GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Institut für Informatik

PRONAYA PROSUN DAS

HTTP2 Push-Based Low-Delay Live Streaming With Stream Termination

Based on

H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham and T. C. Thang, "**HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination**," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423-2427, Sept. 2018.
doi: 10.1109/TCSVT.2018.2850740

HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination

Hung T. Le, *Member, IEEE*, Thoa Nguyen, *Member, IEEE*, Nam Pham Ngoc, *Member, IEEE*, Anh T. Pham, *Senior Member, IEEE*, and Truong Cong Thang, *Member, IEEE*

***Abstract*—HTTP adaptive streaming (HAS) has become popular for delivering videos over the Internet. However, most HAS services are currently based on the traditional HTTP/1.1, which has limitations for live streaming. Meanwhile, the new HTTP/2 was standardized in 2015 with many improvements. In this letter, we present a novel HTTP/2-based adaptation scheme for low-delay live streaming over mobile networks. The proposed scheme leverages two HTTP/2 features called *server push* and *stream termination* to enable a shorter camera-to-display delay. Experimental results with small buffer sizes show that our scheme can provide users with high video bitrate and stable buffer level. Furthermore, the buffer size in live streaming can be reduced to two sec without sacrificing video quality.**

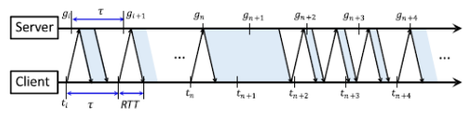


Fig. 1. An example of live streaming over HTTP/1.1. Here, τ is the segment duration and g_n is the time that segment n is generated. Ideally, the request of segment n is sent at time t_n.

effective to use segment durations that are below one second [6].

Introduction

- HTTP adaptive streaming (HAS) has become popular.
    - Multiple version with different bitrates of the same video
    - Chopped into short segments
    - Client is responsible for selecting video bitrate. Based on,

        -connection throughput

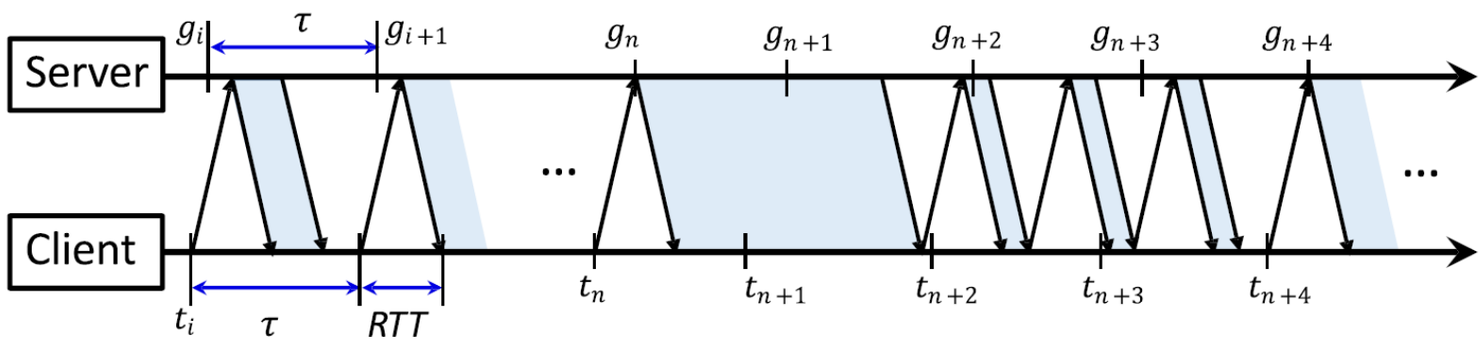
        -client’s buffer level

- HTTP1.1 is used as the delivery protocol.

Limitations of HTTP1.1

- Clients send one request for each video segment.
- Low bandwidth utilization when round-trip time (RTT) increases.
- Change video bitrate after the client has fully received the video segment.
- If download of a segment is too long, the buffer may be depleted.

  Cause video interruptions



Here, τ is the segment duration and g_n is the time that segment n is generated.
The request of segment n is sent at time t_n [ref below].

Image source: H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham and T. C. Thang, "HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423-2427, Sept. 2018.

Limitations of HTTP1.1

- Headers sent with every request

```
Accept:text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding:gzip,deflate,sdch
Accept-Language:en-US,en;q=0.8
Cache-Control:no-cache
Cookie:NTABS=B5; BBC-UID=75620c7ca040deb7c0                                    4dab
%2f5%2e0%20%28Macintosh%3b%20Intel%20Mac%20
%2f537%2e36%20%28KHTML%2c%20like%20Gecko%29                                    fari%
ckns_policy=111; BGUID=55b28cbc20d2e32f221f3ed0e1be9624c960f93b1e483329c3752a6d253e
s1=52CC023F3812005F; BBCCOMMENTSMODULESESSID=L-k22bbkde3jkqf928himljnlkl3; ckpf_ww_
ckpf_mandolin=%22footer-promo%22%3A%7B%22segment%22%3Anull%2C%22end%22%3A%221392834
_chartbeat2=ol0j0uq4hkq6pumu.1389101635322.1392285654268.0111111111111111; _chartbe
ecos.dt=1392285758216
Host:www.bbc.co.uk
Pragma:no-cache
User-Agent:Mozilla/5.0 (Macintosh; Intel Mac OS X 10_9_1) AppleWebKit/537.36 (KHTML
Chrome/32.0.1700.107 Safari/537.36
```

Contain lots of repeated data and can't be compressed

- Allows reusing of connections, but it is serial.
- If one request is slow, other wait.

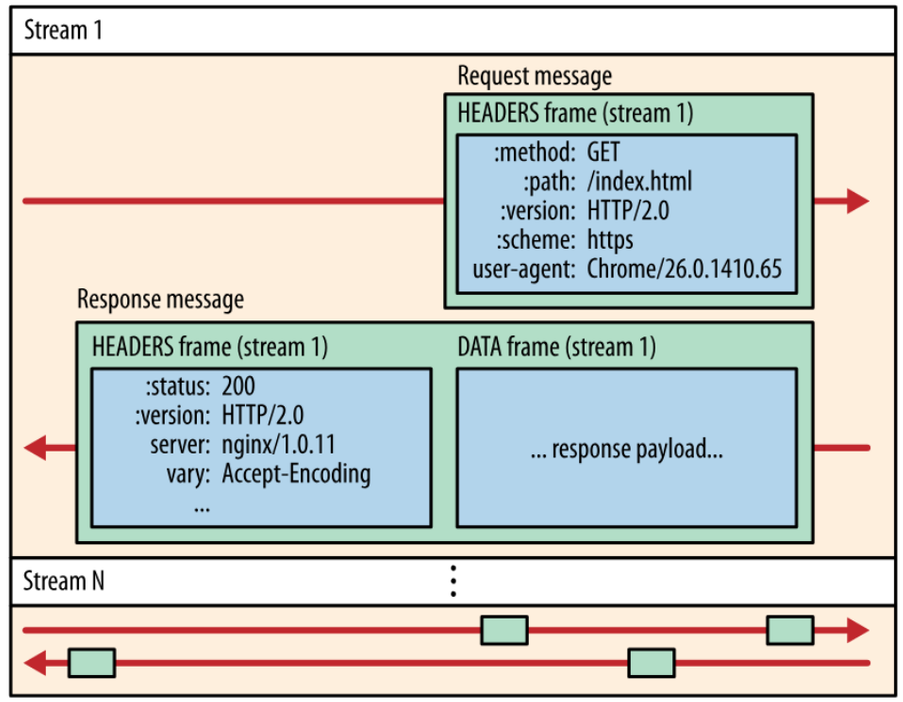
The emergence of HTTP2

- HTTP2 started as SPDY
  - Developed by Google.
  - Published in 2015.

- Currently used by around 15% of all the websites.

- HTTP2 defines streams.
  - Bidirectional sequence of data.
  - One TCP connection can have **multiple streams.**

- The structure inside a stream is called **frame**.
  - Frame types: HEADERS, DATA, SETTINGS, PUSH_PROMISE



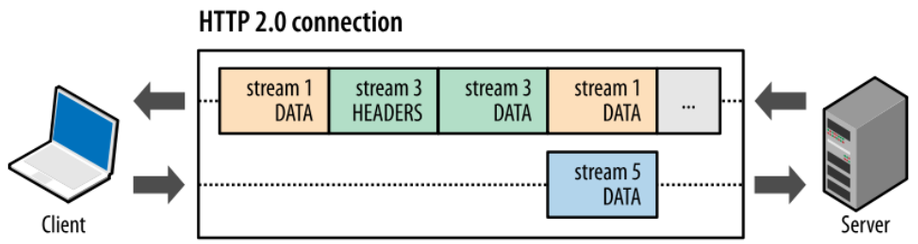


Image Source: https://developers.google.com/web/fundamentals/performance/http2/

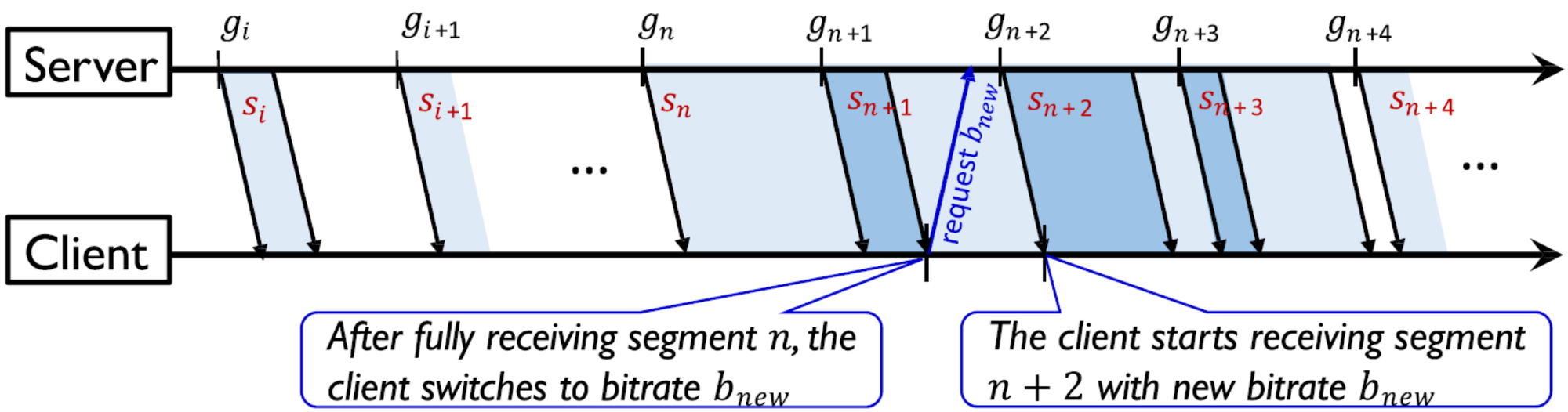
HTTP2 features

- Server push
  - Allows pushing segments without client's request.
  - Eliminate RTT in each segment delivery.
  - To make client aware, server sends the PUSH_PROMISE frame.

- Stream termination
  - Client can terminate a stream.
  - To avoid buffer underflows (for excessively-long segment downloads)
  - terminated by setting END_STREAM flag in the conveyed segment.
  - Or by sending RST_STREAM frame.

- Note
  - HTTP2 stream and frame **are not same** as video stream and frame

HTTP2 features



Here, segment n is generated at time g_n and is then delivered to the client via stream S_n. The server can push concurrently 0, 1, or 2 streams (indicated in white, light blue, or dark blue, respectively). Note that segment $n+1$ is pushed with the old video bitrate [ref below] .

Image source: H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham and T. C. Thang, "HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423-2427, Sept. 2018.

Live streaming and adaptation

- Server push segments that are generated in real time.

- Throughput-based method

  - Effective to small buffer size.

- Buffer-based method

  - Need bigger buffer size.

- Probabilistic method

  - Requires observing segment downloads over a time interval.
  - Not practical.

- Push-K method

  - K segments are pushed per request

- Full push method

  - Server pushes new segments periodically.
  - Bitrates are changed by sending bitrate change request (by client)

- Problems

  - For smaller buffer size video interruptions are unavoidable (buffer underflow).
  - Client can only make decision after receiving a segment.

Proposed Scheme

- Server side

  -Video is provided with number of bitrate version {b}

  -Video is chopped into segment of τ seconds.

  -So a new segment is generated every τ seconds.

  -Server initiate new push stream by sending PUSH_PROMISE frame.

  -Server push segments one-by-one to client in server-initiated stream S_n.

- Client side

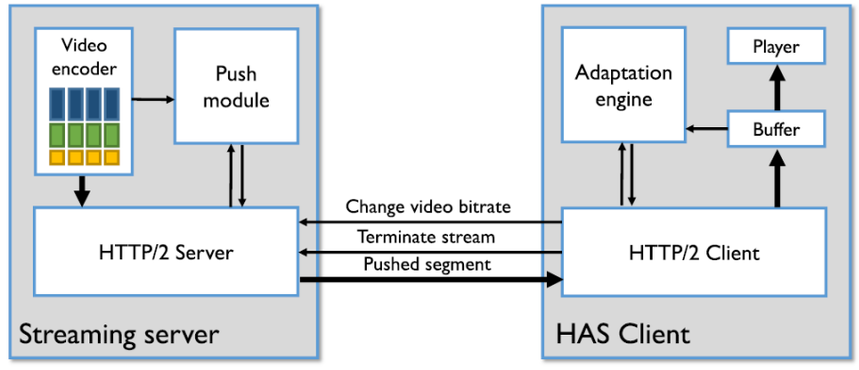
  -Initially client load L segments into its buffer β.

  -Target buffer level, $\beta_{tar} = L \times \tau$ …eq(1)

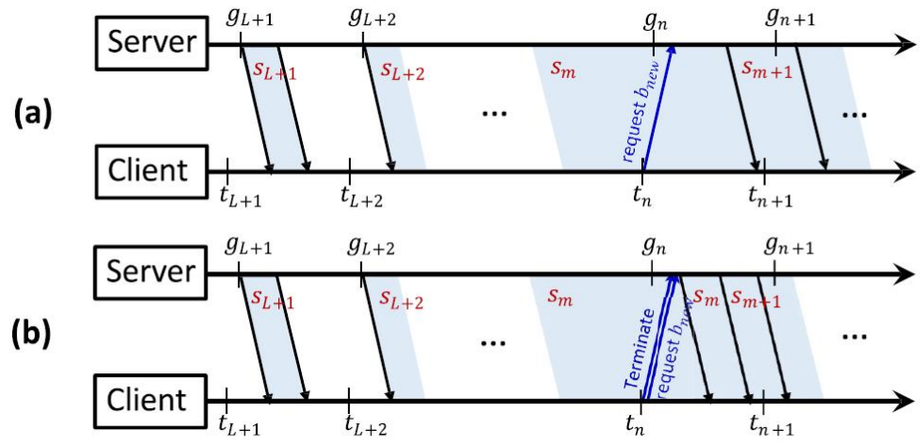
  -β_{tar} seconds for buffering.

  -Client control the video quality (bitrate).

  - Can terminate the stream



Block diagram of our system model [ref below].



Decision time instant t_n. Segment n is generated at time g_n and is then delivered to the client via stream S_n. In this example, the client is receiving segment m ($m < n$) at time t_n [ref below].

Image source: H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham and T. C. Thang, "HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423-2427, Sept. 2018.

Proposed Scheme

- Adaptation timing:

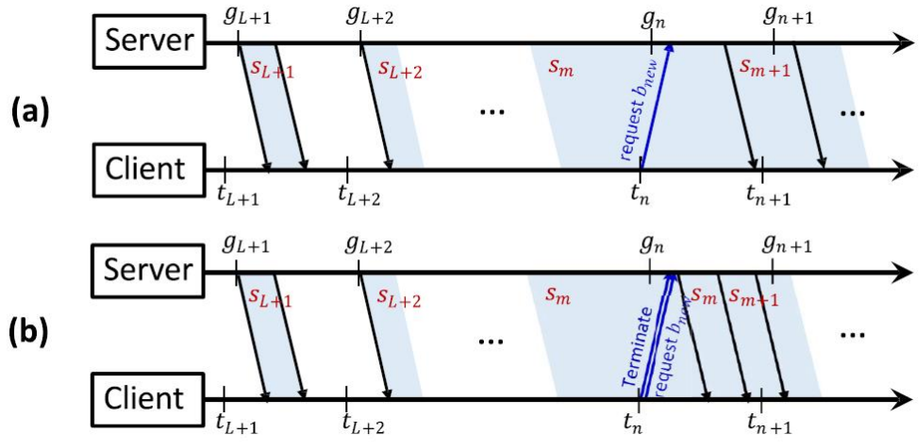
Decision time: $t_n = \theta + (n - L - 1) \times \tau$, ...eq(2)

where, τ is segment duration,

θ : time (steady state starts),

n : total segments,

L : loaded segments to the buffer.



Decision time instant t_n. Segment n is generated at time g_n and is then delivered to the client via stream S_n. In this example, the client is receiving segment m ($m < n$) at time t_n [ref below].

Suppose, m (m < n) segments are received in stream S_m with bitrate b_{cur},
b_{new} : *new bitrate decided at time* t_n.

Client:

a) Request video with b_{new}, applied after server receives the new request,

b) Terminate the stream S_m and request with lower bitrate.

Image source: H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham and T. C. Thang, "HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423-2427, Sept. 2018.

Proposed Scheme

- Adaptation decision:

  i) Estimate the future buffer level for (a) and (b),

  ii) Select the option that provide higher buffer level.

  For case (a)

  current bitrate, $\boldsymbol{b_{opt1} = \max\{b \mid b < T\}}$, …eq(3)

  where, **b**: bitrates and **T**: throughput.

  future buffer level, $\boldsymbol{\beta_{opt1} = \beta_{cur} - {}^{d}/_{T}}$, …eq(4)

  where, $\boldsymbol{\beta_{cur}}$: current buffer level, **d**: remaining data of current segment **m.**

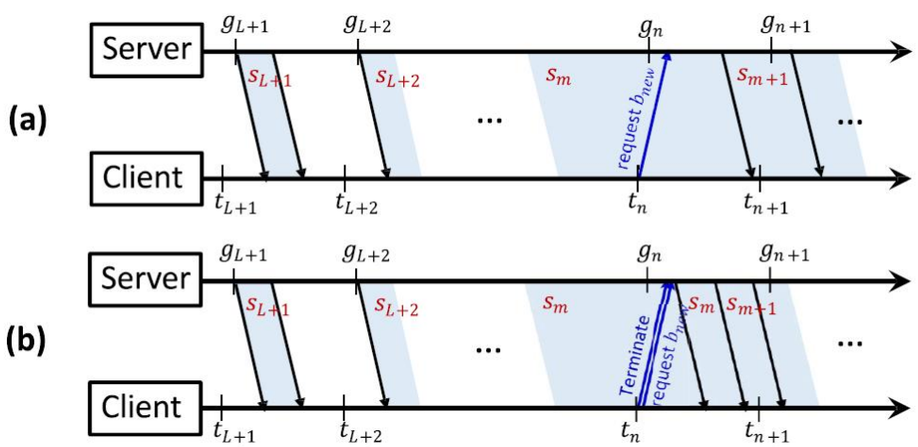
For case (b)

current buffer level, $\boldsymbol{\beta_{opt2} = f(b) = \beta_{cur} - RTT - \frac{b \times \tau}{T}}$, …eq(5)

where, **b**: bitrates, **T**: throughput, $\boldsymbol{\tau}$: segment duration, $\boldsymbol{\beta_{cur}}$: current buffer level, $\boldsymbol{\beta_{opt2}}$: actually the function of $\boldsymbol{f(b)}$.

current bitrate, $\boldsymbol{b_{opt2} = \max\{b \mid (f(b) > \beta_{min}) \wedge (b < T)\}}$, …eq(6)

where, **b**: bitrates, **T**: throughput and $\boldsymbol{\beta_{min}}$: buffer threshold.



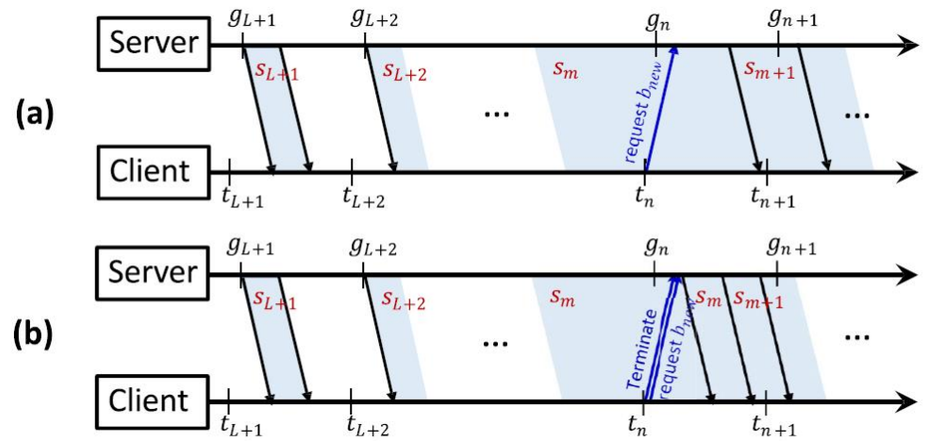
Decision time instant t_n. Segment n is generated at time g_n and is then delivered to the client via stream S_n. In this example, the client is receiving segment m ($m < n$) at time t_n [ref below].

Image source: H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham and T. C. Thang, "HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423-2427, Sept. 2018.

Proposed Scheme

- After computing $\boldsymbol{\beta_{opt1}}$ and $\boldsymbol{\beta_{opt2}}$:

$$\boldsymbol{\beta_{new}} = \begin{cases} \boldsymbol{b_{opt1}} & \boldsymbol{if\ \beta_{opt1} \geq \beta_{opt2}} \\ \boldsymbol{b_{opt2}} & \boldsymbol{if\ \beta_{opt1} < \beta_{opt2}} \end{cases} \quad \text{...eq(7)}$$



Decision time instant t_n. Segment n is generated at time g_n and is then delivered to the client via stream S_n. In this example, the client is receiving segment m ($m < n$) at time t_n [ref below].

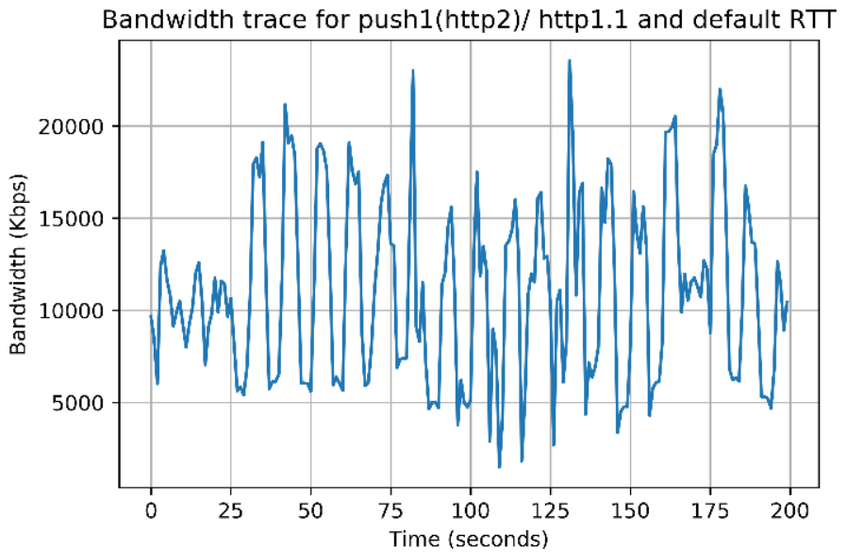
Image source: H. T. Le, T. Nguyen, N. P. Ngoc, A. T. Pham and T. C. Thang, "HTTP/2 Push-Based Low-Delay Live Streaming Over Mobile Networks With Stream Termination," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 9, pp. 2423-2427, Sept. 2018.

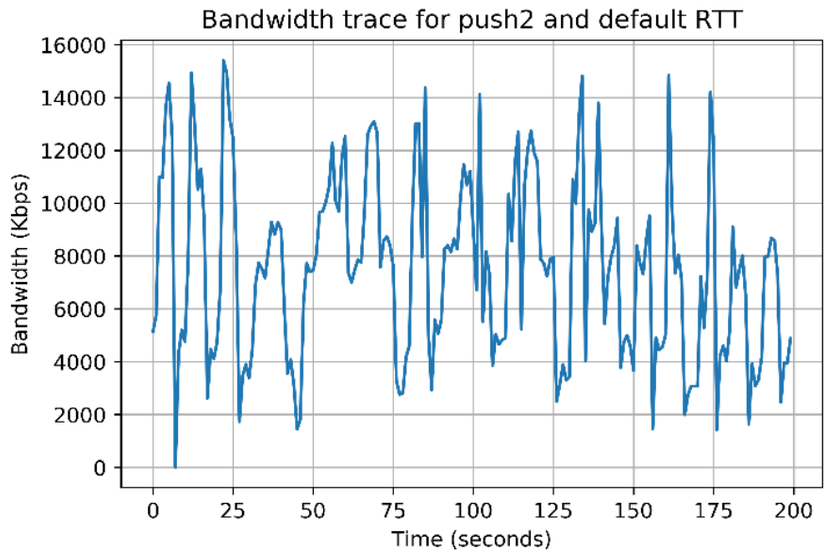
Implementation

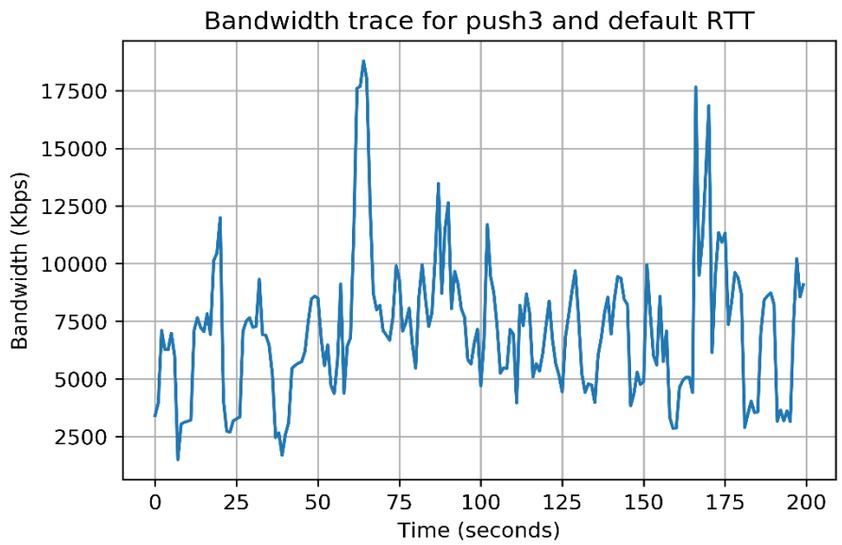
- System: Windows 10
- Language: Python 3.6
- HTTP2 Library: Hyper H2

- Available video bitrate {b} = {50, 150, 300, 600, 1200, 2500, 4000}
- Used method: push 1, 2, 3 and push based method with stream termination (PAT).

Results (bandwidth trace)

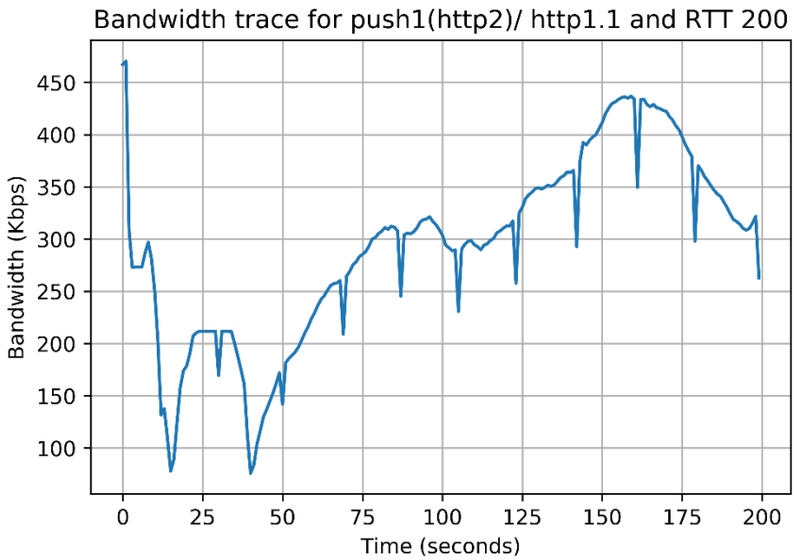


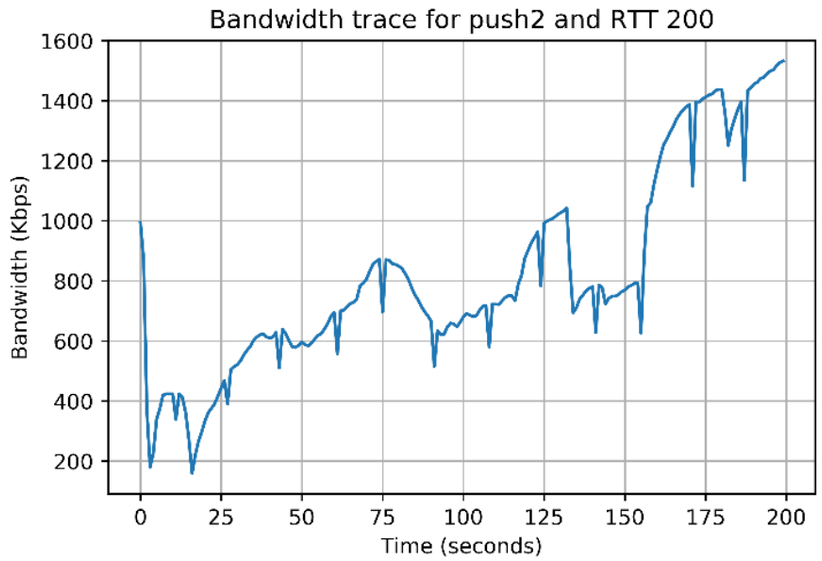


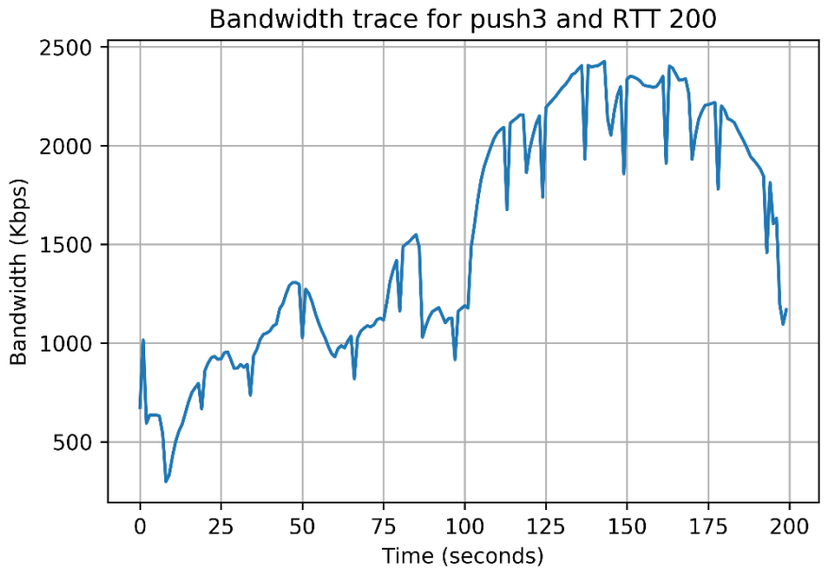


Bandwidth trace examples used in our experiments.

Results (bandwidth trace)

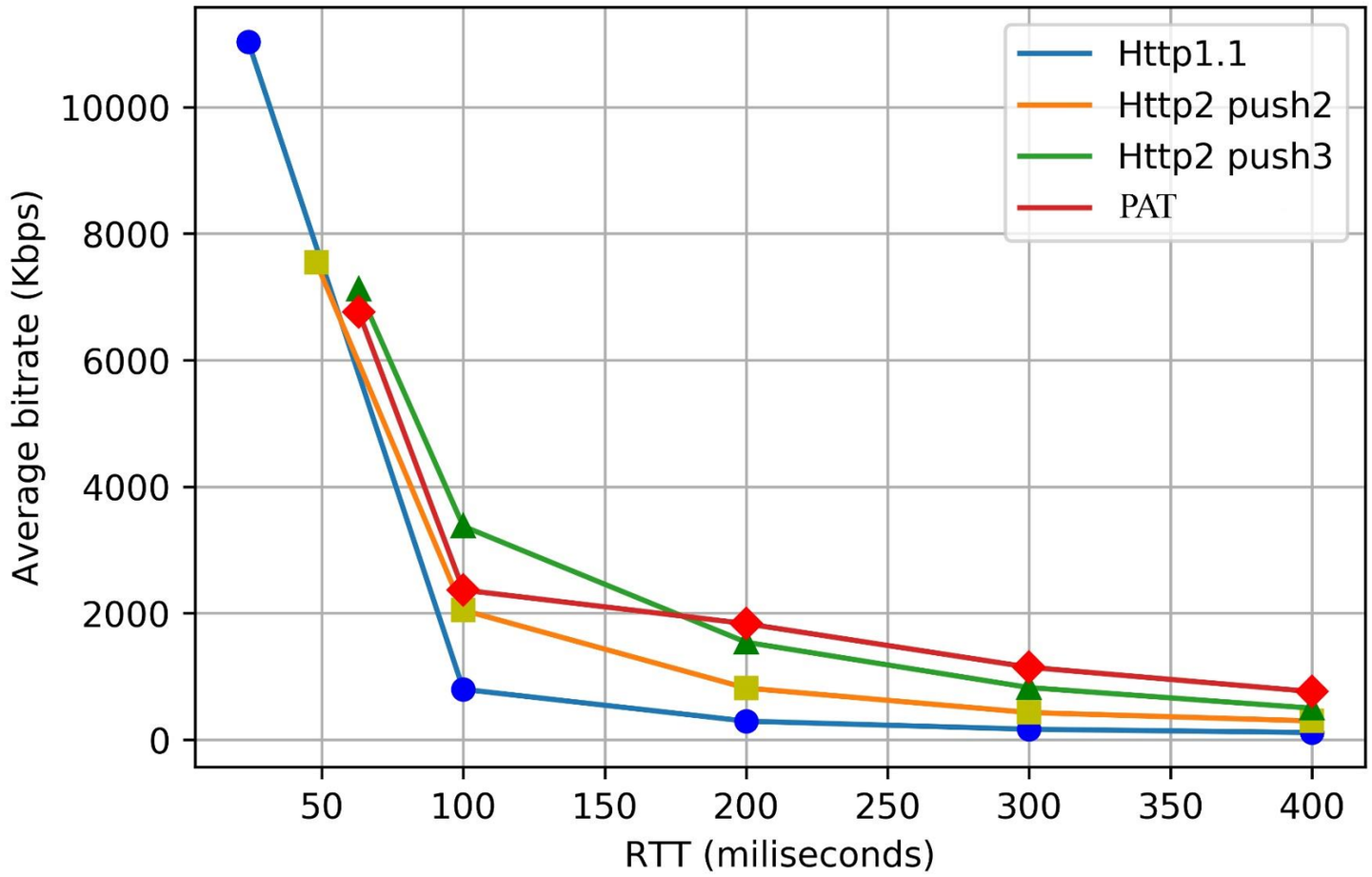






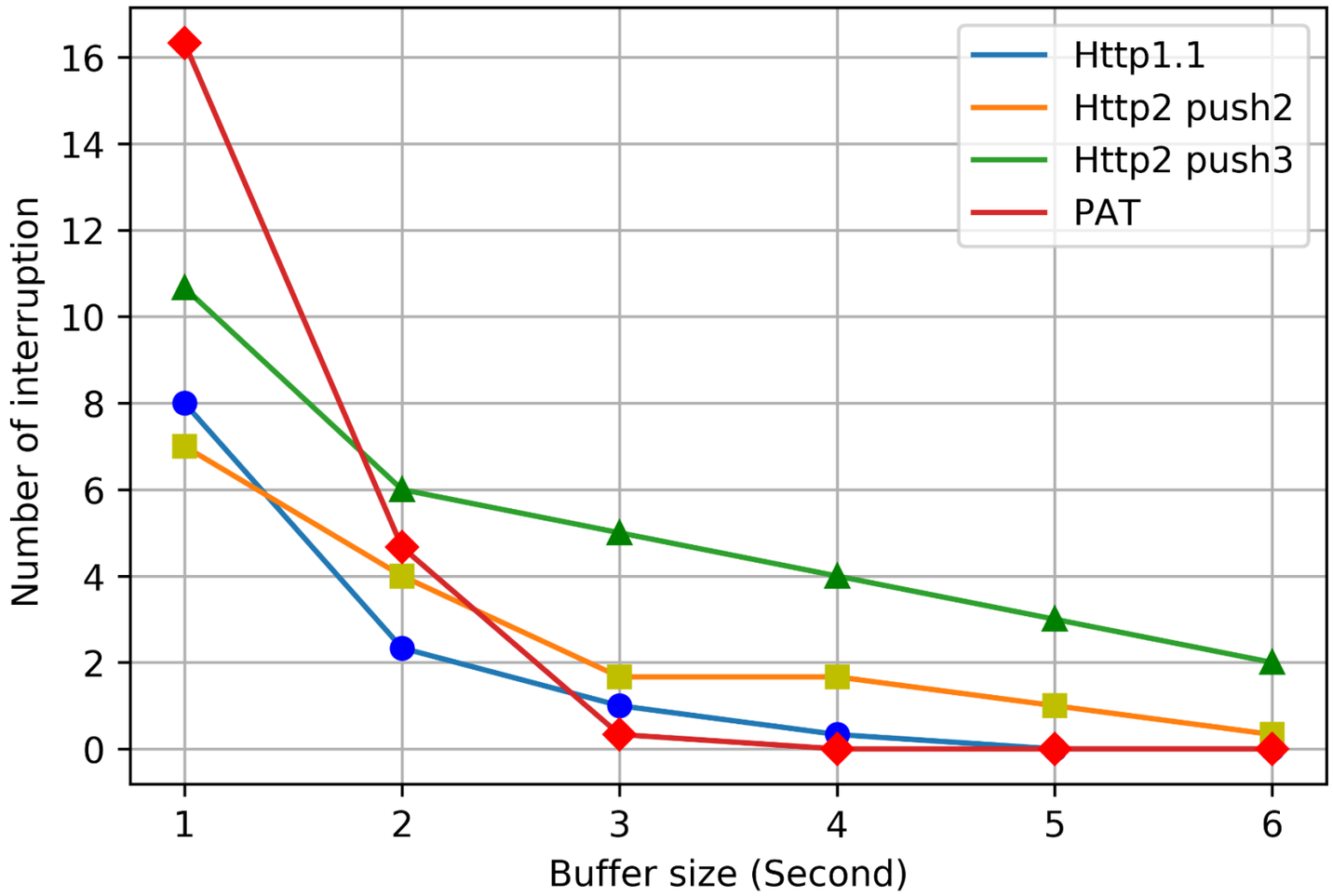
Bandwidth trace examples used in our experiments.

Results (average video bitrate)



Average video bitrate with a large buffer size of 6s.

Results (interruptions)



Impact of the buffer size on the number of interruptions.

Conclusion

- This work leverages server push and stream termination for the first time.
- Smallest buffer size supported by this scheme is 2 seconds.
- In future, video quality will be improved by using other HTTP2 features like,
  - Flow control
  - Stream prioritization

Thank you for listening…

Questions?