
 - Remember your password
- SSH to emucontrol-1, then SSH to ops.cmcl.cs.cmu.edu
- Decrypt the pem
 - cd .ssl
 - openssl rsa -in encrypted.pem -out emulab.pem
 - openssl x509 -in encrypted.pem >>emulab.pem
 - You will be prompted for the password
 - Password can be forgotten after this step!
- Now `exit` to emucontrol-1, copy the decrypted pem file
 - `scp ops.cmcl.cs.cmu.edu:~/ssl/emulab.pem ~/.ssl`

13

Upload Required Files

- From <http://www.cs.cmu.edu/~haowen/15496/>
 - Get either `emustartpack.rar` or `emustartpack.taz`
 - Unpack the archive in a convenient directory
 - e.g. `cd temp`
 - `tar xvzf emustartpack.taz`
 - follow the README:
- copy `file.bashrc` to `~/bashrc` on emucontrol-1.ece.cmu.edu
- copy `file.cshrc` to `~/cshrc` on ops.cmcl.cs.cmu.edu
- copy the whole folder `emuNode/` to `~/` on ops.cmcl.cs.cmu.edu
- Rest of the files are useful examples, etc for reference

14

Outline

- **CMU wireless emulation testbed overview**
- **Setting up your account**
- **Programming the emulator**

15

Setting up an Experiment Session

- CMULab webpage -> Experimentation -> Begin an Experiment
- Project: emulator
- Group: default
- Name: include your username e.g. haowenexp01
- NS File: Described later
- Idle-swap: set to 1
- Max-duration: up to you
- Linktest: skip linktest
- Leave last two boxes unchecked

16

Experiment Session Startup

- May take up to 10 minutes
- Receive an email when it's done.
- Now ssh to emucontrol-1, type:
 - `emulatorDaemon <your user ID> emulator <experimentname>`
- This will generate a file `emulab.txt` in your home directory
- `emuRun` will parse this file to identify node information etc

17

Running the Experiment Session

- ssh to your nodes through ops.cmcl.cs.cmu.edu
 - emucontrol-1 cannot ssh to the nodes
 - Node wired interface address:
 - `<user-defined node ID>.<experiment ID>.emulator.cmcl.cs.cmu.edu`
 - e.g.
 - `emucontrol-1> ssh ops.cmcl.cs.cmu.edu`
 - `ops> ssh nodew1.haowenexp01.emulator.cmcl.cs.cmu.edu`
- Start `emuRun` with the desired xml script file:
 - `emuRun clearChannels2.xml`

18

Multiple Experiments with same topology

- If you use the same experiment topology (e.g. 2 nodes, adhoc mode) for different tests, you don't have to restart the session
- Just end emuRun and start it again for the next test
- The various demos in this recitation are separate experiments but we will use the same Emulab experiment session
- Saves on startup time.
- However if you need a different topology (e.g. more nodes needed) then you must terminate the current session and start a new one

19

Ending your Experiment Session

- **VERY IMPORTANT!**
- **Always Terminate your experiment when done**
 - My EMULab-> Experiments -> (Experiment name) -> Experiment options (left sidebar) -> Terminate experiment
- **If you do not terminate then you are hogging the nodes for other users**
- **Staff will be monitoring the system for hogs:**
 - **Don't be a hog!**

20

Programming Interfaces to the Emulator

- Graphical Interface
 - Good for informal experiments
 - Deploy nodes and move them around
 - Observe mobility and effects on protocols in real time
- XML Scripts
 - Good for simple experiments
 - Quick, easy to learn, fast to implement
- Java Interface
 - Most useful/expressive interface
 - All your assignments after #1 will probably use the Java interface

21

GUI Demo

- Launch emuRun and note the URL for GUI
 - e.g.:
 - http://emucontrol-1.ece.cmu.edu:8180/doc/josh/test/emu_haowen32726.html
- Connect with your browser
 - IE and firefox are ok (for me)
- Site to check if your java is current:
 - <http://www.java.com/en/download/help/testvm.xml>
- Login with your reservation password
 - emailed to you with emucontrol-1 login details
- Play with the buttons and the nodes!

22

XML Interface

- XML interface used to control various emulator parameters
 - Positions of nodes, mobility
 - Characteristics of signal between specific nodes
 - Event-based scheduling of operations
 - Remote execution of applications on nodes
 - rregistry and nodeDaemon must be loaded on remote nodes

23

Example Script

- Show and run example script

24

Java Interface

- Can control every aspect of the emulator
- Java user class referenced and loaded from script
- Class starts execution at `start` method
- Can be event-driven or batch-driven or both

25

Steps to write user code

- Write java program which extends `SimpleUserclass`
- Compile the java program into bytecode program (`javac`)
- Specify the user class in the experiment script
- Run the experiment script

26

Implementing Your Class

```
Typical Usage:
public abstract class MyExperimentClass
extends SimpleUserClass
implements ProcListener
```

- Important Member Objects
 - `dComm`: communication with `nodeDaemon`
 - `emulator`: main class in charge of running the emulator
 - `sigEnv`: signal propagation environment
 - `world`: virtual world
- Method
 - `start ()`: called by the emulator at the start of the emulation
 - `processExited(ProcessID procID, int code)`: called when a `nodeDaemon` remote process exits (see example)

27

Javadoc

- <http://www.cs.cmu.edu/~emulator/doc/api/index.html>
- Basically the class descriptions for the entire emulator (!)
- Use it as an API reference only
 - don't read the whole thing!
- Ask me (haowen+15496ta@cs.cmu.edu) if you are looking for a specific functionality
- In assignments I will generally tell you which classes and methods are needed
 - Or they will be contained in the examples shown in recitation
 - Understand these examples thoroughly (if any line doesn't make sense, ask me)

28

Example Java code

- Show example

29

What's Next?

- Assignment 1 will be released tonight or tomorrow
 - Due in about 1 week (details TBD)
- You will be emailed with your `emucontrol-1` login
- Set up your account and try to run these examples
- All these materials are available at:
 - <http://www.cs.cmu.edu/~haowen/15496/recitation01>
- Extended Office Hours on Friday
 - Starts at 1pm
 - Ends when I feel everyone attending can at least get started on running simple experiments on the emulator
- I will be away 30 March – 1 April

30