Project 2: Adaptive bitrate streaming

Future Internet

ETH Zürich - Spring Semester 2019

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: **15 April 2019 at 15:00**.

• The project must be done in groups of two.

• You can receive 12.5 points for this project: achieving threshold A (4pt), achieving threshold B (8pt), optimize your solution - up to 12.5 points.

• The group that achieves the best overall performance will be invited to present its solution in-class.

• After the deadline, you will have an interview. Each group member must be able to individually explain and demo all parts.
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:

\[
\text{score} = \text{agg_video_bitrate} - 4.3 \times \text{rebuffer_time} - \text{agg_switches_amplitude}
\]

\text{agg_video_bitrate} - the sum of all chunk bitrates across all traces
\text{rebuffer_time} - aggregated rebuffering time across all traces
\text{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 solution, run:

```
python simulator.py
```

The `simulator.py` script will run your solution against multiple network traces, and as a result, it prints the final score.
3 Submit your solution

After you successfully execute simulator.py, the log.csv file is created in the logs folder. After you commit this file, your score on the leader board will be updated.

Task 1 - achieving score 5000 will give you at least 4 points.
Task 2 - achieving the score 5500 will give you at least 8 points.
Task 3 - optimize your solution to reach the top of the leader board.

3.1 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 Final notice

The maximum number of points is 12.5.

In your algorithm, do not hardcode network traces. You can make a decision based on the bandwidth you have observed, but not based on the bandwidth changes that will come in the future.

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](#).

* Using `git push --force` is a bad idea. If you use it on your repository, your score on the leader board will not be updated until the contest ends.