

Assignment 3: Reverse-engineering HTTP video streaming

Different video streaming services adopt different strategies to deliver videos to end hosts. In this project, you will identify what strategy is used by analyzing the network traces. Specifically, use Wireshark (or any other tools) to capture video streaming traces from YouTube, DailyMotion, and Vimeo.

Capture traces for three different end-host devices. Submit your video streaming traces (5 points each).

- a. A laptop that connects to a WiFi network.
- b. A smartphone that connects to a 4G cellular network. You will need to do some research to find out how to capture packets on smartphones.
- c. A smartphone that connects to a WiFi network.

Include details in your report to specify how you captured the traces on the smartphone (10 points)

Part A: To be completed for all three traces (case a, b, and c)

Is the video server using persistent/non-persistent TCP connection? Is it running multiple TCP connections or a single connection during a video streaming session? Explain how you arrive at the conclusion. Include screenshots to highlight specific parts of your trace. (15 points)

Part B: To be completed for all three traces (case a, b, and c)

Is the video server using constant bit-rate or variable bit-rate streaming? If the latter, how does it adapt its streaming bit-rate over time and according to link quality? Explain in detail how you arrive at the conclusion. (15 points)

Part C: To be completed only for traces captured using smartphones (case b and c)

Try to vary the link quality (e.g., making the link weaker by hiding your device from the access point or base station) and see how the streaming service behaves. Explain your observations in the report. (10 points)

Part D: To be completed for trace captured in case a only

Identify the service type using DNS traffic. DNS queries can be used to identify to which service a remote IP belongs to. Using the pcap trace downloaded from each video

service, identify which IP address belongs to YouTube, DailyMotion, and Vimeo. (15 points)

Part E: To be completed for trace captured in case a only

Calculate network counters. Network counters are useful for quality inference because they reveal how much data is exchanged between two IP addresses. Collect all network counters exchanged between the local machine and YouTube's servers. (10 points)

Part F: To be completed for trace captured in case a only

Infer video segment downloads. Capturing the segments that are downloaded during the streaming session is the most useful information for inferring the video quality. In this part, detect when a segment is downloaded and its size. Do this only for YouTube's traffic. (10 points)

Submission instructions

You may write your programs in the following languages: Python. If you want to write in any other language, please talk to me.

You need to submit your homework in a single zip file as follows:

- Part D, E and F should be analyzed programmatically. Please submit your code as well.
- The zip file and (the root folder inside) should be named using your last name, first name, and the homework name, all separated by a dash ('-') e.g. lastname-firstname-HW3.zip
- The zip file should contain all submissions requested in Parts A through F. Have a separate subfolder for each part in your zip file.
- You should provide a README.txt file describing how to run your programs in each part when applicable.

Appendix

- You can use Scapy to read and process packets from a trace. <https://scapy.net/>
- To read and process DNS queries, you can use a library called dnslib: <https://github.com/paulc/dnslib>
- You can also use dpkt library if that helps.
- You are allowed to replace ONE video streaming service with a service of your choice. For example, if you are facing challenges with Vimeo, you can replace it with Netflix or another video streaming service.
If you choose to do so, please remember to mention it in your report, and the reason for the replacement.

