Programming Assignment 1 – Echo Server Using tcp and udp Sockets

Computer Networks
Fall 2014
Due date: 10/9, Thursday

August 28, 2014

This programming assignment involves

1. Use Wireshark to see the packets exchanged while running some network programs.
2. Complete the echo server with tcp stream.

1 Echo Client and Server (datagram version)

You need to download and compile the echo client and server source files: *udpClient.c* and *udpServer*.c which we have demonstrated in our lecture. The files are available from ANGEL under the content area. These files are also available on the server under /home/networkos/shared. Make sure you have completely understood the code before starting to run it. The command that you can use to compile/build these files is gcc. Use man gcc to see the compile options. But usually, the format we follow is

$gcc\ -o\ -Wall\ executable\_file\_name\ source\_file\_name.c$

You can run the executable file by typing ./executable-file-name .

Follow the procedures below to use Wireshark on the communications between client and server with datagram version.

1. Log in to the host linux machine. Its ip address is 172.26.91.120 and the port for ssh is the default port 22. Keep in mind to change your initial password in the machine by using the passwd command. Using either Putty (in windows) or terminals in linux system, start up an ssh session logged into the host machine. Run the echo server on the remote machine on port 6000 as written in the code.
2. Now back in your local terminal window. Using another terminal window, start up Wireshark using the command "sudo wireshark" in Linux or simply launch wireshark in windows. You should see a GUI application.

   Select "Options" from the "Capture" menu. Make sure the following options are selected properly.

   In the "Interface" field, select the Ethernet adapter or the wireless adaptor you are using to connect to the internet.

   In the "Capture Filter" text field, enter the following:
   udp port 6000

   This will select only those packets related to the echo client and server interaction.

   You may also want to check the "Update list of packets in real time" box.

   There are lots of checkboxes here, but you can use the defaults in most cases. Make sure that the box labeled "Capture packets in promiscuous mode" is checked.

Another checkbox gives you the option of updating the capture window in real time. The alternative is to display packets only after capturing has stopped. Either way works fine.

3. Start packet capture by clicking the "Start" option in the "Capture Options" screen or from the pull-down "Capture" menu.

   Start the echo client on your local machine, giving the address of the remote machine as its parameter. Enter three short phrases and see that they are echoed by the server. Then use control-D to indicate end-of-file.

4. Stop packet capture by clicking the "Stop" button in Wireshark's capture window.

   Now answer the following questions:

   (a) How many packets have been captured and caught by the filter?

   (b) How many packets were sent from your local machine and how many from the remote machine?

   (c) What is the Ethernet address of the local machine? What is the Ethernet address of the remote machine?

   (d) Choose one of the data packets sent from the local machine to the remote machine.

      i. What was the text message you entered at the keyboard?

      ii. How many bytes of application data are in the frame? (That is, the string you entered at the keyboard.)

      iii. How many bytes are in the UDP header?

2 Echo Client and Server (stream version)

In this part of the assignment, we will transfer information, i.e. echo inputs using the tcp protocol. The tcp echo client code is provided at ANGEL, i.e. *tcpClient.c*. The server code is provided partially and you should complete the code for the server. Both codes are also available in the server machine under /home/networkos/shared. Hints are given in the code and you should make sure that you do necessary error checking to avoid possible run-time errors.

The requirement of the echo server is: After a connection is made to the server, the server will display the following on the terminal

Connection accepted from client_ip_address port port_number.

Then the server will echo back the input from the client and when disconnected from the client, display connection closed and the total number of bytes echoed. Then the server continues to wait for incoming connections until it is terminated by the user. You may use Wireshark to check if the connection is made correctly or if the data is sent properly.

After finishing the server code, you will run the stream versions of the echo client and server and capture the packets using Wireshark.

1. Start up the stream version of the echo server on the remote machine.

2. Modify the "Capture Filter" field by replacing "udp port" with "tcp port" as follows: tcp port 6000

3. Start packet capture by clicking the "Start" option in the "Capture Options" screen or from the pull-down "Capture" menu.

   Now back in your local terminal window, start the echo client, giving the address of the remote machine as its parameter. Enter three short phrases and see that they are echoed by the server. Then use control-D to indicate end-of-file.

4. Stop packet capture by clicking the "Stop" button in Wireshark's capture window. Now answer the following questions:

   (a) How many packets have been captured and caught by the filter?

(b) How many packets were sent from your local machine and how many from the remote machine?

(c) How many packets were used in setting up the TCP connection? How many were used to send data? How many contained only acknowledgments? How many were used to tear down the TCP connection?

(d) What is the Ethernet address of the local machine? What is the Ethernet address of the remote machine?

(e) Choose one of the data packets sent from the local machine to the remote machine.

(f)
i. What was the text message you entered at the keyboard?
ii. How many bytes of application data are in the frame? (That is, the string you entered at the keyboard.)
iii. How many bytes are in the TCP header?
iv. How many bytes are in the IP header?
v. How many bytes are in the Ethernet header?
vi. What is the total number of bytes in the frame?

Your report should include your tcp server code and answers to the questions in both parts. An electronic submission is preferred. You should also include in your submission the packets you captured in part 1 and part 2.

Please sign the honor code "I affirm that I have adhered to the honor code" when you turn in your report.