

### Purpose

- Study the performance (throughput) of TCP connections
- Change host TCP configurations
- Examine router statistics via SNMP
- Practice host and router configuration commands

### Before the Lab

- Read the description of `iperf` provided on the web site.
- Review the host and router configuration commands you used in the previous labs and come prepared to get these basic steps done quickly.
- Complete the pre-lab, and get it signed when you come to lab.

### Steps to Complete

1. Before you get started, copy the information for your table from the pre-lab to the network diagram. We have connected Buell and Harley with a T1 line and pointed Harley's default route to Buell via the T1.
2. All tables will share switch #1 to connect to Buell, but every table will connect to its own router port; patch your Mac OS machine into this switch, and connect from the switch to your port on Buell. Configure Mac OS and Buell and test your setup.
3. Connect your Linux machine to your assigned port on Harley using a hub, and configure Linux and Harley. On Buell, route traffic to your table via the serial interface (using the interface name "Serial 1/0" as the next hop entry in the ip route command). Make sure Linux can reach the Internet.
4. All tables will use switch #2 to connect the Win2008 machine to the simulated DSL environment. Configure your Win2008 machine, disable the admin interface in the Windows machine and test connectivity (routing on Buell has been set up for you).
5. *We are now ready to take some performance measurements. In this section we will look at traffic over a "bottleneck link" without much delay. Your Linux machine is connected to the Internet by T1 line (1.5Mbps). The only significant delays come from the queue that sits in the router between your Ethernet and the serial line. You are sharing the T1 with all other tables, so expect some variation in your results. You will run iperf with a number of different options. In most cases, options need to be set on **both the iperf server and the iperf client**. Stop your iperf server and restart it with the correct options before each new test.*

Use ping to record the round-trip times between the Linux machine and

- (a) Mac OS
  - (b) Win 2008
  - (c) www.yahoo.com
6. Use `sysctl -a | grep tcp | more` on Mac OS to display all operating system settings related to TCP. Record the values ending in **sendspace** and **recvspace**.

7. Use `sysctl -a | grep tcp | more` on Linux to display all operating system settings related to TCP. Record the values ending in `tcp_rmem` and `tcp_wmem`. Note the differences in the settings available on the two systems.
8. We will now examine how the different settings translate to the TCP window sizes that are available to an application. Like any other networking application, iperf can request a specific TCP window from the operating system (in most programs you cannot control that request, in iperf the “-w” option lets you set the desired window size). The larger the window size, the higher the maximum data rate at which the application can receive. Every operating system has an upper limit of the TCP window size it will permit.
  - Start on Mac OS. Start an iperf receiver (server) with `iperf -s -w 50K`. You are asking for a 50,000 kilo-byte window size, and the Mac should oblige. (*Note: no need at this point to run an actual test, just stop the server with “^C”*).
  - Repeat starting the iperf server, each time doubling the window size you are requesting. Eventually, when you ask for a window that is too large, iperf will issue a warning and tell you that it was using a *much smaller* window size. (Occasionally iperf may round up or down by a small amount, keep going in that case). At this point lower the window size you are requesting, but don’t spend a lot of time getting a precise answer; in your report you will state the maximum window size as “between ... and ...”.
  - Repeat for Linux and Windows.
  - On all three systems, start an iperf server without setting the window size (leave the “-w” off). Record the default window size each operating system assigns in this case.
9. Our first set of tests will use UDP. We will send from Mac OS (iperf client) to Linux (iperf server), simulating, e.g. a media stream from the Internet to a corporate desktop. Both the client and the server need to use the iperf “-u” option. The client also needs the “-b” option to indicate at what rate it will send UDP packets.
  - Start a transfer at 400kbps (-b 400K). Add options to run for about 30 seconds, and generate intermediate reports at 5 second intervals. Note that the client always reports the fixed bandwidth requested, the server reports the actual rate received, as well as jitter (the variation in packet arrival times) and the percentage of lost packets.
  - Repeat the test several times, each time doubling the bandwidth you ask for, ending at 3.2Mbps. Note that the client and server data rates will start to diverge at some point, and the loss percentage will be non-zero. What guesses can you make about the actual bandwidth available on the link, from these tests (keep in mind that you are sharing the link)?
10. In the next set of tests, we use iperf to send from Linux to Mac OS, simulating the uploading of data from a corporate site to an Internet server. Each test has a different set of objectives, be careful not to skip anything.
  - Start a wireshark capture (on either machine), then run a short (20sec) transfer with a window of 40,000 bytes. In the wireshark capture find the receive window advertisement from Mac OS. Record the window size for your report. (no need to save the entire packet capture). **If you plan to work the extra-credit/graduate question, use the packet capture to determine the typical size of a data-carrying packet.**

- Repeat the previous step with a window setting of 200,000 bytes. Again look for the window advertisement. Record the hexadecimal value for the window as well as the one displayed by wireshark. Locate the SYN and SYN/ACK packets and record the TCP options. *This one may be a bit tricky to interpret, ask a lab instructor.*
  - Start a continuous ping from Mac OS to Linux. Start packet captures on *both* machines. Now run a 30 second transfer with intermediate reports in 3 second intervals, using a 20K window size. You should see some increase in the ping times while the transfer is running. Stop and save the wireshark captures once the ping times have returned to “normal”. Note that each ping output shows an ICMP sequence number. When you analyze the data for your lab report, you will be able to match the ping output and ping packets in the wireshark capture by this sequence number. You should also see that the iperf client initially over-reports the throughput; this happens because it sent packets that were either queued or discarded.
  - Repeat the previous step with a large window, e.g. close to the largest allowed value you found earlier. Look for more pronounced changes in ping times. The client will probably over-report the initial data rate by a larger margin as well. Once the test is complete, examine the wireshark capture on Mac OS; look for indications of “lost segments”, “retransmission”, and “duplicate acknowledgements”.
11. *We will now look at the connection between your Windows machine and the network. This connection simulates DSL, it is asymmetric and exhibits the delay often associated with these types of services today.*
  12. In these tests, we are interested in the performance of both uploads and downloads. Use default TCP window sizes, and run your tests for about 30sec. For all tests, start a continuous ping between Mac OS and Windows before you start the iperf test.
    - Run iperf from Mac OS (client) to Windows (server).
    - Run iperf from Windows (client) to Mac OS (server).
  13. *We will now take a quick look at using SNMP to retrieve some statistics from the routers.* Run wireshark on Mac OS to capture the SNMP traffic. Use
    - `snmpget -c public -v 2c 132.235.201.40 system.sysName.0`
    - `snmpget -c public -v 2c -On 132.235.201.40 system.sysName.0`
 to get the system name from Buell. What is the difference in output between these two versions of the command?
  14. Run wireshark on Mac OS to capture the SNMP traffic. Use
    - `snmpwalk -c public -v 2c 132.235.201.40 interfaces`
 and make sure you pipe the output to a text file. Use “show interface” at the router command line to get the Cisco IOS version of the statistics for your Ethernet interface on Buell. Compare the data from SNMP and Cisco IOS; which values can you match up?
  15. While still capturing packets, use
    - `snmpbulkget -c public -v 2c 132.235.201.40 interfaces`
    - `snmpbulkwalk -c public -v 2c 132.235.201.40 interfaces`
  16. *At about the 2 hour mark into the lab, the lab instructor will give each table exclusive access to the T1 or DSL link for about 5-10 minutes at a time.*

- When you have control of the T1
  - (a) Repeat the UDP test you conducted earlier to measure the available bandwidth more accurately.
  - (b) Coordinate with the table next to you to conduct a *simultaneous* iperf test, i.e. both tables send at the same time, in the same direction. Observe how the two TCP flows share the bandwidth.
- When you have control of the DSL link, run the following tests:
  - (a) Repeat the earlier DSL iperf tests (both directions).

### Lab Report Guidelines

Each report is to be written individually, although the data for the lab can be collected during the lab with your partner/group. They should be typed/word processed and brought to class in printed form.

Lab writeups are due **in class** on the Monday following the lab. They don't generally need to be more than a few (several) pages. Officially, they need to be "long enough to answer the questions". See the web page for detailed guidelines. Each lab writeup **must** have a header on the first page that includes:

- Your name
- The lab section that you attended
- Your affiliation (CS ugrad, CS grad, ITS ugrad, MCTP grad)
- Your lab partner's name
- Your lab partner's affiliation

### Things you must include

- The patch panel worksheet
- The signed pre-lab.

### Your report must answer these questions:

1. Show the default window sizes for Mac OS, Linux, and Win2008 based on the iperf output, and report the estimated maximum window size for each system.
2. Show the window scaling option you located in the TCP SYN packet.
3. Summarize the throughput test results:
  - (a) Show the results of all Mac OS to Linux UDP tests (including the tests on the isolated T1). What can you conclude regarding the available bandwidth on the link?
  - (b) Summarize the TCP tests on the T1, and compare to the UDP results.
  - (c) Summarize all tests on the DSL (including the ones when you had exclusive access to the DSL link). What can you conclude about the uplink and downlink rates on the DSL?
  - (d) What can you conclude from the change in ping times during the DSL tests?
4. Refer to one of your Linux to Mac OS runs. Select one of the 3sec time slices and use the round trip times from ping and the reported throughput to estimate the effective TCP window size that was in use during the transfer at that time. Compare your answer to the window size you were using. You will need to use the packet capture to match the ping packets to the elapsed time in the TCP transfer.
5. Explain the measured throughput during the two-table simultaneous test on the T1.
6. Select a packet capture that shows indications of lost TCP packets and answer the questions below:

- 
- (a) Locate the first packet that carries data, then find the packet that acknowledges that data. Show how the acknowledgement number relates to the sequence number and length of the data packet it acknowledges.
  - (b) Locate a place in the packet capture where wireshark indicates that a packet is missing. Find the first acknowledgement following the gap in data flow; which data packet does it acknowledge? Locate the place in the packet capture when the missing data is re-sent, and indicate when that happened relative to the time when the loss occurred.
7. Explain the behavior of the data flow during the interfering ping flood.
  8. Compare the SNMP data for your Buell interface with the data from the `show interface` command.
  9. Show the packets that made up the request and response for one of the `snmpget` commands.
  10. Show the packet (**summary lines only**) for the `snmpwalk` command and explain how the command knew when to stop issuing requests.
  11. Compare, using the **packet summaries**, the `snmpwalk` and `snmpbulkwalk` commands. Explain why `snmpbulkwalk` may be a better choice.
  12. *Graduate Student Question:* In several tests (when you were running ping while sending data with iperf), you observed an increase in round-trip times due to queuing when the link becomes saturated. Select one test and use the data you have to estimate the average size of the queue in bytes and packets during that run.

## Pre-Lab

Fill in the required information below.

	Table 1	Table 2	Table 3	Table 4
Harley	Ethernet 0/0	Ethernet 0/1	Ethernet 0/2	Ethernet 0/3
Linux Network	132.235.201.128/28	132.235.201.144/28	132.235.201.160/28	132.235.201.176/28
Linux Netmask				
Linux IP				
Linux Router IP				
Buell	Ethernet 0/0	Ethernet 0/1	Ethernet 0/2	Ethernet 0/3
Mac OS Network	132.235.201.192/30	132.235.201.196/30	132.235.201.200/30	132.235.201.204/30
Mac OS Netmask				
Mac OS IP				
MAC OS Router IP				
Windows IP	132.235.201.225/27	132.235.201.226/27	132.235.201.227/27	132.235.201.228/27
Windows Netmask				
Windows Router IP	132.235.201.254/27	132.235.201.254/27	132.235.201.254/27	132.235.201.254/27

- Compute the maximum throughput possible with a TCP window size of 32,000 bytes and an RTT of 100msec.
  
- Write the iperf command line to start an iperf server with a window size of 20,000 bytes.
  
- Write the iperf command line to start an iperf client which will run a test for 60 seconds with a window size of 8000 bytes.

