

Report for ECE assignment 2

Feng Yutong 120090266

Task1

In this assignment, I implement the algorithm proposed in *Adaptation Algorithm for Adaptive Streaming over HTTP (2012)*

Alorithm Analysis

This paper proposed a streaming client based adaptive bitrate algorithm. It has three goals:

1. Minimize interruptions of playback due to buffer underruns, i.e, reduce rebuffer time.
2. Maximize the minimum and the average video quality.
3. Minimize the number of video quality shifts.
4. Minimize the time after the user requests to view the video and before the start of the playback, which is equivalent to minimizing t_{start} .

In this algorithm, goal 1 has the highest priority. Goal 2 and 3, goal 2 and 4 consititute a trade-off. The main process is to compare the buffer lever with buffer threshold, bitrate and the range of representation (size) of a segment (chunk).

Implementation

Here shows a simplified sudo code to express the logical of the algorithm:

```
Input: segment information
Output: rate_next, Buffer delay time (B_delay)
static runningFastStart := true
B_delay = 0
rate_next = rate_prev
if runningFastStart and some restraint:
    Part 1:
    if Buffer level < B_min
        if rate_plus is within a certain range
            rate_next = rate_plus
    elif Buffer level < B_low
        if rate_plus is within a certain range
            rate_next = rate_plus
    elif rate_plus is within a certain range
        rate_next = rate_plus
    Part 2:
    elif Buffer level > B_high
        B_delay = B_high - samll duration

else:
    runningFastStart := false
    Part 3:
```

```

if Buffer level < B_min
    rate_next = rate_min
elif Buffer level < B_low
    if rate_mins != rate_min and rate_prev > average_throughput_prev
        rate_next = rate_mins
Part 4:
elif Buffer level < B_high
    if rate_plus is within a certain range
        B_delay = max(Buffer level, B_opt)
Part 5:
else
    if rate_prev = rate_max or rate_plus within a certain range
        B_delay = max(Buffer level - small duration, B_opt)
    else
        rate_next = rate_plus

```

Note:

- RuningFastStart is to achieve goal 4. We start with a low bitrate, and then increase rapidly.
- B_delay is minimum buffer level in seconds of playback when the download must be started. The purpose of B_delay is to be able to delay subsequent downloads in order to reduce the buffer level if we already arrived at highest representation or if the available throughput does not allow to select a higher representation.
- The certain level is based on the average segment throughput during the time interval.
- Part 1 is to check whether the conditions are satisfied to increase the bitrate in the beginning.
- Part 2 is when downloading is too fast, we delay to avoid overflow of buffer.
- Part 3 exit the FastStart phase, change bitrate according the buffer.
- Part 4 keeps the buffer level close to $B_{opt} = \frac{B_{low} + B_{high}}{2}$ to achieve a similar sensitivity to both positive and negative spikes.
- Part 5 deals with bitrate at highest representation or if the available throughput does not allow to select a higher representation.

Environment and execution

- Environment

python3

- Execution In terminal 1: `python studentComm.py`

In terminal 2: `python simulator.py inputs/traceHD.txt inputs/manifestHD.json`

Evaluation

```
testHD:
Results:Average bitrate:1733333.3333333333buffer time:0.101switches:5
Score:1136504.892848072

badtest:
Your trace file is poorly formed!Results:Average bitrate:683333.3333333334buffer time:1.8889999999999998switches:4
Score:444328.1271543594

testHDmanPQtrace:
Results:Average bitrate:500000.0buffer time:91.115switches:0
Score:4669.348620686024

testPQ:
Results:Average bitrate:500000.0buffer time:91.115switches:0
Score:4669.348620686024
|

testALThard:
Results:Average bitrate:683333.3333333334buffer time:1.8889999999999998switches:4
Score:444328.1271543594

testALTsoft:
Results:Average bitrate:1733333.3333333333buffer time:0.101switches:5
Score:1136504.892848072
```

This paper has many system parameters, such as time duration, alpha, buffer threshold. Since the value is based on the dataset of the paper with multiple (>10) available bitrate, the provided testcase may not bring the optimal results given those parameters. However, some insights can be seen through the result:

1. Total rebuffer time is much smaller than the given buffer-based algorithm since the highest priority is goal 1, to minimize the rebuffer time rather than get a high quality
2. B_delay is always zero in our cases since the bitrate is chosen relatively low