National University of Computer and Emerging Sciences

Laboratory Manuals

for

Computer Networks - Lab

(CL -3001)

File Name	SecX1/X2_RollNo_NameOn Flex
Submission Requirement	1 Word file with task-wise solution and **screenshots**. 2 **Code files and output.**
Semester	Fall 2024

Department of Computer Science
FAST-NU, Lahore, Pakistan

Lab Manual 08

Objective:

- Observing the structure and working of **TCP** & **UDP** Protocols in Wireshark.
- TCP Socket Programming using Multithreaded Server to handle multiple clients at the same time

Lab Statement 1: Analyzing TCP Packets using Wireshark **(10)**

Step 1: Run Wireshark.

Step 2: Load the trace file **tcp-ethereal-trace-1**

Step 3: Now filter out all TCP packets by typing "tcp" (without quotes) in the filter field towards the top of the Wireshark window. You should see a series of TCP and HTTP messages between the host in MIT and gaia.cs.umass.edu. The first three packets of the trace consist of the initial *three-way handshake* containing the SYN, SYN ACK and ACK messages. You should see a series of "TCP Segment of Reassembled PDU" messages being sent from the host in MIT to gaia.cs.umass.edu. Recall from the previous lab that there is no such thing as an HTTP Continuation message – this is Wireshark's way of indicating that there are multiple segments being used to carry a single HTTP message. You should also see TCP ACK segments being returned from gaia.cs.umass.edu to the host in MIT.

Question 1: What is the IP address and TCP port number used by the client computer (source) that is transferring the file to gaia.cs.umass.edu?
Question 2: What is the IP address of gaia.cs.umass.edu? On what port number is it sending and receiving TCP segments for this connection?
Question 3: What is the sequence number of the TCP SYN segment that is used to initiate the TCP connection between the client computer and gaia.cs.umass.edu? What is in the segment that identifies the segment as a SYN segment?
Question 4: What is the sequence number of the SYNACK segment sent by gaia.cs.umass.edu to the client computer in reply to the SYN? What is the value of the Acknowledgement field in the SYNACK segment? What is it in the segment that identifies the segment as a SYNACK segment?

Question 5: In packet 9, **Ack = 2026** and **Seq = 1**. Explain these values?
Question 6: In packet 16, **Ack = 7866** and **Seq = 1**. Explain these values?
Question 7: Why Wireshark uses relative sequence and ack?

Lab Statement 2: Analyzing UDP Packets using Wireshark (5)

Step 1: Run Wireshark

Step 2: Load the trace file **dns-ethereal-trace-2.**

Step 3: Now filter out all non-UDP packets by typing "udp" (without quotes) in the filter field towards the top of the Wireshark window

Step 4: Analyze the UDP Packets and answer the following questions

Question 1: Select the first DNS packet in the trace. Determine, how many fields there are in the UDP header

Question 2: From the packet content field (click on any header and observe the display in the Packet Bytes Window), determine the length (in bytes) of each of the UDP header fields.

Question 3: The value in the Length field is the length of what? Verify your claim using the selected packet.

Question 4: What is the port number to query the DNS Server?

Lab Statement 3: Multithreaded ECHO server using TCP (10)

You are required to design a **Multi-Threaded Echo Server and a Simple client**. The server uses a TCP protocol to connect to clients. Server will be listening for clients to connect to it and as soon as a client connects, it assigns a separate thread for further processing. The thread will be responsible to receive the data from the client and *echo* it to the client until the client sends the "DISCONNECT" command. **The server can handle maximum 3 clients at a time.**

Client will be a simple program which after connecting to the server will take the input from the user and send it to the server, then outputs the response on the terminal received from the server. It will do the same until user enters "DISCONNECT". Upon entering "DISCONNECT" the client shall close the socket and exit.

Following are the steps which Server should perform:

1. Receives a connection request from client and pass the socket descriptor returned by the accept() to the thread and goes back to listen for more connections for **clients< 4**. If the fourth client tries to connect then server sends the client message that "**Server Full**"

2. Meanwhile this is what the thread do
   - Receive what the client sends.
   - Echo back what client sends.
   - If client has sent "DISCONNECT" then close the socket and quit.

3. Receive more data from the same client.

You should cater 3multiple clients that will be sending connection requests to server.

Following are the steps which Clients should perform:

1. Take input from the user.
2. Send input to the server
3. If input is “DISCONNECT” then close the socket and exit otherwise continue to step 4.
4. Wait for server’s response.
5. Print the server’s repose.
6. Go back to step 1.