CSC329 - Computer Networks
Assignment-1

King Saud University
Department of Computer Science
Fall 2021
Dr. Azzam Alsudais

Assignment Instructions

- **Teamwork**: Students need to perform this assignment in teams of only 2 students.
- **Academic integrity**: The work in this assignment should be the student's own original work. Any form of plagiarism will not be tolerated, and will result in a zero mark for the whole assignment.
- **Grading**: Grading is going to be offline. However, the instructor reserves the right to perform interview-based grading, in which students answer additional questions about their work on the assignment.
- **Submission**: See the end of this document for submission instructions.
- Please read the entire assignment prior to solving any question.

Software Requirements

- **Wireshark**: You will need to have Wireshark installed for this assignment.
- **Operating System**: You may use whatever OS you find yourself comfertable with.
- **Internet Connection**: You will have to have a stable Internet connection in order to carry out the work required for this assignment.

Due Date is Oct 28, 2021 at 11:55PM

Student-1 Name: .....

Student-1 KSU-ID: .....

Student-2 Name: .....

Student-2 KSU-ID: .....

Introduction

The goal of this assignment is to familiarize yourself with network traffic analysis. Each question is designed as an experiment; As such, you are given some instructions that should help you conduct each experiment.

Experiment-1: Video Streaming

In this experiment, your job is to analyze video streaming network traffic.

Steps

Before you perform the two runs below, locate a Youtube video that has a 4k version as well as a 144p version, but don't play the video just yet. **Note**: to make your life easier, make sure you close any browser tabs/windows as well as any other applications that connect to the Internet while you perform this experiment.

Run-1

1. Set the video quality to 4k (2160p).
2. Start capturing network packets in Wireshark.
3. Let the video play for a period of ≈4 minutes.
4. Stop the video and stop capturing packets in Wireshark.
5. Save the capture file (File: Save), and set the file name to "high-quality".

Run-2

Repeat the same steps in Run-1, but this time select the lowest quality (144p), and save the capture file as "low-quality".

Questions

Q1: Traffic Metrics (15 points)

Open each of the capture files in Wireshark, and fill in the empty fields in Table 1. **Note**: make sure you consider only Youtube traffic when filling in the information in the table.

Table 1: Experiment-1 Required Information

Your host's MAC address		
The destination MAC address		
Your device's IP address		
Youtube server's IP address		
Youtube's server port number		
Your host's port number		
Metric	**Run-1**	**Run-2**
Capture file size (in MB)		
Total number of bytes transferred		
Total number of packets		
Number of TCP segments		
Number of TCP SYN segments		
Number of TCP PUSH segments		
Number of TCP ACK segments		
Number of TCP SYN segments		
Number of TCP RST segmentsc		
Number of TCP FIN segments		
Number of non-TCP packets		

Q2: Explain why video streaming services usually use TCP as an underlying transport service. (5 points)

Q3: WHOIS Youtube (10 points)

Perform the `whois` command on the Youtube server's IP address.

1. Who does the IP address belong to (OrgName/person)?

2. In which city does the IP address resides?

3. Is the OrgName related to Youtube/Google? If not, Explain why.

Experiment-2: Domain Names and Internet Routes

The goal of this experiment is getting yourself familiar with trace routing and domain names. Useful commands: `traceroute`, `nslookup`, and `whois`. **Note: each student needs to perform this experiment at his home.**

Domain Names

This is a list of domain names that we are going to focus on in this experiment.

1. example.com
2. google.com
3. twitter.com
4. facebook.com
5. youtube.com
6. netflix.com
7. alriyadh.com
8. ksu.edu.sa
9. ccis.ksu.edu.sa

Questions

Answer the following questions to the best of your knowledge.

Q1: Web Servers IP Addresses (10 points)

For each of the domain names above, find out the IP address of the web server hosting that domain name. Fill in the empty fields in Table 2. **Note**: each student is requested to fill in the corresponding column. **Each student needs to perform this experiment at his home (access networks need to be different)**. Therefore, it is normal to have different answers.

Q2: Is the OrgName/person the same for `ksu.edu.sa` and `ccis.ksu.edu.sa`? If not, explain why. (10 points)

Q3: Route Tracing (10 points)

Perform `traceroute` (`tracert` on Windows) for each of the domain names above. Keep track of the following information: *number of hops*, *number of hops timed out*, and *maximum latency* (across all hops). Fill in the required information in Table 3. **Each student needs to perform this experiment at his home (access networks need to be different for each participating student)**

Table 2: Experiment-2 IP Addresses

	Student-1 KSU-ID:		**Student-2 KSU-ID:**	
Domain Name	**IP Address**	**OrgName/person**	**IP Address**	**OrgName/person**
example.com	*93.184.216.34*	*Derrick Sawyer*	*93.184.216.34*	*Derrick Sawyer*
google.com				
twitter.com				
facebook.com				
youtube.com				
netflix.com				
alriyadh.com				
ksu.edu.sa				
ccis.ksu.edu.sa				

Table 3: Experiment-2 Route Tracing Metrics

	Student-1 KSU-ID:			**Student-2 KSU-ID:**		
Domain Name	**# of hops**	**# of timeouts**	**Max latency**	**# of hops**	**# of timeouts**	**Max latency**
google.com						
twitter.com						
facebook.com						
youtube.com						
netflix.com						
alriyadh.com						
ksu.edu.sa						
ccis.ksu.edu.sa						

Experiment-3: Socket Programming

In this experiment, you are asked to write a client-server application (both can run on same host). The client should send information about student-1 (name and KSU-ID, which can be hardcoded for the purpose of this assignment), and the server should respond with student-2 information. An example is provided below.
Example: Suppose student-1 is Abdullah and his KSU-ID is 439000000 and student-2 is Mohammad and his KSU-ID is 439111111. This is an example of the message exchange between the client and the server.

- *client*: "Hello, I'm Abdullah and my KSU-ID is 439000000".
- *server*: "Welcome, I'm Mohammad and my KSU-ID is 439111111".

Setup

- **Two versions**: write two versions of this client-server application, one that uses **UDP** and another that uses **TCP**. You will end up with 4 programs: `udpcleint.py`, `udpserver.py`, `tcpclient.py`, and `tcpserver.py`. You my reuse the code we discussed in class.
- **Hosting**: you may run both the client and server on the same host.
- **Server's port number**: set the server's port number to 10329.
- **Wireshark**: for each of the questions below, you will need to run Wireshark and capture traffic. Make sure you select the loopback interface if you are running both the client and server on your machine.

Questions

Q1: Using UDP as a Transport Protocol (20 points)

In this part of the experiment, you will work on the UDP versions of the client and server.

Steps

1. Open Wireshark and setup the capture interface to the appropriate one (typically this is the loopback interface if you are running both programs locally).
2. Start capturing network packets.
3. Run the UDP server program.
4. Run the UDP cleint, and enter the appropriate string (as explained above).
5. After the server returns a response, hit the "stop capturing packets" button in Wireshark.

1- What is the port number the was assigned to the client socket?

2- How many UDP packets were sent during the communication between the UDP client and server?

3- How many bytes were transferred between the client and the server?

Before the next questions, filter out any packets that do not belong to the communication that took place between the client and server.
4- Attach a screen shot of the Wireshark window that shows the string sent by the *client* program (this is the UDP payload of the packet sent from the client to the server).
Assign "udp-s1" as the name for your screenshot.

5- Attach a screen shot of the Wireshark window showing the string sent by the *server* program (this is the UDP payload of the packet sent from the server to the client).
Assign "udp-s2" as the name for your screenshot.

Q2: Using TCP as a Transport Protocol (20 points)

Perform the same 5 steps of the previous question but for the TCP versions of the programs.
1- What is the port number the was assigned to the client socket?

2- How many TCP packets were sent during the communication between the TCP client and server?

3- How many bytes were transferred between the client and the server?

4- What are the sequence numbers of the 3 TCP segments that belong to three-way handshaking phase?

Before the next questions, filter out any packets that do not belong to the communication that took place between the client and server.
5- Attach a screen shot of the Wireshark window that shows the string sent by the *client* program (this is the TCP payload of one of the packet sent from the client to the server).
Assign "tcp-s1" as the name for your screenshot.

6- Attach a screen shot of the Wireshark window showing the string sent by the *server* program (this is the TCP payload of one of the packet sent from the server to the client).
Assign "tcp-s2" as the name for your screenshot.

Submission

Collect the following files in one folder, compress it (preferably using .zip), and submit the compressed file. **Only one student needs to submit.**

- Your solution as a PDF file. Answers need to be written in this PDF file (not in a new document).
- The source code for the client and server programs (for both UDP and TCP, so there needs to be 4 versions.
- The screenshots that are requested in experiment 3.

Submissions should be uploaded to LMS.