Washington University in St. Louis

Engineering

7. The Open Network Lab

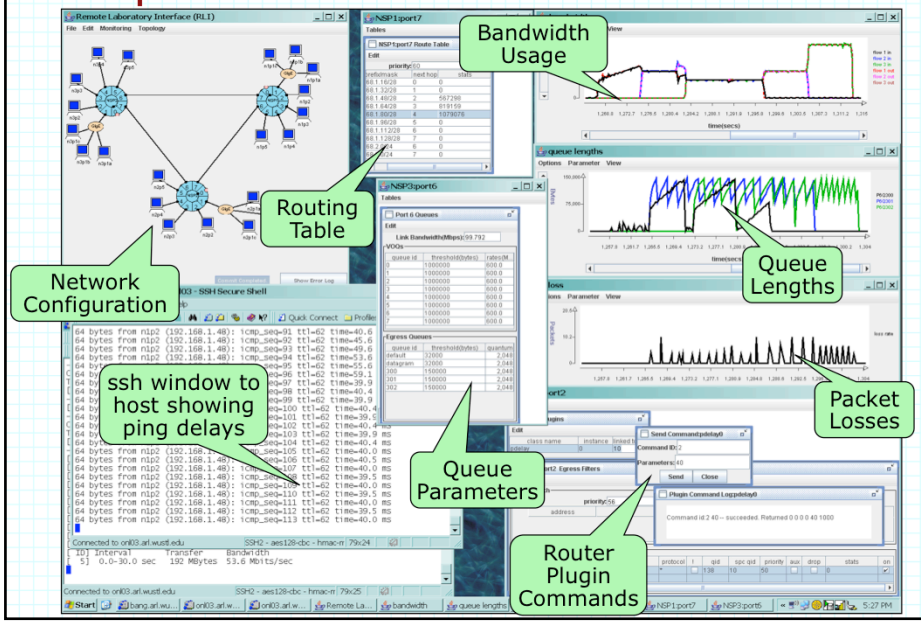
- Overview and getting started
- Building a network topology
- Configuring routes and filters
- Monitoring traffic

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

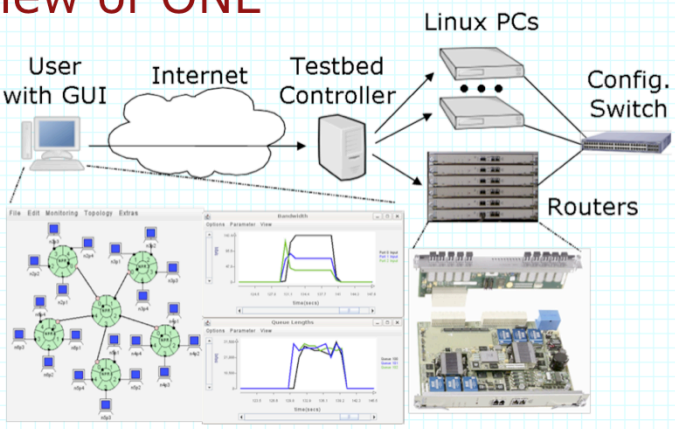
Sample ONL Session



Washington University in St. Louis

Engineering

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

4

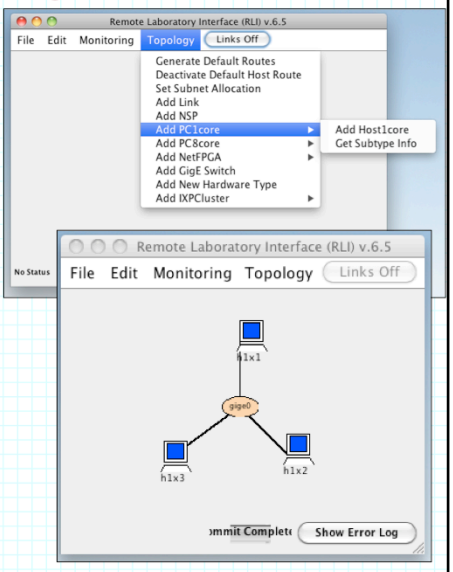
Getting Started

- 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

Washington University in St. Louis | Engineering

Building a Basic Configuration

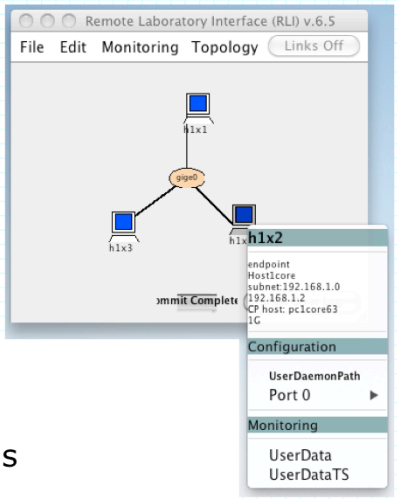
- Select
  - » Topology➔Add PC1core ➔AddHost1core
  - » and specify 3 hosts
- Select
  - » Topology➔Add GigE Switch
  - » and specify single 1 Gb switch
- Select
  - » Press "Links Off" button
  - » and wire hosts to switch
- Select
  - » File➔Save As ...
  - » File➔Make Reservation
  - » File➔Commit (do not forget!!)



6

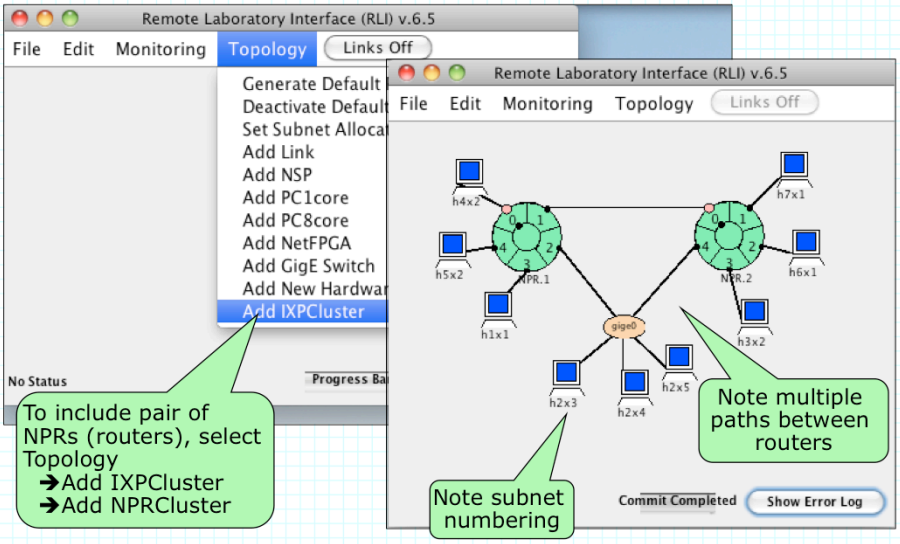
Running Applications

- 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"



Washington University in St.Louis | Engineering

Working with Larger Configurations

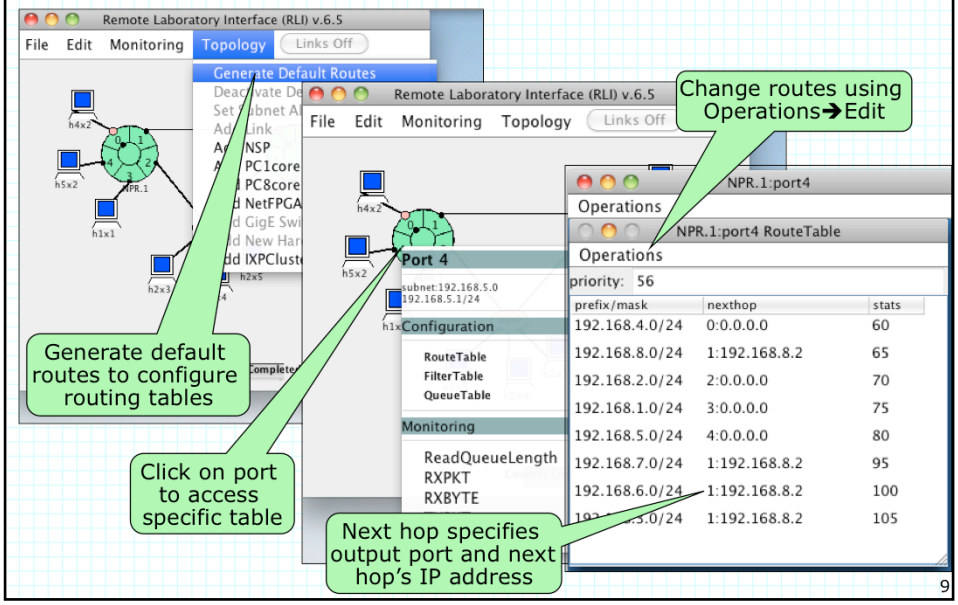


8

Washington University in St.Louis

Engineering

Working With Routing Tables

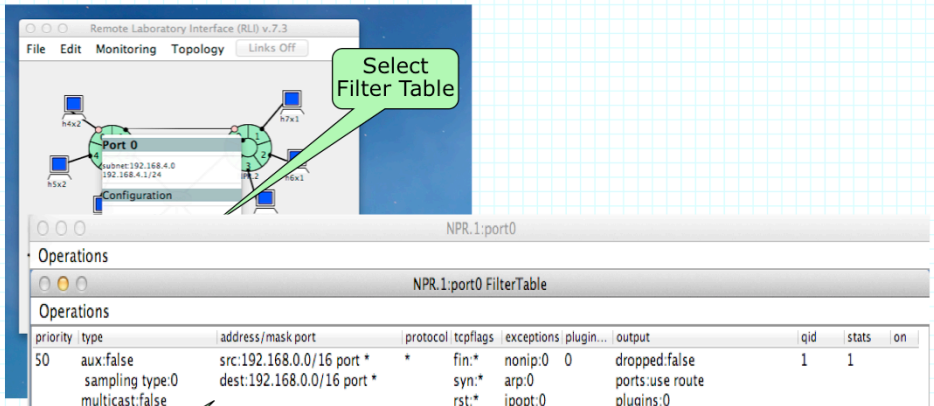


9

Washington University in St.Louis

Engineering

Configuring Packet Filters



NPR.1:port0

Operations

NPR.1:port0 FilterTable

Operations

priority	type	address/mask port	protocol	tcpflags	exceptions	plugin...	output	qid	stats	on
50	aux:false sampling type:0 multicast:false	src:192.168.0.0/16 port * dest:192.168.0.0/16 port *	*	fin:* syn:* rst:* psh:* ack:* urg:*	nonip:0 arp:0 ipopt:0 ttl:0	0	dropped:false ports:use route plugins:0 pluginSelection:plugin(unicast)	1	1	☑

address prefixes and ports

protocol and flags (TCP)

packet treatment

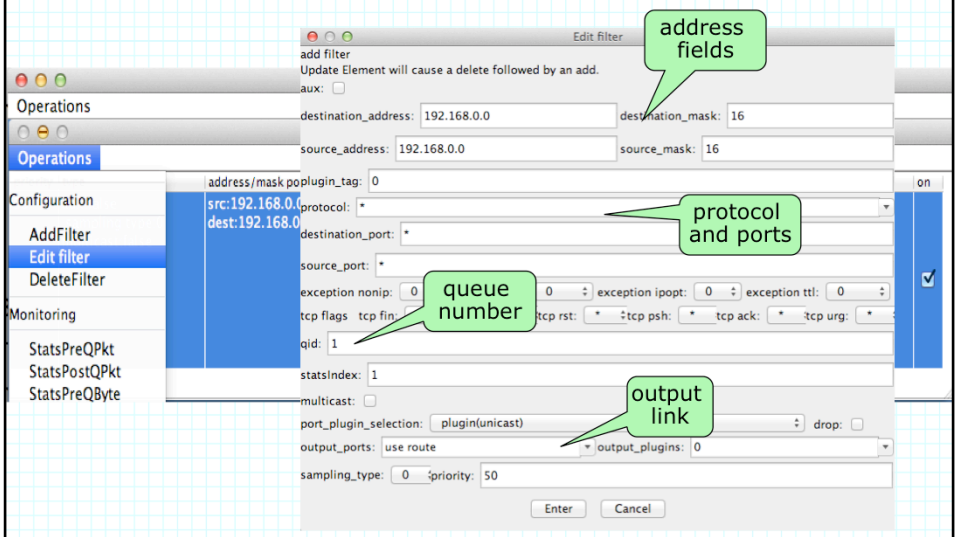
turn on/off

10

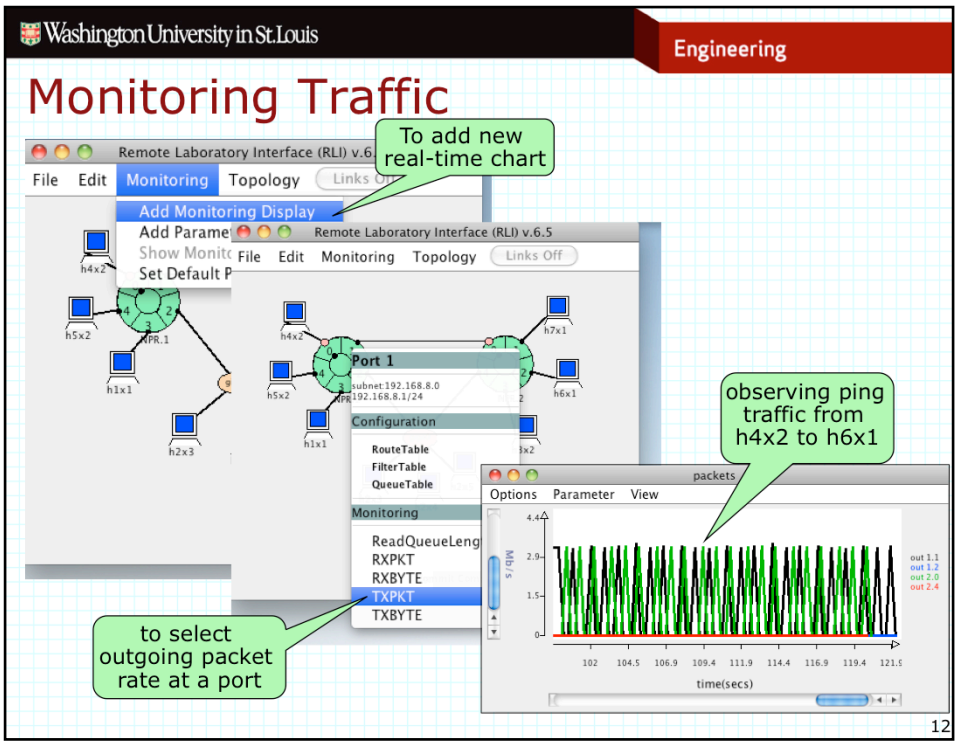
Washington University in St.Louis

Engineering

Configuring Packet Filters

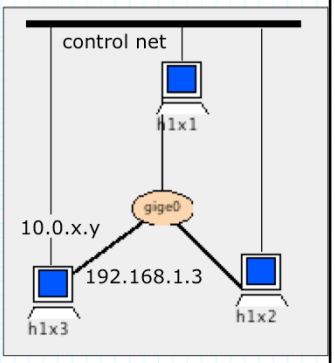


11


Washington University in St.Louis
Engineering
Monitoring Traffic
To add new real-time chart
Remote Laboratory Interface (RLI) v.6.5
File Edit Monitoring Topology Links Off
Add Monitoring Display
Add Parame
Show Monitc
Set Default P
Port 1
subnet:192.168.8.0
192.168.8.1/24
Configuration
RouteTable
FilterTable
QueueTable
Monitoring
ReadQueueLeng
RXPKT
RXBYTE
TXPKT
TXBYTE
observing ping traffic from h4x2 to h6x1
packets
Options Parameter View
Mbit/s
time(secs)
to select outgoing packet rate at a port
12

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/onl/.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* and *http://www.gnu.org/software/bash/manual/bash.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