ECE 671 – Computer Networks
Fall 2011 – Lab 2

Prof. Wolf, prepared by Jin Zhu

Please answer the following questions and submit your lab report electronically on SPARK. Grading will be based on the rubric posted on the course website.

Lab Goals

- Understand the principles of the congestion control mechanism in TCP
- Evaluate the performance of two TCP flows when they shared the same queue
- Evaluate the performance of TCP and UDP flows when they compete for the same link

ONL Setup

- The detail of the ONL setup and configuration is in Lab 1 “ONL Setup” part.
- Download the new base configuration file from the course web site into your personal direction and open it in the RLI.
- In this assignment, we use two NPRs and four PCs. This configuration is a simple dumbbell topology with 50ms one-way delay. The Monitor window is the bandwidth chart and queue length chart separately. In the bandwidth chart, we monitor the bandwidth in port 2.4 (incoming link), 2.1 and 2.2 (outgoing link). In the queue length chart, we monitor congestion at the bottleneck port by monitoring the length of queue 64 at port 1.1.

Part 0: Preparation

If you are not familiar with the congestion control mechanism in TCP, read about how it works. You do not need to understand the fine details of this mechanism – just the basic concepts and operation.

Part 1: Single TCP Connection

In this part, you need to use the iperf command to establish the TCP connection between the client (**h1x1**) and server (**h4x1**). The default delay of connection is set to 50ms, you need to set the window size to **64KB** and change the link rate between NPR.1 and NPR.2 to **10Mbps** and **100Mbps** separately to observe the changing of bandwidth and queue length. The method of change the link rate is the same as Lab 1. Make sure to let iperf run long enough (tens of seconds) to get a stable throughput result.

To run an iperf measurement, you need to start the iperf program in server mode:

iperf -s -w 64kB

To send traffic, start the client on the transmitting end-system:

iperf -c h4x1 -w 64kB -t 30 –i 5

In your report:

- Show the bandwidth and queue length variety in different link rate
- Observe whether there is "Saw tooth" showed in queue length when the link rate is increasing, explain the reason of variety of bandwidth and queue length in different link rate

Part 2: Two TCP connections

In this part, you need to use the iperf command to transmit TCP from h1x1 to h4x1 and h2x1 to h3x1. Set the window size to **64kB** and change the link rate between NPR.1 and NPR.2 to **1 Mbps**, **10Mbps** and **100Mbps** separately. Before establishing the connection, you could use the ping command to check the delay time in two TCP connections.

In your report:

- Explain the bandwidth variety of two TCP connections when they share a single link in different link rate
- Observe whether there is the bandwidth oscillation of two TCP connections in different link rate; Show the measured oscillation of bandwidth; Determine if the oscillations synchronize over time or not, discuss the result
- Show the queue length chart in different link rate and explain the difference of queue length when link rate is increasing.

Part 3: UDP and TCP Competition

In this part, you need to establish TCP connection from h1x1 to h4x1 and UDP connection from h2x1 to h3x1. The default delay is 50ms. In the TCP connection, the window size should be set to **64kB**, change the link rate between two routers to **10Mbps** and **100Mbps.** In the UDP connection, you need to set the bandwidth to **90 Mbps** to observe the bandwidth and queue length changing.

Establish a UDP connection, you need to enter the iperf command in the server part (h3x1):

iperf -s -u

In the client part (h2x1) , start the client on the transmitting end-system:

iperf -c h3x1 –u –b 90m -t 30 –i 5

In this command, the “-b 90m” means set the bandwidth of UDP to 90Mbps.

In your report:

- Show the result of bandwidth and queue length graph in different link rate and different bandwidth of UDP connection
- Explain how UDP and TCP compete for bandwidth on the shared link.