Using the Open Network Lab

Jon Turner
Computer Science & Engineering
Washington University

www.arl.wustl.edu/~jst
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
  - 1 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
  - about 100 rack-mount computers to serve as end systems
ONL Testbed Configuration

control subnet

Internet

online server

24 Gb/s ring

configuration switches
Equipment Photos
NSP Organization

- Scalable architecture built around ATM switch core
  - Core provides 2 Gb/s bandwidth per port (2x speedup)
- Port processors (PP) implement packet processing
  - Field Programmable Port Extender (FPX) implements routine packet processing
  - Smart Port Card (SPC) hosts programmable extensions
- Control Processor (Linux PC) handles configuration
  - Can support routing protocols, OA&M, etc.
NPR Overview

- Network Processor-based Router
  » Specifically, based on Intel's IXP 2800
- 5 1-Gigabit Ethernet Ports
- Supports real-time configuration and traffic monitoring

- A few IXP details
  » 16 MicroEngines (MEs)
  » xScale for control (MP)
  » 4 SRAM channels
  » 3 DRAM channels
  » TCAM support

[Diagram of network processor architecture]
NPR SW Modules and HW Mapping

Diagram showing the flow of data through various modules and hardware components, including Rx (2 MEs), Mux (1 ME), Parse, Lookup, Copy (3 MEs), XScale, TCAM, FreeBSD Manager (1 ME), Stats (1 ME), Queue Manager (1 ME), Header Format (1 ME), and Tx (1 ME). The diagram also includes Plugin 0 (1 ME), Plugin 1 (1 ME), Plugin 2 (1 ME), Plugin 3 (1 ME), and Plugin 4 (1 ME).
RLI menus

- Transfer specified configuration changes to network components
- Reserve configured network for a specified time period
- Create and configure real-time monitoring displays
- Add components to network configuration
- Configure routing tables in IP routers
Configuring Topologies

Before commit
Host names (and IP addresses) assigned based on router ports

After commit
Accessing Routing Tables

- Third address digit specifies router
- Fourth address digit is $16^{(port\#+1)}$
- Range of 16 addresses per port
Configuring Queues

Select to add entries to queue table

Number of bytes queue can send on each "turn"

Threshold determines queue size
Sending Traffic to Queues with Filters

- Re-route all packets to subnet 3
- Use queue 100 on port 2
- Priority 50 so filter overrides route
Generating Traffic with Iperf

Sample uses

- `iperf -s -u`
  - run as UDP server on port 5001
- `iperf -c server -u -b 20m -t 300`
  - run as client sending UDP packets to server at 20 Mb/s for 300 secs.
- `iperf -s -w 4m`
  - run as TCP server on port 5001
  - set max window to 4 MB
- `iperf -c server -w 4m -t 300`
  - run as client, sending as fast as possible, with max window 4 MB

available at http://dast.nlanr.net/projects/Iperf/
installed on all onl hosts.
Adding More Data

forward and return traffic

click on labels to change text
Using Iperf

single UDP stream

start UDP sender
Getting Started

- Go to onl.wustl.edu and request account
- Read and follow getting started instructions
  - download Remote Lab Interface (RLI)
  - if you don’t already have them
    - install Java runtime environment
    - install SSH client
  - configure SSH tunnel (7070) for connecting RLI to main ONL server
- Configure simple topology and reserve session
  - explore menus, start trying things
- Read online tutorial (focus on NPR sections)
  - try things out as you read