7. The Open Network Lab

- Overview and getting started
- Building a network topology
- Configuring routes and filters
- Monitoring traffic

*Jon Turner*
The Open Network Lab

- Internet-accessible networking lab (onl.wustl.edu)
  - built around set of extensible gigabit routers
  - intuitive Remote Lab Interface makes it easy to get started
  - extensive facilities for performance monitoring

- Variety of resources
  - 4 eight port routers, called Network Services Platform (NSP)
    - highly configurable
    - embedded processor at each port with plugin environment
  - 14 five port Network Processor based Routers (NPR)
    - also highly configurable
    - each has five processor cores reserved for plugins
  - 6 four port NetFPGA cards
    - hardware can be reconfigured to implement different devices
  - over 100 rack-mount computers that serve as end systems
    - including multicore servers with 8 cores and 48 cores
Overview of ONL

- Remote access through the Internet using a graphical user interface (called the RLI)
- Provides access to variety of hardware resources
- Experimental networks built with configuration switches
Getting Started

- Request an account at onl.wustl.edu
  - read Getting Started page and look at tutorial pages
- Download the RLI (version 7.5 is the latest)
  - to run RLI you must have Java Runtime Environment installed (version 1.6 or higher)
- Create experimental network and save to a file
  - details on next slide
- Open an SSH connection to ONL with “tunnel” for RLI
  - on command line: ssh -L 7070:onlsrv:7070 user@onl.wustl.edu
- Make a reservation using RLI and wait for confirmation
  - note the start time in the confirmation message
- After start time, select File ➔ Commit in RLI
  - open ssh connections to your ONL hosts and run applications
Building a Basic Configuration

- Select
  - Topology ➔ Add PC1core ➔ Add Host 1core
  - and specify 3 hosts
- Select
  - Topology ➔ Add GigE Switch
  - and specify single 1 Gb switch
- Select
  - Press “Links Off” button
  - and wire hosts to switch
- Select
  - File ➔ Save As ...
  - File ➔ Make Reservation
  - File ➔ Commit (do not forget!!)
Running Applications

- Open ssh connections to hosts
  - first ssh to onl.wustl.edu
  - then, to pc1coreNN ...
  - note, each host has two names
- Run standard apps
  - on h1x1, type “ping h1x3”
  - on h1x2, type “iperf -s”, then on h1x1, type “iperf -c h1x2”
  - on h1x2, type “sudo /usr/sbin/tcpdump -i data0” then repeat ping from h1x1
- Run basic client/server programs
  - on h1x2, run java UdpEchoServer
  - on h1x1, run java UdpEchoClient h1x2 9876 "hello world"
Working with Larger Configurations

To include pair of NPRs (routers), select Topology ➔ Add IXPCluster ➔ Add NPRCluster

Note subnet numbering

Note multiple paths between routers
Working With Routing Tables

1. Change routes using Operations→Edit
2. Port 4
   - Generate default routes to configure routing tables
   - Click on port to access specific table
   - Next hop specifies output port and next hop's IP address
Configuring Packet Filters

- Select Filter Table
- Address prefixes and ports
- Protocol and flags (TCP)
- Packet treatment
- Turn on/off
Configuring Packet Filters

- Address fields
- Protocol and ports
- Queue number
- Output link
Monitoring Traffic

To add new real-time chart

observing ping traffic from h4x2 to h6x1

to select outgoing packet rate at a port
Hosts with Multiple Interfaces

- ONL hosts have two interfaces
  - data interface used for experimental network
    - names of form hAxB. and addresses of form 192.168.AxB addresses,
  - control net interface used to talk to ONL servers
    - names of form pc1coreNN, addresses of form 10.0.x.y
    - file /users/oni/.topology defines shell variables of form $hAxB that map to control net names

- To force server connections to use a specific interface, bind socket to desired interface’s IP address
  - forces communication to server to come through specified iface
- If socket left "unbound", system uses wildcard interface
Using the Linux Command Line

- ONL hosts all run Linux
  - to use them effectively, need to be familiar with the command-line interface provided by the “shell” (specifically, bash)

- Common commands
  - ls to list files in a directory (aka folder)
  - cd to change the current working directory
  - more to examine contents of a file
  - ssh to open an ssh connection to another computer
  - man to get documentation for a given command
    - type “man ls” to learn how to use the ls command
  - See http://www.ee.surrey.ac.uk/Teaching/Unix/index.html

- You need to invest some time up-front to learn Linux
  - but, you’ll be much more productive once you do
Using Wireshark in ONL

- Wireshark is implemented using the X-windows system
  - ssh supports tunneling of X-windows commands, allowing you to run Wireshark in ONL with GUI running on laptop
- On Urbauer lab computers
  - first type "startxwin" in a command prompt window
  - this starts new window; in this window, type
    ssh -X userName@onl.wustl.edu
    source /users/onl/.topology
    ssh -X onlHostname (e.g. $h4x2)
    sudo wireshark
  - this starts Wireshark on remote machine, but your local computer acts as the display for remote instance of Wireshark
- Can use this with other X-windows applications also