

Data Communication and Computer Network Laboratory

Master of Computer Application

Second Year, First Semester

Session: 2024-25

Assignment - II

Date: 28/08/2024

Experiment 1

The objective of this laboratory experiment is to understand the details of the Transmission Control Protocol (TCP) operation. TCP is a transport layer protocol. It is used by many application protocols like HTTP, FTP, SSH etc., where guaranteed and reliable delivery of messages is required.

To do this experiment you need to install the Wireshark tool. This tool would be used to capture and examine a packet trace. Wireshark can be downloaded from www.wireshark.org.

Step1: Capture a Trace

- (i) Launch Wireshark
- (ii) From Capture→Options select Loopback interface
- (iii) Start a capture with a filter of “`ip.addr==127.0.0.1 and tcp.port==xxxx`”, where xxxx is the port number used by the TCP server.
- (iv) Run the TCP server program on a terminal.
- (v) Run instances of the TCP client program on separate terminals and send requests to the sever.
- (vi) Stop Wireshark capture

Step2: TCP Connection Establishment

To observe the three-way handshake in action, look for a TCP segment with SYN flag set. A “SYN” segment is the start of the three-way handshake and is sent by the TCP client to the TCP server. The server then replies with a TCP segment with SYN and ACK flag set. And finally the client sends an “ACK” to the server.

For all the above three segments record the values of the sequence number and acknowledgment fields. Draw a time sequence diagram of the three-way handshake for TCP connection establishment in your trace. Do it for all the client connections.

Step3: TCP Data Transfer

For all data segments sent by the client, record the value of the sequence number and acknowledge number fields. Also, record the same for the corresponding acknowledgements sent by the server. Draw a time sequence diagram of the data transfer in your trace. Do it for all the client connections.

Step4: TCP Connection Termination

Once the data transfer is over, the client initiates the connection termination by sending TCP segment with FIN flag set, to the server. Server acknowledges it and sends it's own intention to terminate the connection by sending a TCP segment with FIN and ACK flags set. The client finally sends an ACK segment to the server.

For all the above three segments record the values of the sequence number and acknowledgment fields. Draw a time sequence diagram of the three-way handshake for TCP connection termination in your trace. Do it for all the client connections.

Experiment 2

The objective of this laboratory experiment is to look at the details of the User Datagram Protocol (UDP). UDP is a transport layer protocol. It is used by many application protocols like DNS, DHCP, SNMP etc., where reliability is not a concern.

Step1: Capture a Trace

- (i) Launch Wireshark
- (ii) From Capture→Options select Loopback interface
- (iii) Start a capture with a filter of “`ip.addr==127.0.0.1 and udp.port==xxxx`”, where `xxxx` is the port number used by the UDP server.
- (iv) Run the UDP server program on a terminal.
- (v) Run multiple instances of the UDP client program on separate terminals and send requests to the sever.
- (vi) Stop Wireshark capture

Step2: Inspect the Trace

Select different packets in the trace and browse the expanded UDP header and record the following fields:

- Source Port: the port from which the UDP segment is sent.
- Destination Port: the port to which the UDP segment is sent.
- Length: the length of the UDP segment.