ECE 4110 Internetwork Programming
Lab 2: TCP Traffic Generator

Group Number: _____

Member Names: _____ _____

_____ _____

Date Issued: Tuesday Jan.24, 2006
Date Due: Tuesday Jan. 31, 2005
Last Edited: 1/24/2006

Lab Goals

- Finish a TCP sockets programs to transmit and receive TCP datagrams.
- Analyze TCP network traffic using *tcpdump* and *ethereal* Linux network utilities.

Prelab

These questions pertain to the tcp_gen.c and tcp_sink.c code. They are designed to help you understand the presented code as well as the differences between our code and Steven's example code from the lecture slides. The code for these files should be attached to your handout, or you can get it from the NAS as described in Section I.

P1. In both our and Stevens' tcp client programs, the IP address of the sink is read in from the command line when the program is started. However, in Stevens' example the IP address is always the first argument passed to the program, and so is always in the same place within argv[]. In our code, however, we have a variable number of command line arguments, so we cannot guarantee that the IP address and port number will always be in the same place. How is this handled in our code (i.e. how and where are the IP address and port number of the sink stored for future access?)

Section I: Coding *tcp_gen* & *tcp_sink*

Before you start this lab, make sure that you can access NAS (refer to ECE4110 Lab 1). After you are sure that NAS is mounted and ready for access, perform the following steps to copy and extract the source code that you will need to understand, comment, and modify:

$cd /mnt/nas/Lab2 <ENTER>
$cp lab2.tar ~/stevens_book/unpv12e/lab2.tar <ENTER>
$cd ~/stevens_book/unpv12e <ENTER>
$tar –xvf lab2.tar <ENTER>
$cd lab2 <ENTER>
$ls <ENTER>

Now, you will see a list of files. You will make modifications to a traffic generator program named *tcp_gen.c* and to a receiver program named *tcp_sink.c*. Then, your job is to understand, comment, and complete these two pieces of software. To compile and link them, you will use the command *$make <ENTER>*.

As a starting point, look in the source code of the provided *tcp_gen.c* and *tcp_sink.c* for specifications of exactly how the programs should perform (on the first lines of these files, there is a description on how to execute them from the command line).

You may find that some declared variables are not needed, so make sure that you fully understand the data flow within the programs. Also add some code to print out the number of MBITS PER SECOND that were sent as well as received. This statistic should be written to the screen after the <CTRL+C> key sequence is used to stop the program.

Section II: Getting data from *tcp_gen* & *tcp_sink*

At this point you programs should finished and ready to perform data transmission. The scenario to execute these programs should be as follows:

To run under X Windows:

1. startx <ENTER> (If you have already started X Windows, do not type *startx*)

2. Open 2 terminals (by right click the mouse; choose the 'New Terminal' menu option)

2.a) in one terminal, type

$./tcp_sink -D - p 5000 <ENTER>

2.b) in the other terminal after the first terminal is started, type

$./tcp_gen -D -P 5000 -a 57.35.6.(your_ own_IP_address_last_byte) <ENTER>

3. To stop do type <CNTRL C> in the ./tcp_gen terminal. You will see some summary results in both terminals.

4. You will need to do a <CNTRL C> in the ./tcp_sink terminal to stop that process also.

Note that you will need to do a *ps* and *kill* all of your traffic generator processes before you can run this again (use *man* to learn more about *ps* and *kill* Linux commands). Note that you may have to use ***ps –ef*** to see your traffic gen/sink processes.

Using a packet size of 1,000 bytes per packet and a packet size variance of zero bytes, determine what the inter-arrival time value you should use to approximately generate 2, 4, and 6 Mbits/second of traffic. Also determine what the maximum number of Mbits/sec you can receive with your traffic generator by experimenting with the inter-arrival times. Fill in the following table and turn it in with your printout:

Input Inter-arrival time in seconds	Mbits/sec **received**
	2 Mbits/sec (With Debug off)
	4 Mbits/sec (With Debug off)
	6 Mbits/sec (With Debug off)
Put IA measured on sender in this box	Debug ON only at the sink. Enter the maximum Mbps for the sink and gen. Sink _____ Gen _____
Put IA measured on sender in this box	Debug OFF at both sink and gen. Enter the maximum Mbps for the sink and gen. Sink _____ Gen _____
Question: Show the math equation and answer to how many Mbits/sec you expect if you were to input a packet size of 500 bytes, zero variance and an inter-arrival time of 0.01 seconds	

When you have completed Sections I and II, demonstrate to the TA the maximum number of Mbits/sec you can generate with a packet size of 1,000 bytes, and a packet size variance of zero. Make sure the TA records the maximum number of Mbits/sec that you demonstrated. Look in the Turn-in list for any deliverables at this point in the lab.

Check-off point: GTA Signature _____ Date _____

Section III: Network Traffic Analysis using *tcpdump* and *ethereal*

For this part of the lab, you will need to run *tcp_sink* on 'gatechftp,' which is a machine that will be used as the receiver side. To make sure your results don't get mixed up with those of other students, you will need to use a unique port. Your port number should be 5000 + <your_group_number> (e.g. for group 1, that's port 5001).

On your machine, open a terminal and type *$telnet 57.35.6.245 <ENTER>* (this will establish a telnet session between your machine and gatechftp). Use *linux_class* as the user and *linux_class* as the password. Once the telnet session is opened, type the following commands.

$cd lab2 <ENTER>

$./tcp_sink –p <your_port_number> <ENTER>

(this executable is already there for you. DO NOT PUT YOUR OWN UDP_SINK CODE ONTO GATECHFTP, USE ONLY THE PROGRAM ALREADY THERE. IF YOU GET A CORE DUMP MAKE SURE NO OLD UDP_SINK PROCESSES ARE RUNNING, IF SO KILL THEM. Use *$ps -ef | egrep "tcp_sink"<ENTER>* (*tcp_sink* between quotes) to check for running processes. (Do not run the sink in debug mode, -D argument)

On your machine, you will have to run *ethereal*, *tcpdump*, and your *tcp_gen* program to generate and capture TCP traffic, so just follow the instructions below.

Later in the semester you will be asked to hand in printouts of Ethereal output. What this means is that you take a SCREENSHOT of the Ethereal window and print it out. DO NOT PRINT USING ETHEREAL'S PRINT COMMAND. Printing the actual packet information for all of the captured packets results in pages and pages of print out, most of which is useless anyway. To take a screenshot, press the "Print Screen" key on your keyboard (upper right), and when prompted save the file as desired. Next, type lpr "filename" in a console window, where filename is the name of the screenshot file (for printer setup instructions, see the Appendix).

In one terminal, start network analyzer (*ethereal*) by typing *$ethereal & <ENTER>* or clicking the Redhat icon in the lower left corner of your screen and selecting System Tools => Network Analyzer.

View=>Time Display Format=>Seconds since previous packet

Under capture => Options

Turn on 'capture packets in promiscuous mode'

Turn on 'update list of packets in real time'

Turn on 'automatic scrolling in live capture'

Turn off 'Enable MAC Name resolution'

Turn off 'Enable network name resolution'

Turn off 'Enable transport name resolution'

click [OK] button

In a second terminal, type *$tcpdump –S –nli eth0 <ENTER>* to start *tcpdump* and make this window as wide as possible so it will be easier to read. (You can see how to use *tcpdump*, by typing '*man tcpdump'*)

In a third terminal, type *$./tcp_gen –D –i 0.2 –P <your_port_number> –a 57.35.6.245 <ENTER>* to start your TCP traffic generator.

Type *<CTRL+C>* in traffic generator window after 10 to 20 packets have been transmitted. Stop *ethereal* from collecting any more data by clicking the [Stop] button. Type *<CTRL+C>* in *tcpdump* window to stop it from collecting any more data. On gatechftp, stop your sink from running by pressing *<CTRL+C>*. Then, type *$ps -ef | egrep "tcp_sink"<ENTER>* and find the process corresponding to the *tcp_sink* you were running by looking at the port numbers. Once you find that process, the process ID number for that process is located in the second column from the left. *Kill* the process using that number. **Make sure you don't kill some other student's *tcp_sink*.**

Q1. How do you think the type of data from *tcpdump* compares to the type of data from *ethereal*?

Q2. Which do you like better?

Look at the protocol detail available in *ethereal*. With the mouse, select one of your TCP datagrams. In the lower window, expand 'Ethernet II' by clicking on the plus sign. Answer the following questions:

Q3. What is the value of your Ethernet MAC address of the machine you are using?

Q4. What is the value in the 'Differentiated Services Field' (correctly called Type of Service field) in the IP header of this TCP datagram?

Q5. What does the value mean? (Don't ask TA for the answer)

Q6. What was the source port number?

Q7. Where does the source port number come from? (e.g., ports up to 1024 are called reserved ports) (Don't ask TA for the answer) (To see the source port number, expand the field named Transport Control Protocol)

Fill in the following table (for at least 5 of the TCP packets from your *tcp_gen* run) with the information obtained from your *tcp_gen* (with debug mode ON) and the *ethereal*:

As seen on your PC by *tcp_gen*		As observed by *ethereal*	
Packet size (bytes)	Inter-arrival time(sec)	TCP segment size (bytes)	Inter-arrival time(sec) of TCP segments

Q8. What seems to match and what does not match? Why?

Q9. In Ethereal, identify the TCP three-way handshake and connection termination frame sequences. Copy them in the space below. (Include the Source, Destination, Protocol and Info columns)

a) TCP three-way handshake

Source	Destination	Protocol	Info

b) Connection Termination

Source	Destination	Protocol	Info

Q10. In Ethereal, look at the first 200 frames and describe what is going on. Refer to your lecture notes and class books (you can also use the internet). Keywords: three-way handshake, ACK, sequence numbers, SYN, etc.

Q11 Looking at *ethereal*, there is one line of additional data between each of your TCP datagrams you sent out. What is this? Explain.

Turn-in List

1. Turn into the TA (at the end of Section II) a hardcopy printout of your code. Make sure that you **highlight** the changes that you made. (note: highlight is different than underline, enclose in a box, etc) We now have a printer in the lab. Directions for setting up the printer and printing are given in the appendix and on the printer itself. Please try not to waste paper, i.e., do not print another whole copy of a file for a few small changes, just mark them in pen.
2. The GTA check-off sheet along with the completed table in Section II.
3. Questions and Answers in Section III.
4. The completed table in Section III.

Grading notes

1. How well documented and well commented your programs are. Neatness and style does count. You should improve on this even in the parts that we provided to you.
2. Group number and group member names in the very first comments of both of your programs as well as the date when the file was modified should be included.
3. You will want to put your programs on a floppy occasionally, so if you loose data on your Hard Drive in a crash, you will not lose your programs. The appendix contains some guidelines on how to do that.

Appendix A: Useful tips

Note: You can always use *man Linux command* (For example, *man mount*) to get information about Linux commands.

To use a floppy:

cd / <ENTER>
Now, you should insert the floppy
mount /dev/floppy <ENTER>
If you are in the directory where the file you want to copy is located:
cp <filename> /media/floppy <ENTER>
To check to see if the file is now on your floppy
cd /media/floppy <ENTER>
ls <ENTER>
Before you remove the floppy
You should type
cd / <ENTER>
umount /dev/floppy <ENTER>
Now, you should remove the floppy

To put a file from floppy onto the Linux machine:

cd / <ENTER>
Now, you should insert the floppy
mount /dev/floppy <ENTER>
cp media/floppy/<filename> <destination path> <ENTER>
Wait a while… this puts the files in <destination path>
Before you remove the floppy
You should type
cd / <ENTER>
umount /dev/floppy <ENTER>
Now, you should remove the floppy

To Setup the Printer

-Go to the **Red Hat icon => System Settings => Printing**
-On the GUI, click **New**
Name: **printer_4112** (typed in)
Queue Type: select **JetDirect** Printer (radio button)
Forward
Printer: **57.35.6.9** (typed in)
Port: **9100** (typed in, but should be **default**)
Forward (to Drive select)
Select **HP => LaserJet4000T/TN**
Apply

To print from a root prompt, just type: **lpr <filename>**
You can also print from Gedit, but the default paper is A4, even though the printer is set up for US letter, so it needs to be changed before printing.