

CSE 473S – Introduction to Computer Networks Fall 2011

Lab Assignment 3

Objective: Experiment with UDP and TCP packets

Preliminaries

- You should have completed lab assignments 1 and 2
- Download configuration file *lab3udp.exp* and *lab3tcp.exp* from the course web page <http://www.cse.wustl.edu/~jain/cse473-11/> (right-click on lab3udp.exp and save link as or save target as. Repeat for lab3tcp.exp)
- Save the files in *.onldir* directory

Both configuration files sets up a network topology enabling a sender at host n1p2 to communicate with a receiver at host n2p2 as shown in Figure 1. The two configuration files differ mainly in the amount of buffers in one of the router interfaces.

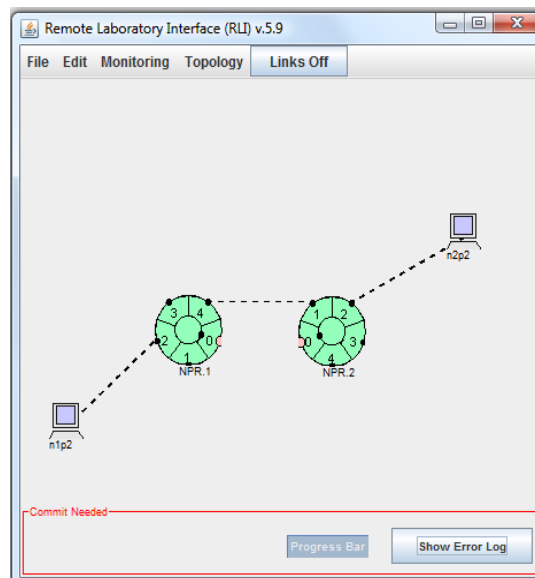


Figure 1: Network topology for lab assignment 3

Start the RLI and open the configuration file.

Four windows are supposed to appear and contain:

- Configured network topology
- Display monitoring the length of queue 64 at port 4 of NPR1
- Display monitoring ACK packets from receiver host to the sender host
- Display monitoring the traffic bandwidth from the sender to the receiver:
 - RXBYTE1.2 – traffic arriving at port 2 of NPR1
 - RXBYTE2.1 – traffic arriving at port 1 of NPR2
 - TXBYTE2.2 – traffic leaving port 2 of NPR2

The experiment will use **iperf** commands to generate UDP and TCP traffic. The commands will be clearly explained whenever they are used. Though, a summary of the iperf commands can be found at the following link:

<https://onl.wustl.edu/> → OLD-NPR Tutorial → Filters, Queues and Bandwidth → Generating Traffic with Iperf

❖ Pre-run Analysis

1. Report the threshold (bytes) of queue 64 at port 4 of NPR1
2. Report the bottleneck capacity for the packet path from host n1p2 to host n2p2

❖ Run-time ONL Experiment

➤ UDP traffic flow from host n1p2 to host n2p2

Build an SSH tunnel to the testbed, start the RLI, open the configuration file lab3udp.exp, and make reservation for the hardware resources. You can make the reservation (minutes, hours, days, or even weeks) in advance. When your reservation time comes, reestablish the SSH tunnel and commit your topology.

Log into n2p2 through its control network interface name (e.g., onlusr> ssh pc1core49) and execute the command **iperf -s -u** to start the host as a UDP receiver.

Log into n1p2 through its control network interface name and start it as a UDP sender: execute the command **iperf -c n2p2 -u -b 10m -t 20**. The command sets up the sender to transmit at rate of 10 Mbps for 20 seconds. The packets will be received by host n2p2.

3. Include screenshots of the monitoring displays of bandwidth, queue length, and ACK from receiver charts.
4. Answer the following:
 - a. What are the bottleneck capacity and average throughput in Mbps?
 - b. Is there any significant (more than 10 packets, assume each packet is 1500 bytes) queue along the path?
 - c. What leads to the impulse in the returning traffic (“ack”) display?

Log into n1p2 through its control network interface name and start it as a UDP sender: execute the command **iperf -c n2p2 -u -b 15m -t 20**. The command sets up the sender to transmit at rate of 15 Mbps for 20 seconds. The packets will be received by host n2p2.

5. Include screenshot(s) of the monitoring displays of bandwidth, queue length, and ACK from receiver charts.
6. What is the rate of queue growth in the beginning and queue drain later in Mb/s?

➤ TCP traffic flow from host n1p2 to host n2p2

This section will use lab3tcp.exp which can be found on the website. Note that unlike UDP, TCP periodically sends ACK packets back to the senders.

Log into n2p2 through its control network interface name (e.g., onlusr> ssh pc1core49) and execute the command **sudo /usr/local/bin/net/sack-off** then **iperf -s -w 4m** to start the host as a TCP receiver with 4 MB receiver socket buffer. Note that iperf message indicates that although a TCP window size 4Mbytes was requested, a TCP window size 8Mbytes was granted.

Log into n1p2 through its control network interface name and start it as a TCP sender: execute the command **sudo /usr/local/bin/net/sack-off** then **iperf -c n2p2 -w 4m -t 30**. The command sets up the sender to transmit TCP packets to host n2p2 for 30 seconds. It also requests for a TCP window size 4MB. Iperf message indicates that sender connects to receiver at TCP port 5001 and a TCP window size 8MB is granted.

7. Report traffic size (MBytes) and bandwidth (Mbits/sec) transmitted by host n1p2 as reported by the iperf message.
8. Report the number of packets dropped by NPRs. This number is stored by the queue manager in NPR register 31 which can be accessed through the ReadRegisterPacket menu item at the center of the NPR icon.
9. What are the buffer sizes of various router queues along the path in bytes and packets?
10. Include screenshot(s) of the monitoring displays of bandwidth, queue length, and ACK from receiver charts.
11. Answer the following based on the above 3 displays:
 - a. What is the approximate average throughput after the initial surge?
 - b. Is the average throughput close to the bottleneck capacity?
 - c. What events lead to the drops in queue length and its oscillating behavior?
 - d. Which events according to you indicate TCP’s reaction to timeouts and which to triple duplicate acks?