

# Homework 1

CS341 Introduction to Computer Networks, KAIST  
(2024 Fall)

Due: 11:59 PM (KST), October 1, 2024

## INSTRUCTIONS TO STUDENTS:

- Any collaboration or assistance of any kind is strictly prohibited; all work must be your own.
- Your submission *will surely* be compared with the submissions for your peers for plagiarism detection. Any academic dishonesty will be directly reported to the university.

## 1. Measuring Network Topology Around Your Campus Network (40 points)

For this homework question, your task is to measure the network topology around the KAIST campus network. The final output of this task is to present a tree of network topology with the root being the KAIST network. The network topology should be presented at the Autonomous System (AS) level granularity. Autonomous System (AS) is a collection of IP networks and routers under the control of a single entity in the Internet and each AS has its unique AS number. For example, the KAIST campus network makes one AS with its AS number being 1781. You are asked to find ASes that are direct neighbors of the KAIST AS and the ASes that are neighbors of the direct neighbors. That is, you are asked to find two-hop neighbors of the KAIST AS. Feel free to look up the textbook to understand the concept of ASes and the AS-level topology.

For this task, you will need to conduct **traceroute** measurements. This tool sends packets to a destination of your choice and records the IP addresses of the routers that the packets traverse. **whois** is another tool that you can use to find the AS number of an IP address; for example **whois -h whois.cymru.com " -v 143.248.155.65"**. Using these tools, you can learn the AS number of each router in the path from a source to a destination of your choice.

Your final submission file **as-level-topology.txt** should contain multiple lines of three AS numbers separated by a dash. For example, if you find the KAIST AS (1781) is directly connected to AS 1234, which is directly connected to AS 5678, you can write the first line in the topology file below.

```
(read as-level-topology.txt)
1781-1234-5678
1781-1234-6789
1781-2345-7890
...
```

You can list as many such triplets as you find through route measurements.

For each line in the topology file **as-level-topology.txt**, you must include a separate traceroute measurement file **traceroute-<AS1>-<AS2>-<AS3>.txt** that contains the traceroute measurement that you used to find the AS-level path **AS1-AS2-AS3**. For example, if you have a line **1781-1234-5678** in the topology file, you must include a file **traceroute-1781-1234-5678.txt** that contains the traceroute measurement that you used to find the AS-level path **1781-1234-5678**.

Hint: You can conduct `tracert` from a computer in the KAIST network to any destination IP address. You may also try `tracert` to the KAIST network from some vantage points outside the KAIST network (e.g., looking glass servers that are hosted by many major ISPs).

Locate all the files in a directory named `hw1-task1`, which is under the root directory of your submission file.

```
./
./hw1-task1/
./hw1-task1/as-level-topology.txt
./hw1-task1/tracert-1781-1234-5678.txt
./hw1-task1/tracert-1781-1234-6789.txt
./hw1-task1/tracert-1781-2345-7890.txt
...
```

## 2. Youtube Traffic Analyzer (40 points)

Dynamic Adaptive Streaming over HTTP (DASH) is a popular video streaming protocol that is used by many video streaming services, including Youtube. In DASH, a video is divided into small chunks, and each chunk is sent to clients with some time interval (typically a few seconds). The size of each chunk is determined by the video content and studies have shown that the varying pattern of the chunk size can be used to infer the video content.<sup>1</sup>

Your task is to write a simple program that measures the chunk sizes of a Youtube video you play on your computer. You can use any programming language of your choice. You need to submit the source code of your program and the segment size measurements in `youtube_burst_measurements_#.csv`, where `#` is the index of the Youtube video you analyzed; see the Appendix for the Youtube video links. You are required to make one measurement (about 10-minute playtime) for the assigned video; yet, you can voluntarily make more measurements for the same video and submit them in separate lines of the same csv file.

```
./
./hw1-task2/
./hw1-task2/source-code/
./hw1-task2/source-code/(source code of your program)
./hw1-task2/measurement/
./hw1-task2/measurement/youtube_burst_measurements_#.csv
```

**Note:** To simplify the task, you are asked to fix the video quality to 720p and play the assigned video in a web browser on your computer without skipping or pausing. When conducting the measurement, make sure that you are playing only the assigned video and no other videos are being played on your computer at the same time.

## 3. Academic Integrity (20 points)

(Yes or No) I have read and understood the KAIST SoC Honor Code.

```
./
./hw1-task3/
./hw1-task3/my-answer.txt
```

---

<sup>1</sup>For example, see the paper: Schuster, Roei, Vitaly Shmatikov, and Eran Tromer. "Beauty and the burst: Remote identification of encrypted video streams." *USENIX Security* 2017.

## Submission

- Please **zip** the root directory of your submission files and name it as **hw1 - #.zip**, where **#** is your student ID.
- And then, submit the single zip file to the KLMS system.

## Appendix: Youtube Videos

Pick the youtube video that matches the last digit of your student ID. You will use this video to measure the segment sizes in the Youtube Traffic Analyzer task above.

0. [https://www.youtube.com/watch?v=ZJl\\_9cpaFjY](https://www.youtube.com/watch?v=ZJl_9cpaFjY)
1. <https://www.youtube.com/watch?v=V3uAanKBsP8>
2. <https://www.youtube.com/watch?v=LN4zQHuaM-M>
3. <https://www.youtube.com/watch?v=vzfpvc2nJMA>
4. <https://www.youtube.com/watch?v=nzr3ZFR5Mb4>
5. <https://www.youtube.com/watch?v=LqmiC3bWvDs>
6. <https://www.youtube.com/watch?v=XFFgvWwcptk>
7. <https://www.youtube.com/watch?v=4zDrRbx djic>
8. <https://www.youtube.com/watch?v=lEDY5qsifsc>
9. <https://www.youtube.com/watch?v=nVlViLDUf2A>