Homework 2

CS341 Introduction to Computer Networks, KAIST
(2024 Fall)

Due: 11:59 PM (KST), October 30, 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.
- Word limit: Your answers should be within the word limit when it is specified in each question. We will *not* read beyond the word limit when grading.

1. Selfish TCP Congestion Control (50 points)

Suppose you're downloading a large file from a web server that implements the standard TCP congestion control algorithm. You want to achieve an unfairly large download bandwidth (say 10x-100x the typical rate). You have already tried a download accelerator that creates multiple TCP connections to the server (to increase the aggregate download rate). The download accelerator works to some extent, but the server seems to limit the number of concurrent connections per client to a few connections, thus limiting the overall download rate. So, alternatively, you plan to manipulate the TCP acknowledgments by sending ACKs with artificially higher ack numbers, even if some segments are lost due to packet loss.

1. Explain how this strategy could potentially allow you to achieve an unfair bandwidth advantage. In your answer, discuss (in no more than 200 words):
   - How TCP congestion control reacts to acknowledgments.
   - Why sending ACKs with higher ack numbers could increase the server's transmission rate.
   - The effect of skipping over lost segments and how TCP might behave in response.
2. Analyze the risks and potential downsides of this approach. Consider (in no more than 200 words):
   - How the server might detect this manipulation.
   - Potential negative impacts on the TCP connection's reliability (e.g., incorrect retransmissions, data corruption).
   - Whether there are any mechanisms in modern TCP stacks that might mitigate this type of behavior.
3. Discuss the ethical implications of using such a strategy. Consider (in no more than 200 words):
   - How this unfair behavior affects the overall health of the network.
   - What consequences might occur if multiple users adopted similar tactics on the same network.
4. From the perspective of an unethical receiver, propose a strategy to effectively address the problem of data corruption due to missing segments (resulting from sending artificially higher ACKs). How could this strategy help in achieving a faster download speed while still ensuring complete data integrity? (in no more than 200 words)

2. Observing TCP Congestion Control Behavior (50 points)

In this homework assignment, you are asked to observe the congestion control behavior of a TCP connection on your computer. Specifically, you should observe how the number of segments sent (if your computer is the sender) or received (if your computer is the receiver) changes as the congestion window size adjusts over time. To do this, follow these steps:

- Set up a connection between your computer and a remote host that you will use for this observation. You may set up your own remote host (e.g., a virtual machine, a cloud server, etc.) or use an existing one (e.g., any public server that hosts a large file). You are encouraged to make the round-trip time (RTT) between your computer and the remote host to be not too small to observe the congestion control behavior more clearly. If the two hosts happen to be too close (e.g., in the same local network), you can artificially increase the RTT using a tool like `tc`.
- Use Wireshark (or `tcpdump`) to capture the packets exchanged between your computer and the remote host during a TCP session (e.g., `iperf3`).
- Analyze the captured packets to observe how the number of segments sent/received changes over time. For your observation, you are suggested to use the RTT between your computer and the remote host as a reference.

Instructions:

1. Explain your setup and the steps you took to observe the congestion control behavior (in no more than 300 words). Provide details on:
   - The environment you used (e.g., virtual machines, cloud servers).
   - How you initiated the TCP connection.
   - How you used Wireshark (or `tcpdump`) to capture packets.
2. Analyze your packet capture (in no more than 300 words). In your analysis:
   - Explain how the number of segments sent/received changes over time. Include one or more plots that illustrate the behaviors you observed. Make sure your plots are clearly labeled and easy to understand for grading. Readability of the plots will be evaluated as well.
   - Describe the TCP congestion control behaviors you observed from the plots. Clearly mark on the plots the key points that demonstrate the congestion control behaviors.
   - Attach the Python script with the name `analyze.py` that you used to generate the plots.[1] Include specific examples (as inputs to your script) from the Wireshark capture that demonstrate the congestion control behaviors. Executing `analyze.py` should read `dump.pcap`, parse it, then silently (without any interaction with user nor popup windows) produce `plot.png` that contains all plots you have used.
   - Include `requirements.txt` file that lists all Python 3 modules you have used in `analyze.py` in pip freeze format. You can use any Python 3 modules that support Ubuntu 22.04 Python 3.10. You can create the file using `pip3 freeze > requirements.txt`
   - Attach your packet capture file `dump.pcap`.
   - Attach your plot image file `plot.png`. If you have multiple plots, put all of them in a single file.

Submission

- Place your submissions in following structure:

[1] The word limit does not apply to the script.

```
/
|- 1-1.txt
|- 1-2.txt
|- 1-3.txt
|- 1-4.txt
|- 2-1.txt
|- 2-2.txt
|- analyze.py
|- plot.png
|- dump.pcap
|- requirements.txt (if needed)
```

- `N-M.txt` contains your answer of Problem `N` Subproblem `M`. We will count words of txt files; include your answers only, excluding problem number, problem description, or others.
- `analyze.py` must create `plot.png` using `dump.pcap`. You can assume that executed path is `/` and `open('./dump.pcap', 'rb')` works.
- Your Python script must be runnable with `pip3 install -r requirements.txt; python3 analyze.py` on a fresh Ubuntu 22.04 machine. Please do not use deprecated Python features and avoid using too new (above 3.10) features. This should silently (no popup window) produce `plot.png` with exactly same content with the file you have provided.
- `zip` all the files into a single `zip` file (zip file name does not matter), and upload it to GradeScope. The GradeScope system will unzip your submission, display your files (not a folder).