CSE 5231 – Spring 2024

Computer Networks

Homework #2

Due: 04/07/2024 11:30PM

Note: For any questions, please make sure to attach some proof (e.g., screenshots) for your own work. At least one of the screenshots for each question should include the timing information of the PC. Also, attach your pcap files with your submission.

Q1) Please capture your random/daily network traffic via Wireshark. Demonstrate the fields in the network and transport layer of 2 different application packets and discuss each field in short (why we need that field, etc.). Please attach the packet capture to your submission. (20p)

Q2) Display your PC's ARP and routing table via the terminal. Add a new ARP and routing table entry manually and display it again to prove that they are added properly. Shortly explain what these tables mean and what you actually do when manual entries are added to these tables. (10p)

Q3) Run traceroute commands for 10 different servers in one single country that is not located within North America (US, Canada). Display the results (terminal output is fine) and have an insightful discussion on them. Please come up with some ideas of what the network topology in the Internet backbone for your traffic looks like and explain (considering only these 10 servers). (10p)

Q4) Generate a DNS lookup to your high school website. Display and explain the packet exchanges from your machine for this DNS lookup. If your high school does not have a website, prove that it does not have it with the DNS lookups by assuming the high school website would be yourhighschoolname.edu.org. Then, do the same thing for your undergrad institute. (10p)

Bonus: Do you think the response would be the same if you run the same query again tomorrow? What about next year? Why/why not? Please discuss shortly. (**5p**)

Q5) Find an HTTP website and visit that via any browser. Before visiting this website, start Wireshark and capture the traffic, and discuss the details of the Wireshark packet capture (only for this HTTP traffic). Please list and explain all the packets involved for the sake of this traffic by specifying why we need those packets. (15p)

5.1. How many routers are there between your PC to this webserver hosting the website? How did you find it? (5p)

Bonus: Please consider the following scenario and answer accordingly: You bought a new PC, connected to your home router, and the first and only thing you do on your PC to access this exact http website. Please list all communication/protocols that may need to happen in the background. (**5p**)

Q6) Socket programming: Implement a client and a server that communicate via UDP. The port numbers should be your 1-(yourFITIDLast3). Read from the text file given and send it using these UDP programs from client to server. You can use a library for this task, such as Scapy or any other tool of your choice. Show these packets with Wireshark capture. (10p)

6.1. As the next step for this question, do the same thing with TCP. What are the differences you can observe? (10p)

Example: Let's say my FIT last 3 is: 123, the port numbers will be 1123.

For both of these, please attach your codes as well as pcaps to your submission. No code/no pcap will get 0.

6.2: Create multiple clients connecting to the same UDP server and send your name+surname from the second client while the first one is still sending the text file provided. Display the packet captures via Wireshark. Your Wireshark screenshots should clearly display the traffic for this question (i.e., filter the traffic). (5p)

6.3: Do 6.2 via TCP. (5p)

6.4: (Bonus): Install mininet into your machine (it requires Linux systems, VM would work). Run these programs within mininet network using the hosts in the simulation environment. (**20p**)

Q7 (Bonus): Implement the TCP handshake from scratch. Send SYN message from your PC to google.com and capture SYN-ACK reply message. Then, send an ACK with a wrong sequence number (ACK = #FIT-ID). Display your network traffic in Wireshark and explain what happens after the ACK is sent. (**10p**)

Q8 (Bonus): Implement a DNS recursive resolver and make the iterative and recursive calls through a client on the same machine to this resolver. (**10p**)