Project 2: Adaptive bitrate streaming

Future Internet

ETH Zürich - Spring Semester 2021

General Future Internet project stipulations:

- This is 1 of 4 projects. Together they account for 50% of your final course grade.
- Always cite and reference appropriately. Do not use other students' code outside of your group.

This particular project stipulations:

- Submission deadline: **19 April 2021 at 15:00**.
- Threshold A deadline: **14 April 2021 at 15:00**. If *threshold A* is not reached by this deadline, the total number of points will be reduced by 20%.
- The project must be done in groups of up to two people.
- You can receive 12.5points for this project: achieving threshold A (4pt), achieving threshold B (8pt), optimize your solution - up to 12.5points.
- After the deadline, you will have an interview. Each group member must be able to individually explain and demo all parts.
  Interview date: **20 April 2021 at 15:15**.
  Not being able to defend the project solution means zero points from this project for both group members.
- You **must not use git push force** and similar commands to revert existing commits or change their metadata.

1 Introduction

Video streaming services want to provide the highest possible video quality without causing video stalling under various network conditions.
Video providers have every video stored in multiple quality levels (e.g. 360p, 1080p, 4k). Intuitively, each quality level requires proportional network bandwidth. Under ideal network conditions (i.e. constant bandwidth), there is no need for adaptive delivery. However, when bandwidth changes are very common, for example with mobile devices using cellular networks, we want to deliver video adaptively and change video quality depending on network conditions.
Each video is divided into chunks. In this assignment, we assume that every chunk has a constant duration of 4 seconds. All video chunks are available in 6 quality levels. We have to decide which chunk to fetch. Each quality level (approximately) requires the following network bandwidth:

```
300Kbps, 750Kbps, 1200Kbps, 1850Kbps, 2850Kbps, 4300Kbps
```

Your task is to design an algorithm for video delivery: adaptive bitrate streaming (ABR) algorithm.
Your algorithm is evaluated using real-world traces collected on mobile devices.
User experience is negatively affected by two types of events (1) video stops (rebuffering), and (2) frequent changes in video quality. The optimization goal of this exercise takes these two tasks into account:

```
score = agg_video_bitrate - 4.3 * rebuffer_time - agg_switches_amplitude

agg_video_bitrate - the sum of all chunk bitrates across all traces
rebuffer_time - aggregated rebuffering time across all traces
agg_switches_amplitude - aggregated differences of consecutive
                         chunk bitrates across all traces
```

2 Setup

You implement your ABR algorithm by modifying:

```
cd project2
vim abr.py
```

To test your soulution, run:

```
python3 simulator.py [--video_trace <dir>] [--verbose]
```

The *simulator.py* script will run your solution against multiple network traces, and as a result, it prints the final score. If not specified, video trace is set to the provided default video trace.

3 Submit your solution

After you commit your project, your score on the leader board will be updated.
Task 1 - achieving the score **75** on the given video trace will give you at least 4 points.
Task 2 - achieving the score **90** on the given video trace will give you at least 8 points.
Task 3 - Optimize your solution to reach the top of the leader board.

3.1 Leaderboard ranking

Your solution will be run on 10 hidden video traces and the provided set of bandwidth traces.
To generate a random video trace:

```
python3 generate_random_video.py --seed <seed> --dest <video_dest_directory>
```

To test your solution with the generated video:

```
python3 simulator.py --video_trace <video_dest_directory> [--verbose]
```

3.2 Related literature

Rate Adaptation for Adaptive HTTP Streaming (*SIGCOMM 2011*)

A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service (*SIGCOMM 2014*)

Neural Adaptive Video Streaming with Pensieve (*SIGCOMM 2017*)

4 Important remarks

- In you algorithm, **do not hardcode network traces**. You can make decision based on the bandwidth you have observed, but not based on the bandwidth changes that will come in the future.
- **Do not modify the abr function input parameters**: Your function will be plugged in a different environment, with the same characteristics of the one you have been provided with.
- **Additional packages installation**: If your solution requires some additional python3 packages, please write a **setup.sh**. If you notice that your solution is not running in the leaderboard server (i.e. the leaderboard shows an Error message for your team), please ping the TA to install such packages.
  IMPORTANT: This process might require time, it is indeed discouraged.
- **Leaderboard**:
  - Your solution is re-evaluated each time you push some changes. Please consider batching your commits.

- Your solution is evaluated two times: one with the public video trace, one with 10 hidden video traces.
- The public score refers to the average QoE on the public video trace. The hidden score refers to the average QoE on the hidden video traces. Solutions are ranked according to the latter score.
- In order to keep a reasonable leaderboard refresh rate, your solution cannot take more than 5 minutes to run. Solutions that take more than 5 minutes are forcedly terminated and "Error" is displayed in the leaderboard.
- If your solution encounters some exceptions in execution, the message "Error" is displayed in the leaderboard.
- Your solution is displayed with the team name you provide in the file **team_name.txt**.

5 Final notice

The maximum number of points is *12.5*.
Before the deadline, write a short description (up to 5 sentences) about how your algorithm works and store it in:

`project2/explain.md`

You can find the leader board here.