NADA Implementation in Mozilla Browser and Draft Update

draft-ietf-rmcat-nada-11

Xiaoqing Zhu, Rong Pan, Michael A. Ramalho, Sergio Mena,

Paul E. Jones, Jiantao Fu, and Stefano D'Aronco

July 2019 | IETF 105 | Montreal, Canada

Draft Update per AD Review

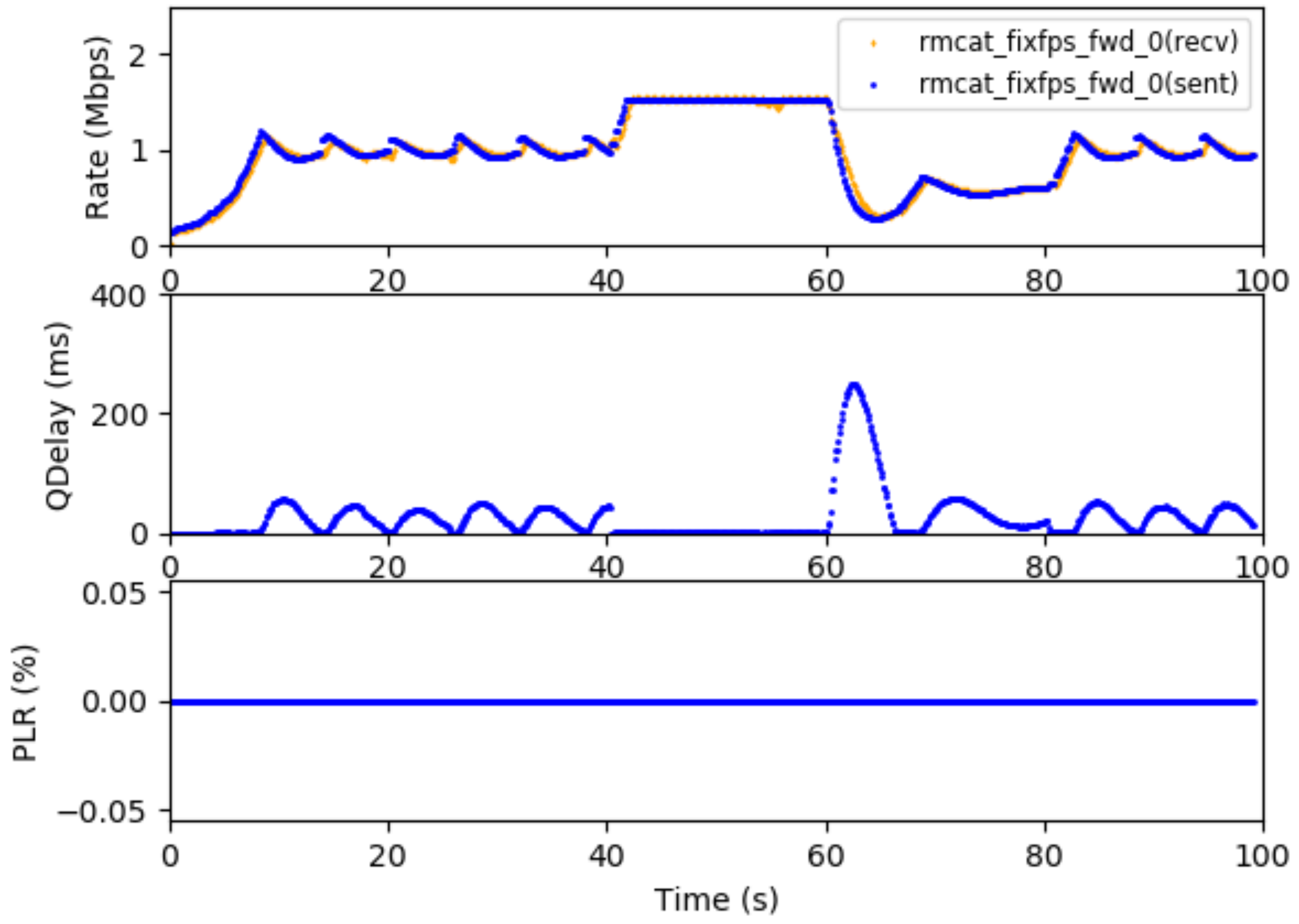
- Section 5.2.2: further clarification on how rate shaping buffer size influences the adjustment of target video rates (r_vin) and sending rate (r_send):
  - Added discussion to show that typical rate changes introduced by the rate shaping buffer is on the order of 10s of Kbps: e.g., rate shaping buffer length of 2000 Bytes leads to a rate change of 48Kbps for the recommended parameters BETA_S =0.1 and BETA_V=0.1;
  - Added bounding conditions such that the rate change introduced by rate shaping buffer is less than 5% of the congestion control reference rate
  - Further limited final sending rate to below R_max, and final video target rate to above R_min.
  - Noted that this feature is optional and as a compliments the congestion control module
- Section 10: added text on Security Considerations
- Incorporated additional editorial comments from AD review

Section 10. Security Considerations

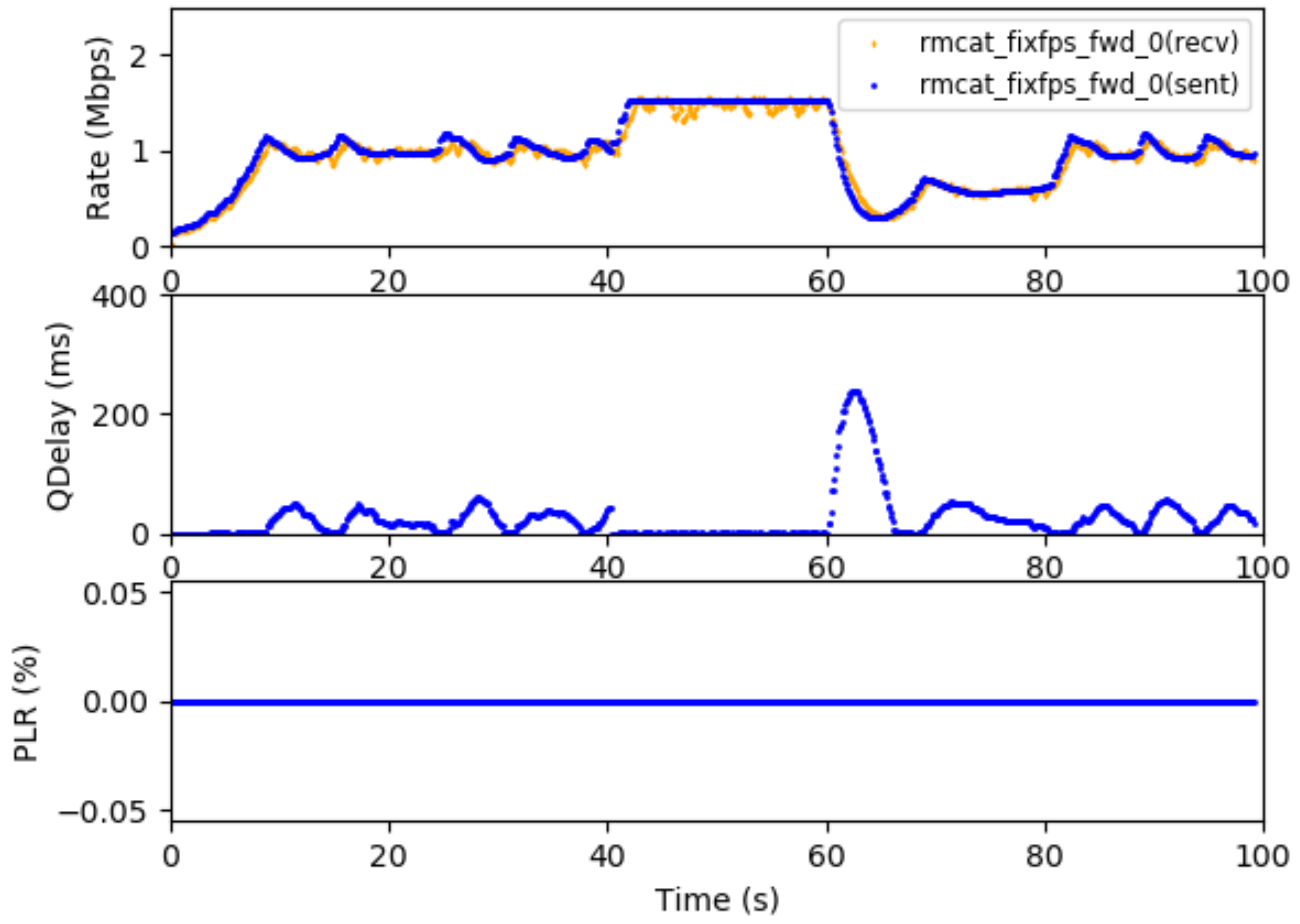
The rate adaptation mechanism in NADA relies on feedback from the receiver. As such, it is vulnerable to attacks where feedback messages are hijacked, replaces, or intentionally injected with misleading information, similar to those that can affect TCP. It is therefore RECOMMENDED that the RTCP feedback message is at least integrity checked. The modification of sending rate based on send-side rate shaping buffer may lead to temporary excessive congestion over the network in the presence of a unresponsive video encoder. However, this effect can be mitigated by limiting the amount of rate modification introduced by the rate shaping buffer, bounding the size of the rate shaping buffer at the sender, and maintaining a maximum allowed sending rate by NADA.

Influence of Rate Shaping Buffer: TC5.1

BETA_S = 0.0 I BETA_V = 0.0

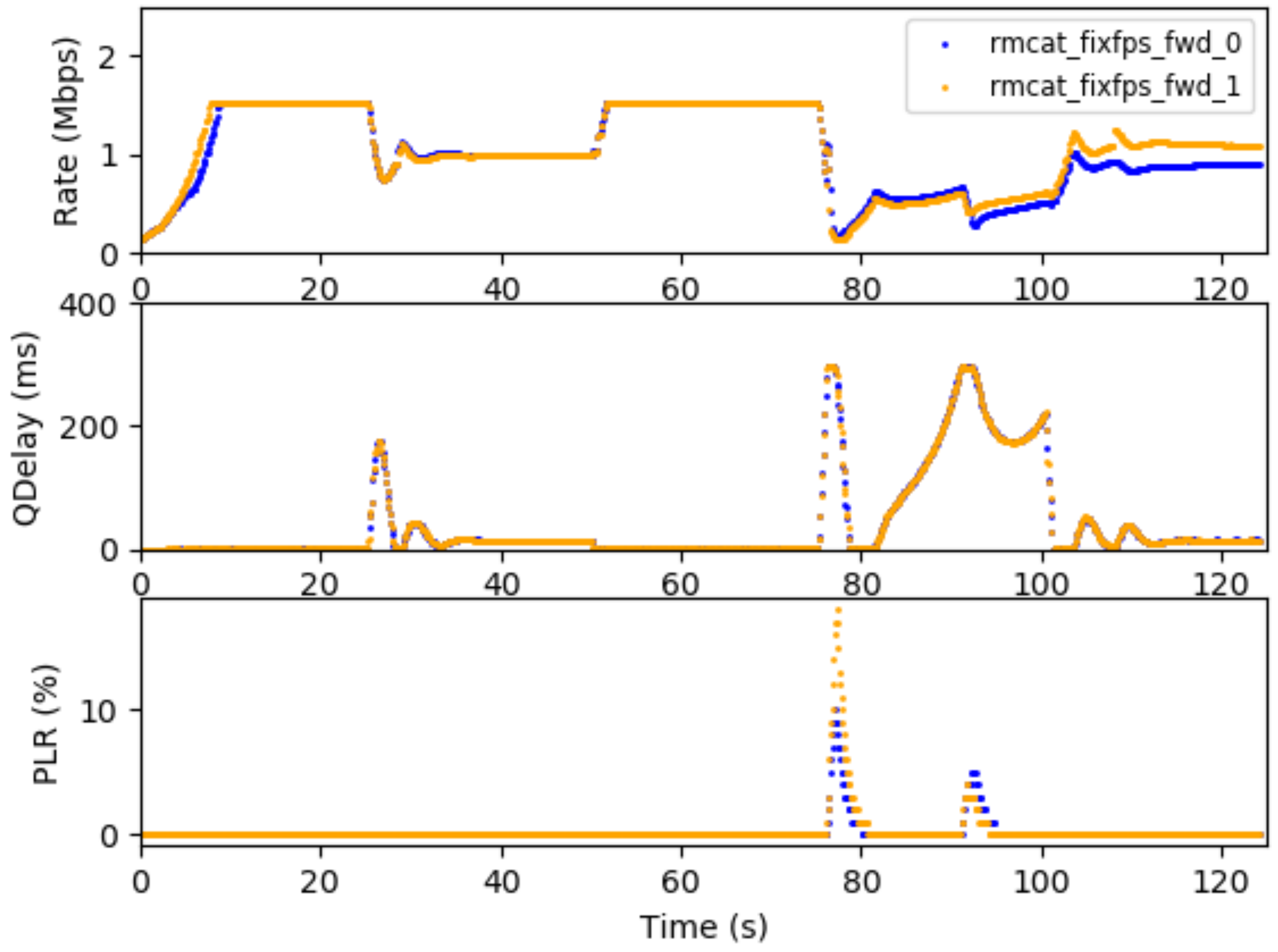


BETA_S = 0.1 I BETA_V = 0.1

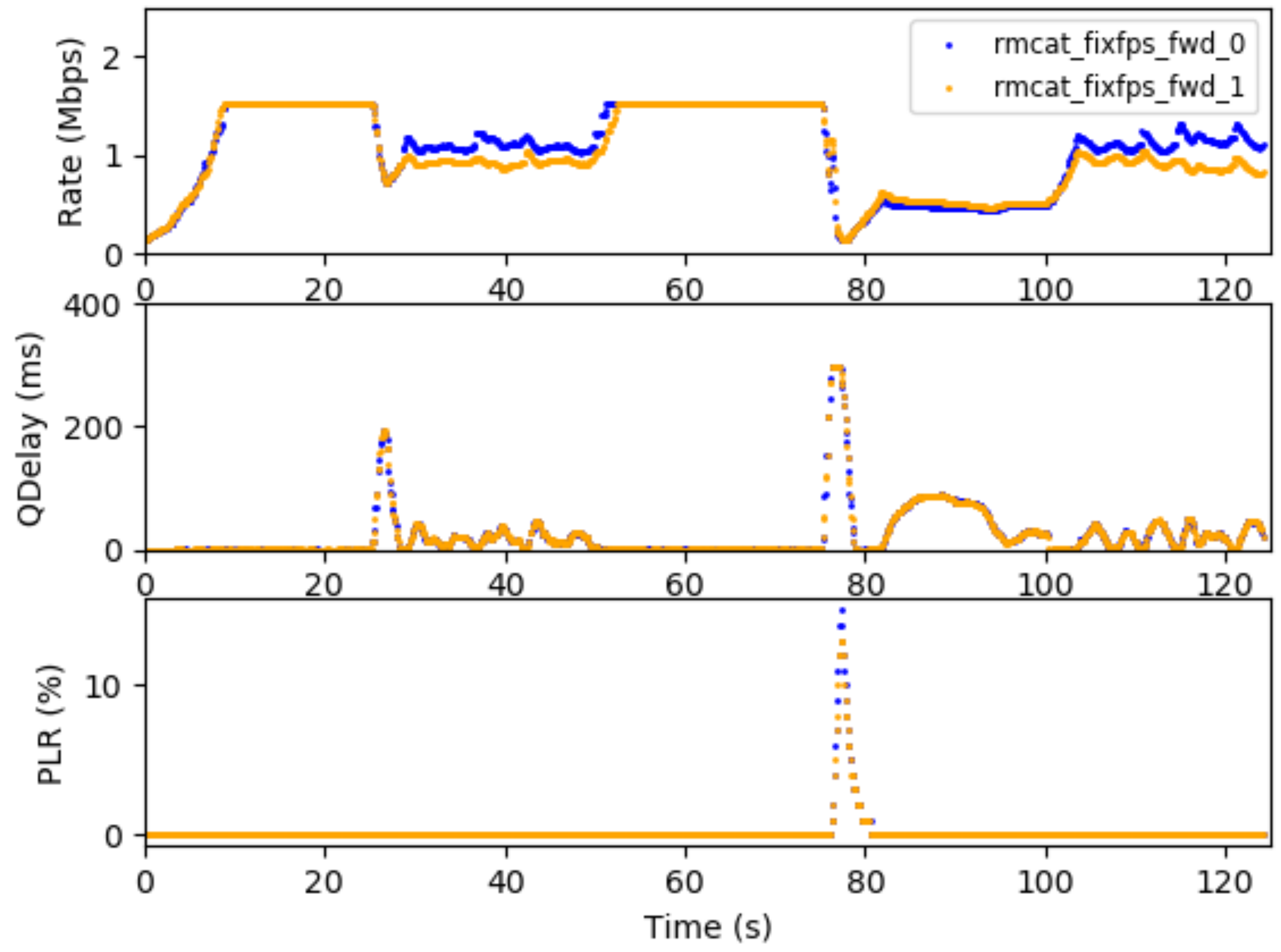


Influence of Rate Shaping Buffer: TC5.2

BETA_S = 0.0 I BETA_V = 0.0

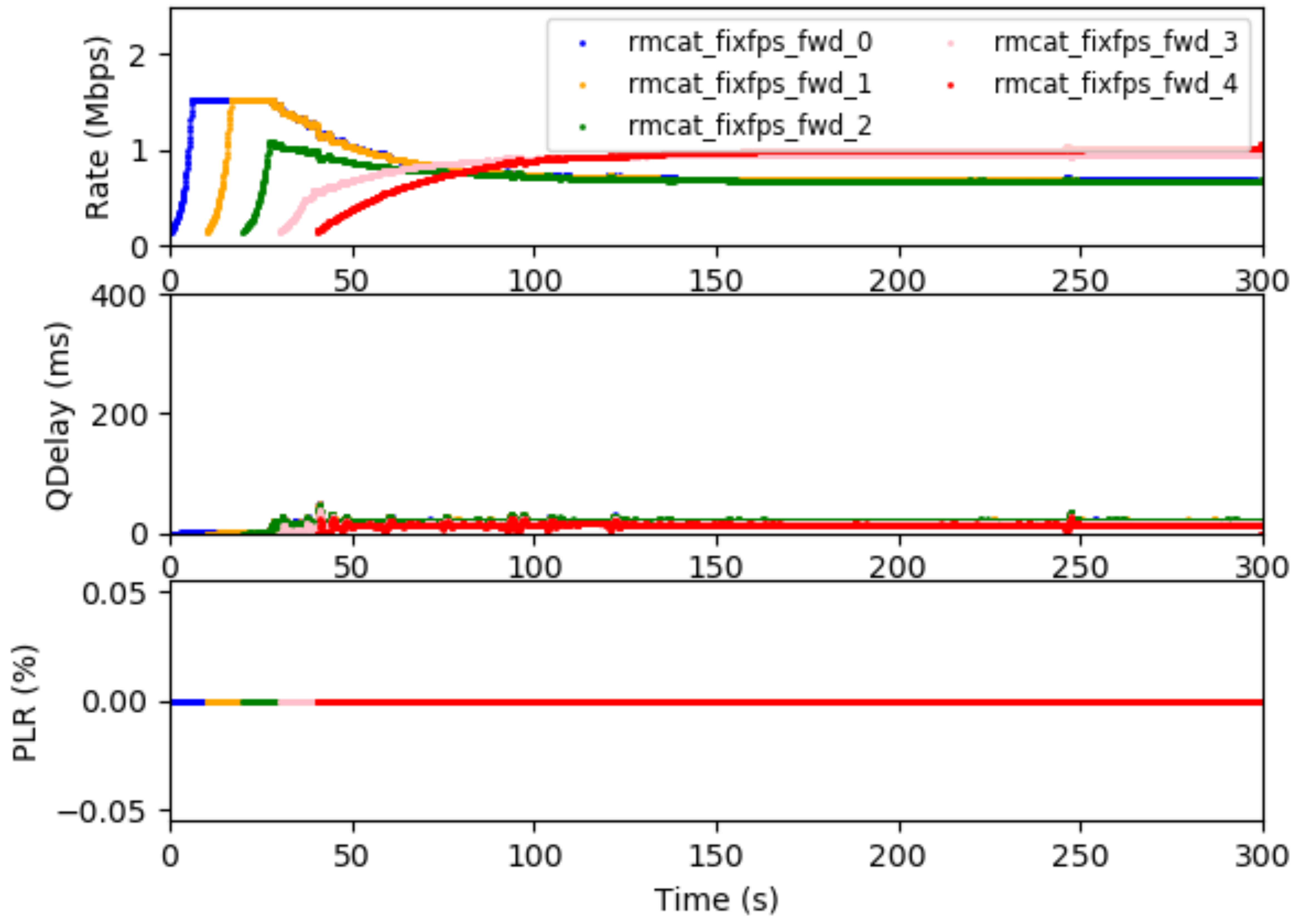


BETA_S = 0.1 I BETA_V = 0.1

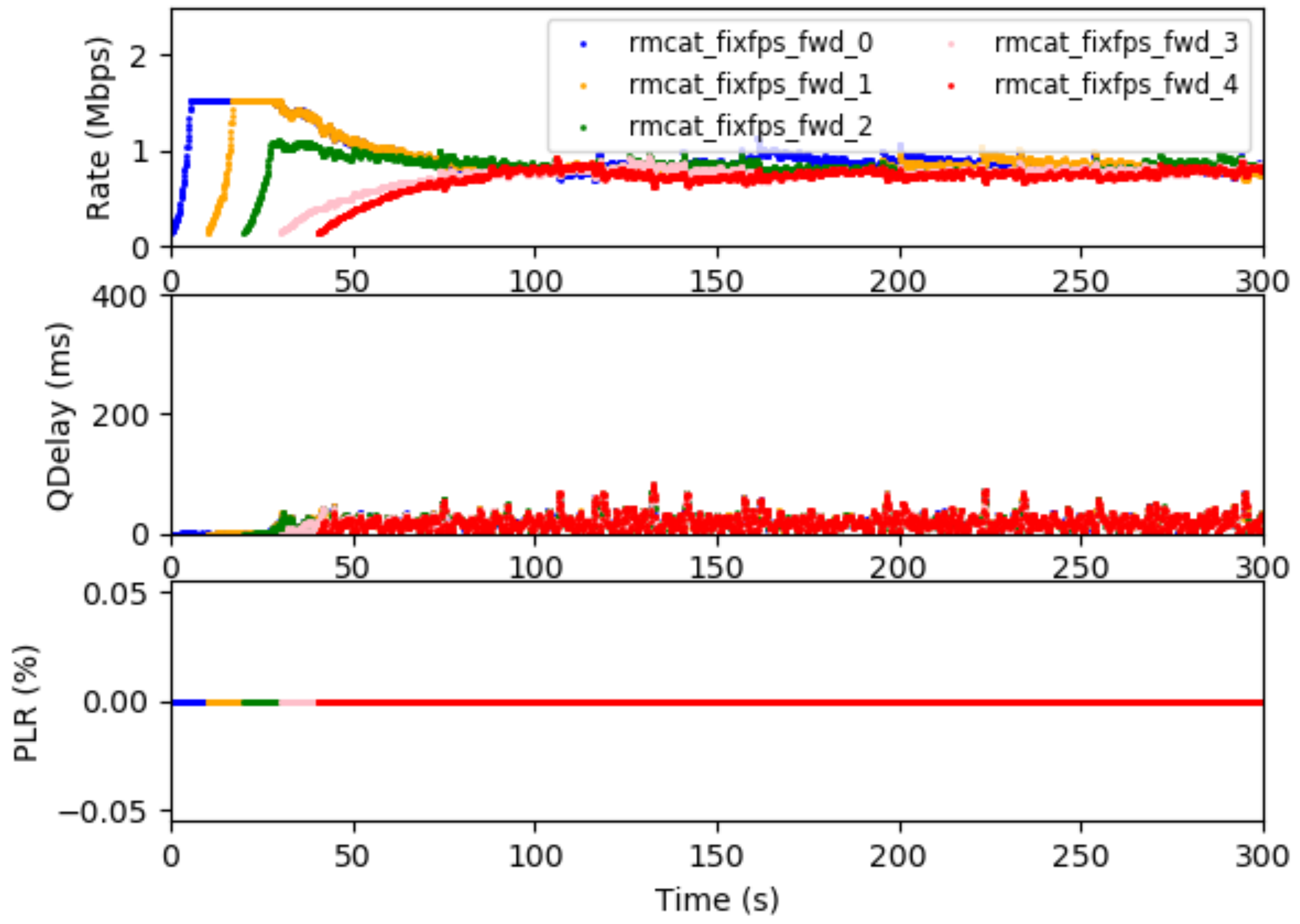


Influence of Rate Shaping Buffer: TC5.5

BETA_S = 0.0 I BETA_V = 0.0

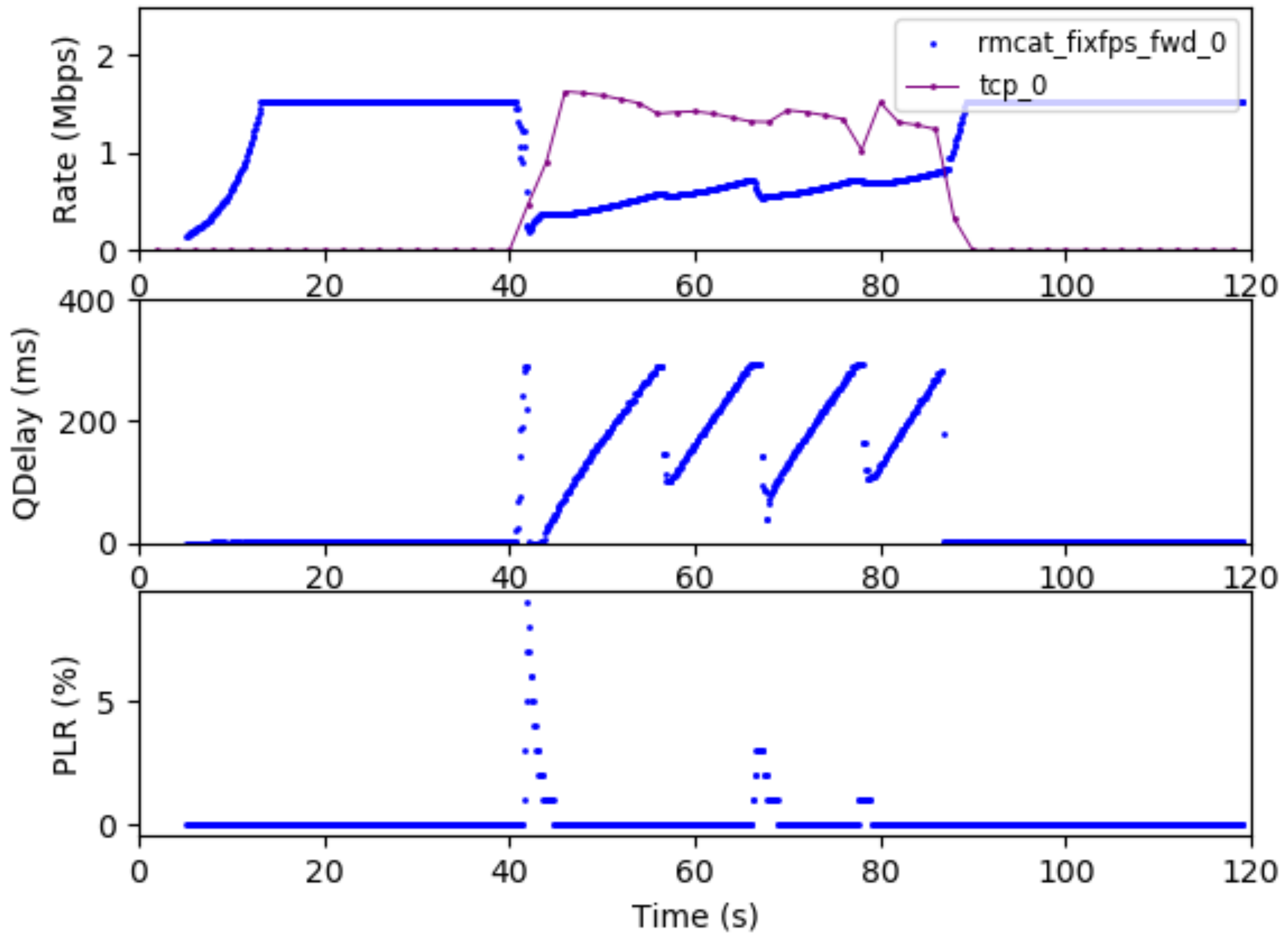


BETA_S = 0.1 I BETA_V = 0.1

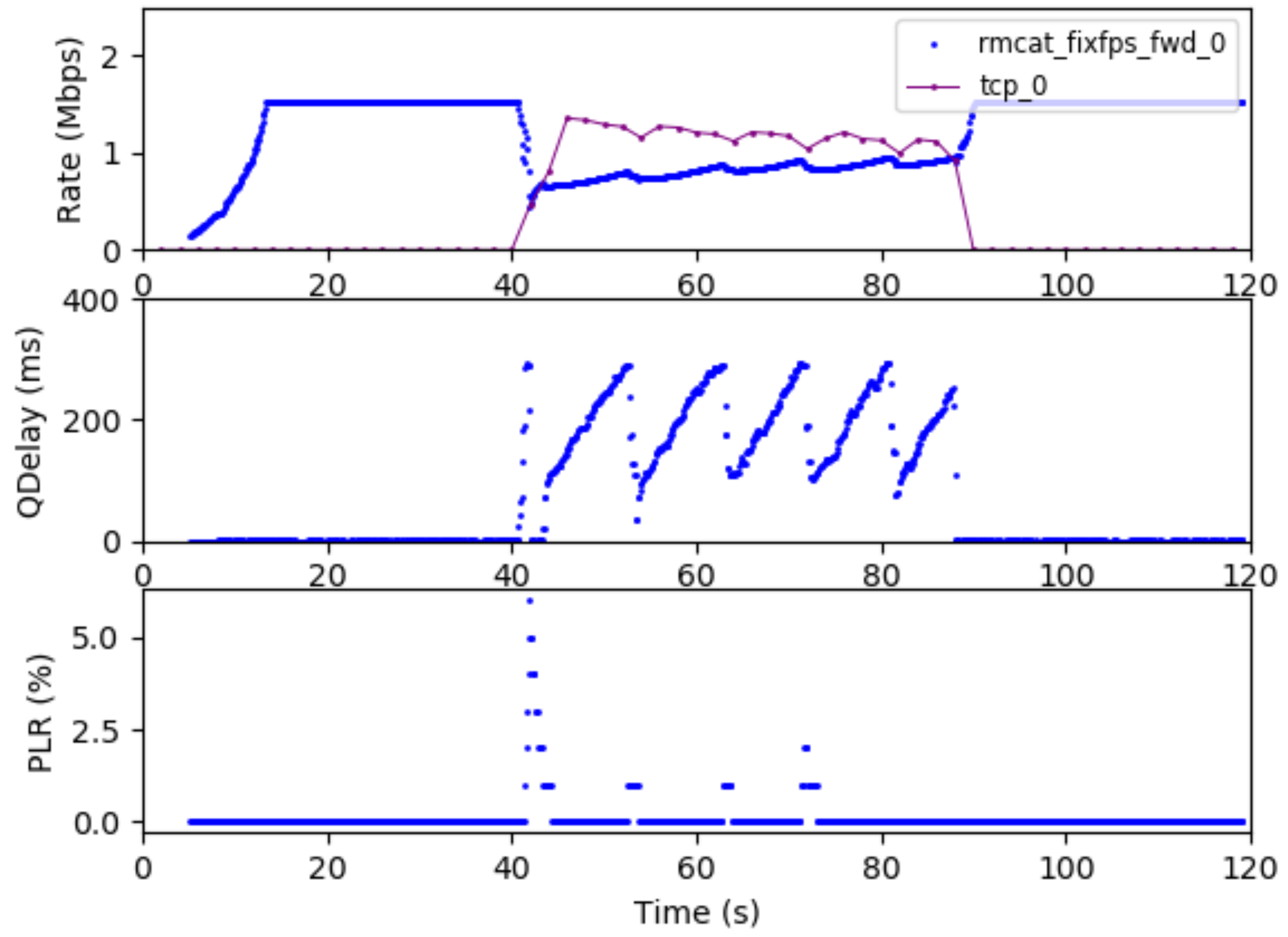


Influence of Rate Shaping Buffer: TC5.6

BETA_S = 0.0 I BETA_V = 0.0



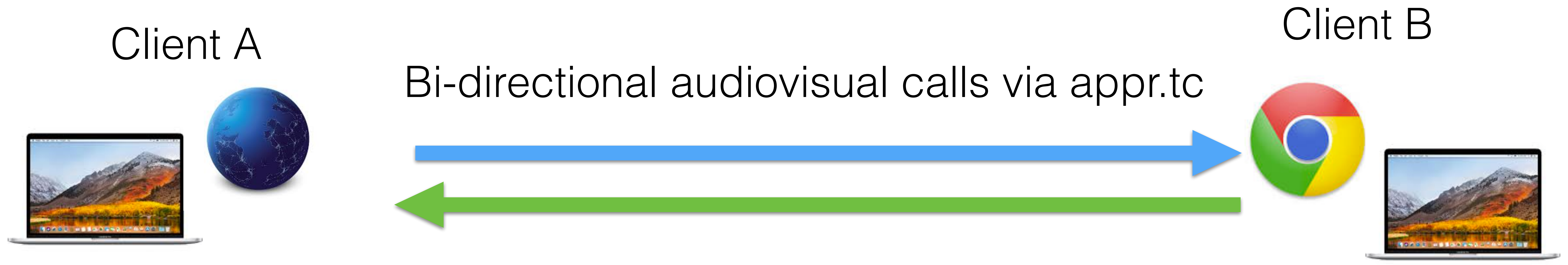
BETA_S = 0.1 I BETA_V = 0.1



Updated NADA Implementation in Mozilla

- Rebased modifications to mozilla-central nightly branch from May 29, 2019:
  - Incorporates a more recent version of WebRTC in Mozilla
  - Publicly accessible dev branch on GitHub: https://github.com/sergio-mena/gecko-dev/commits/nada
- Summary of code changes:
  - Replaces delay_based_bwe.cc/h with nada_owd_bwe.cc/h in the congestion_controller module within WebRTC
  - NADA-based bandwidth estimation using relative one-way delay (OWD) as congestion signal
  - Interoperates with unmodified receivers by leverages per-packet info via transport_cc feedback message as defined in http://www.ietf.org/id/draft-holmer-rmcat-transport-wide-cc-extensions-01
  - Exports statistics using existing WebRTC logging framework

Test Setup for Browser-based Calls



Firefox Nightly w/ modifications

- Bandwidth adaptation via NADA
- Maximum sending rate: 3Mbps
- Default resolution: 720p (limits sending rate to 2.5Mbps)
- Stats of outgoing flow reported by the NADA module

Unmodified Chrome browser

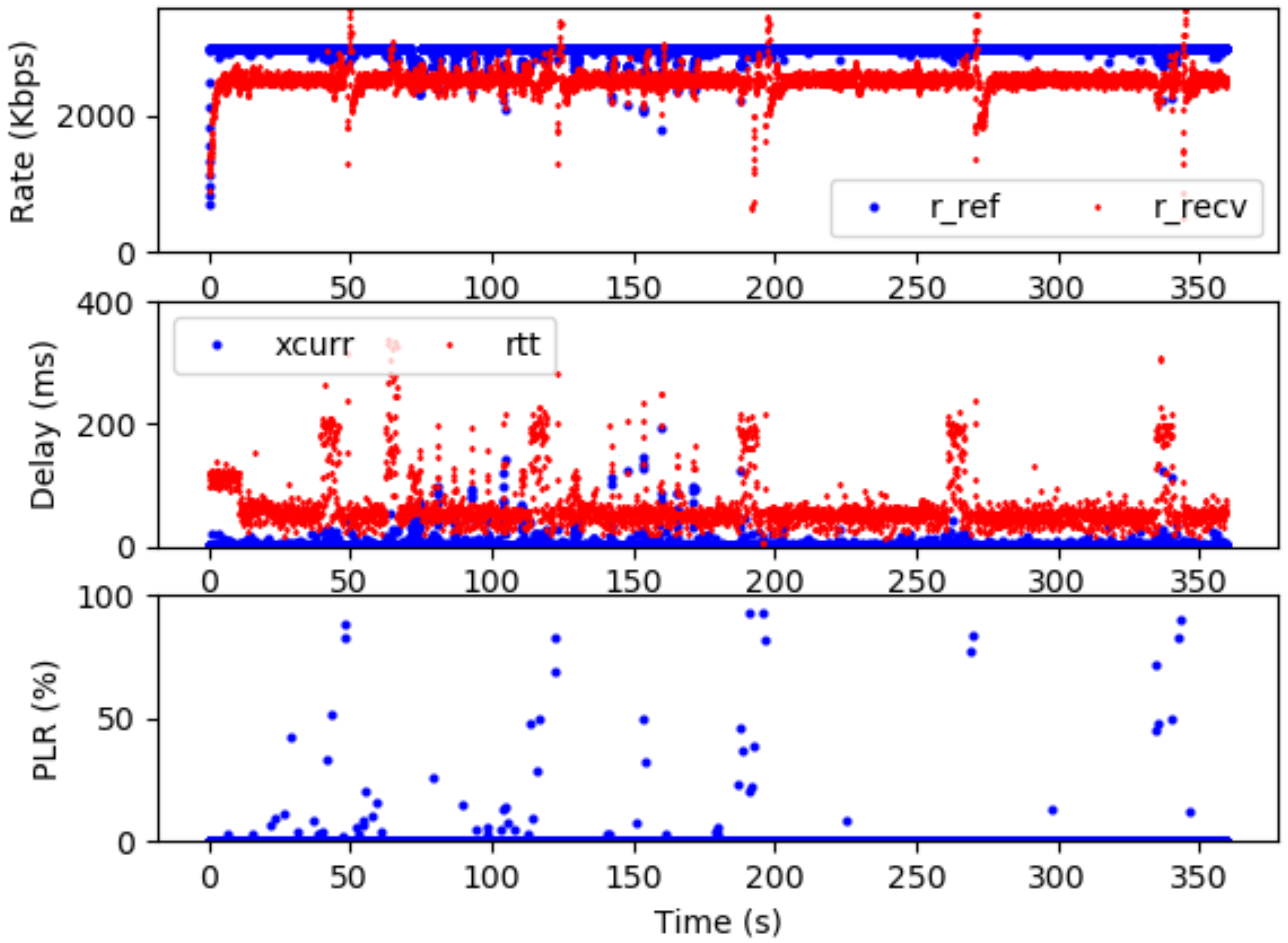
- Default bandwidth adaptation
- Stats of incoming/outgoing flows displayed on the *webrtc-internals* panel in browser

Scenario A: Local Session over Home Wi-Fi

- Client A:
  - Location: Austin, Texas
  - Connection: home Wi-Fi

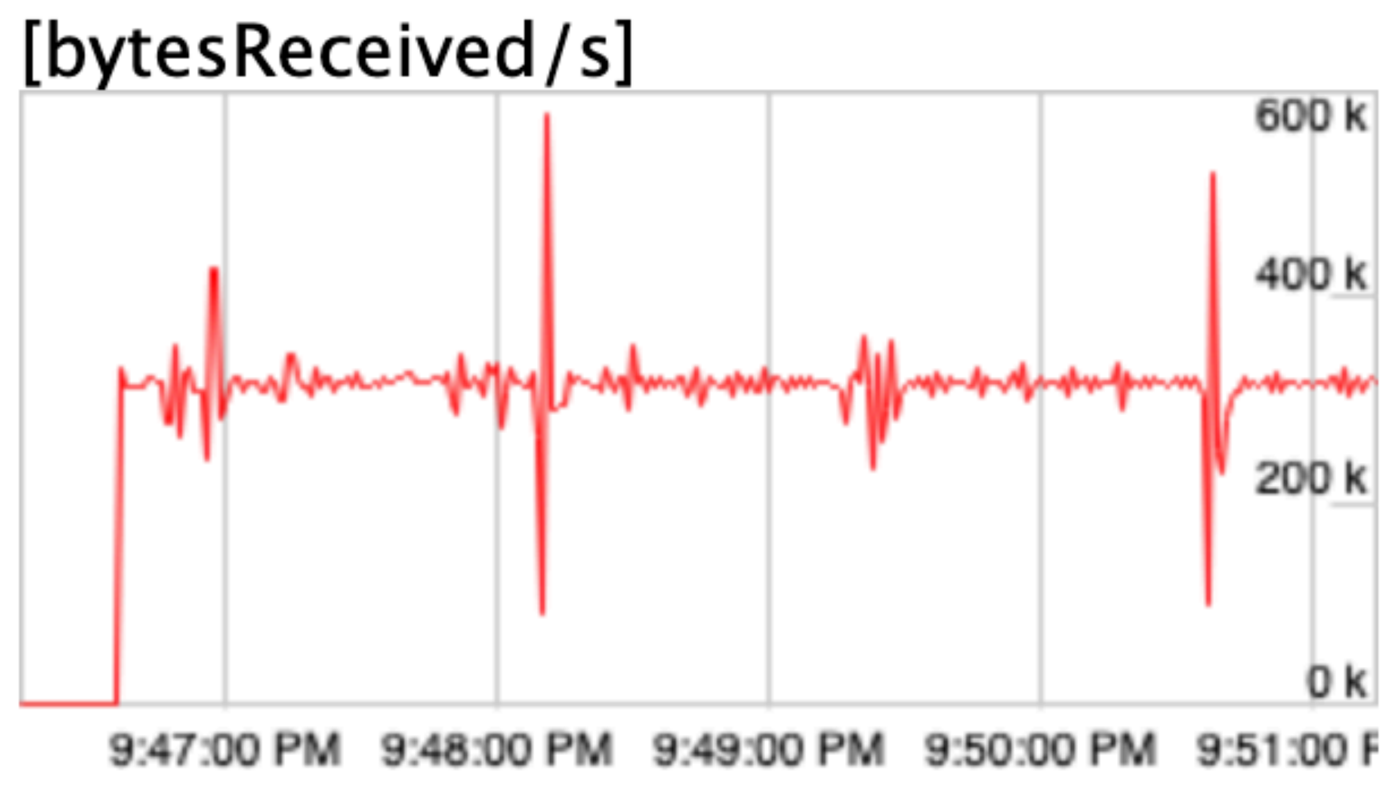
- Client B:
  - Location: Austin, Texas
  - Connection: home Wi-Fi

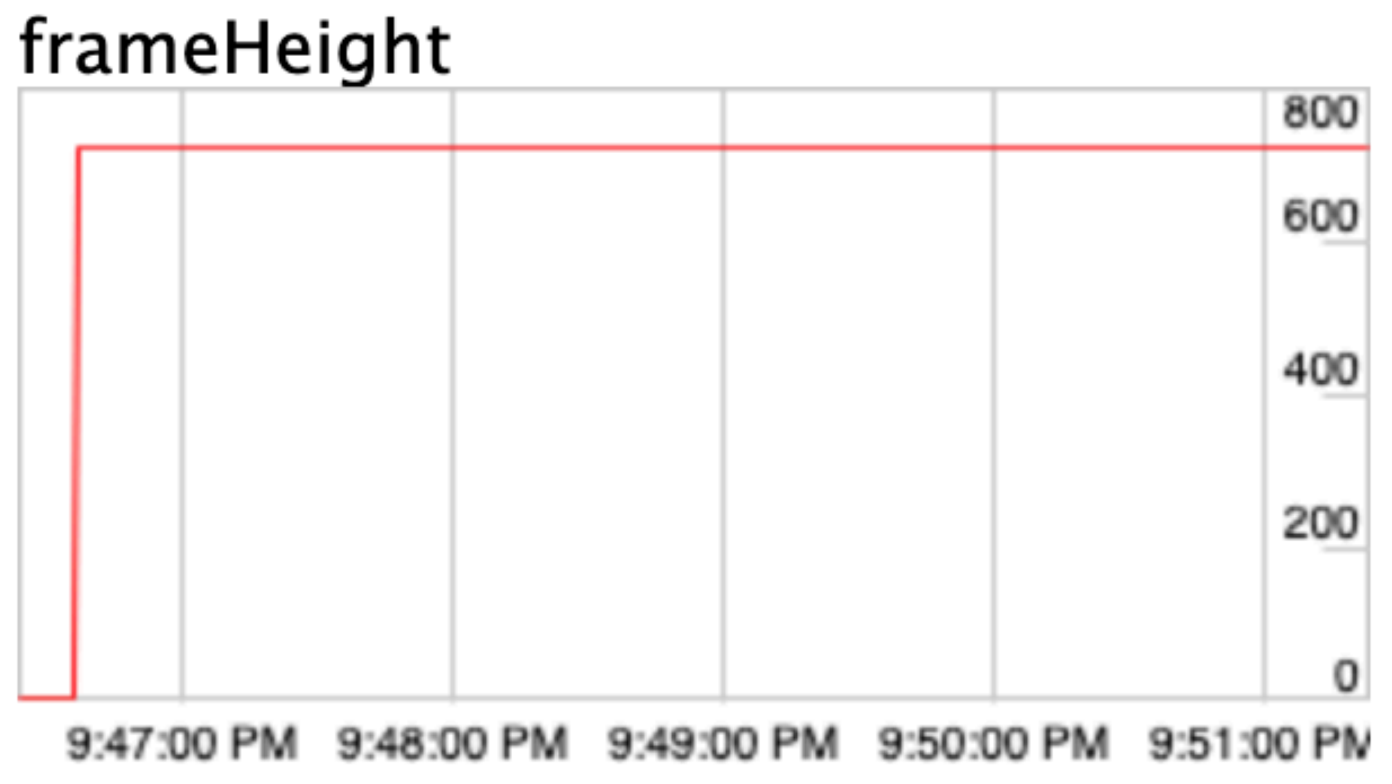
- Path Characteristics:
  - Baseline RTT: ~ 5ms
  - Max RTT: 1.2 sec
  - Overall loss ratio: 1.6%

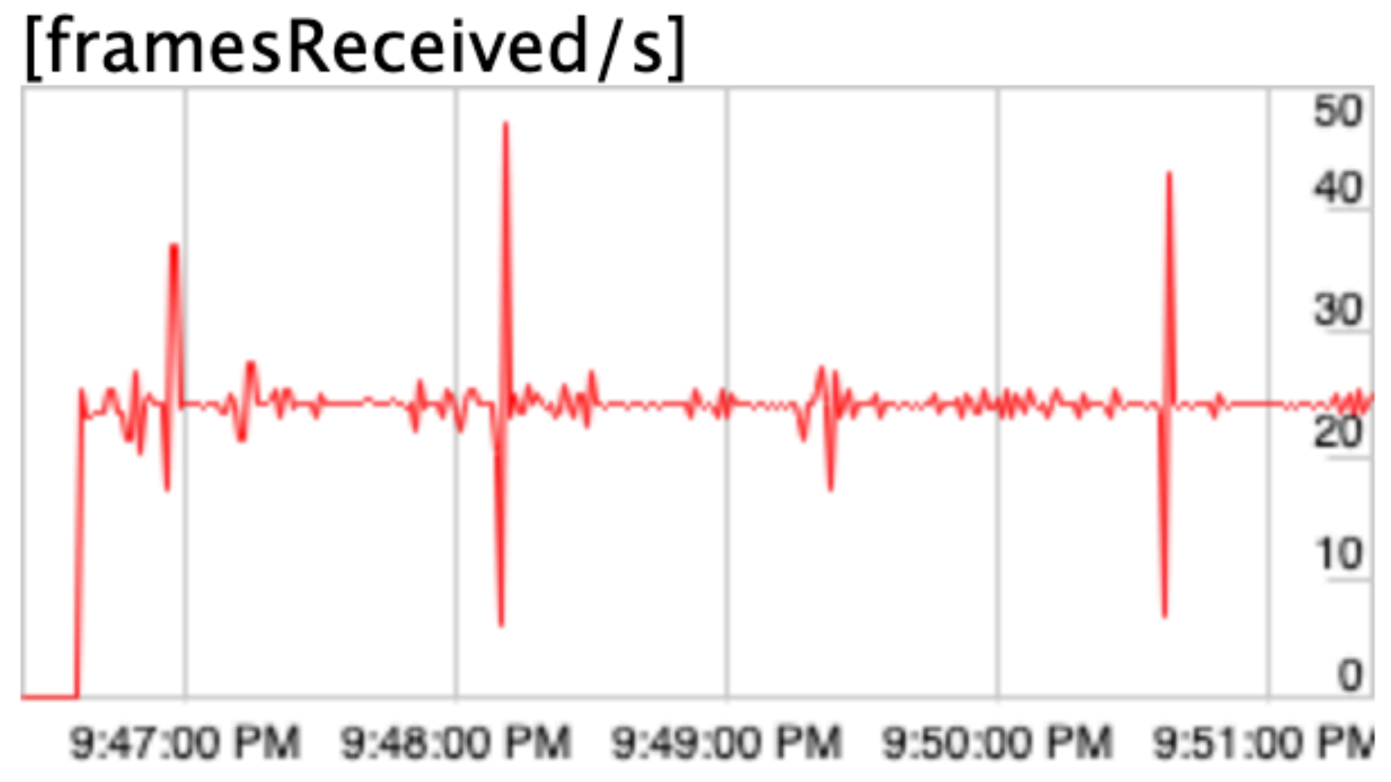


Scenario A: Screenshots from Chrome

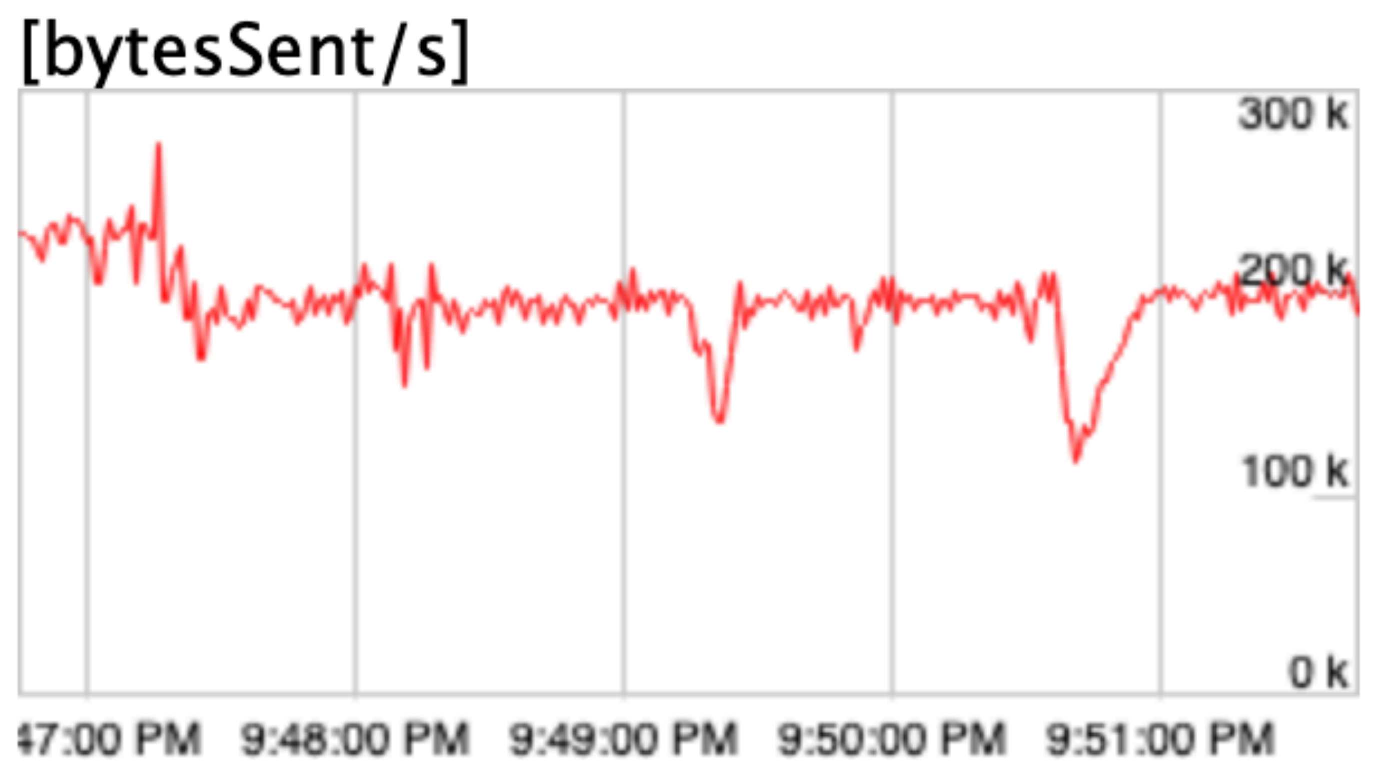
A —> B

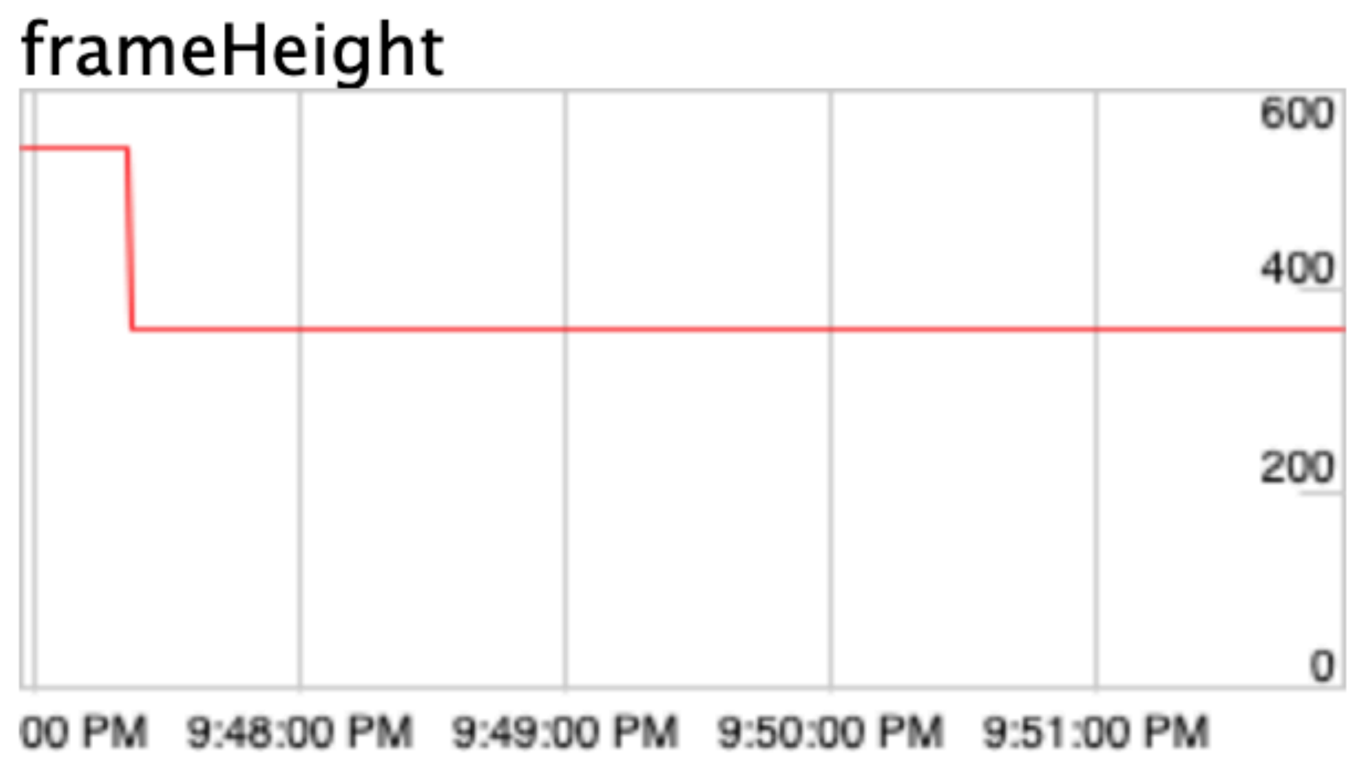


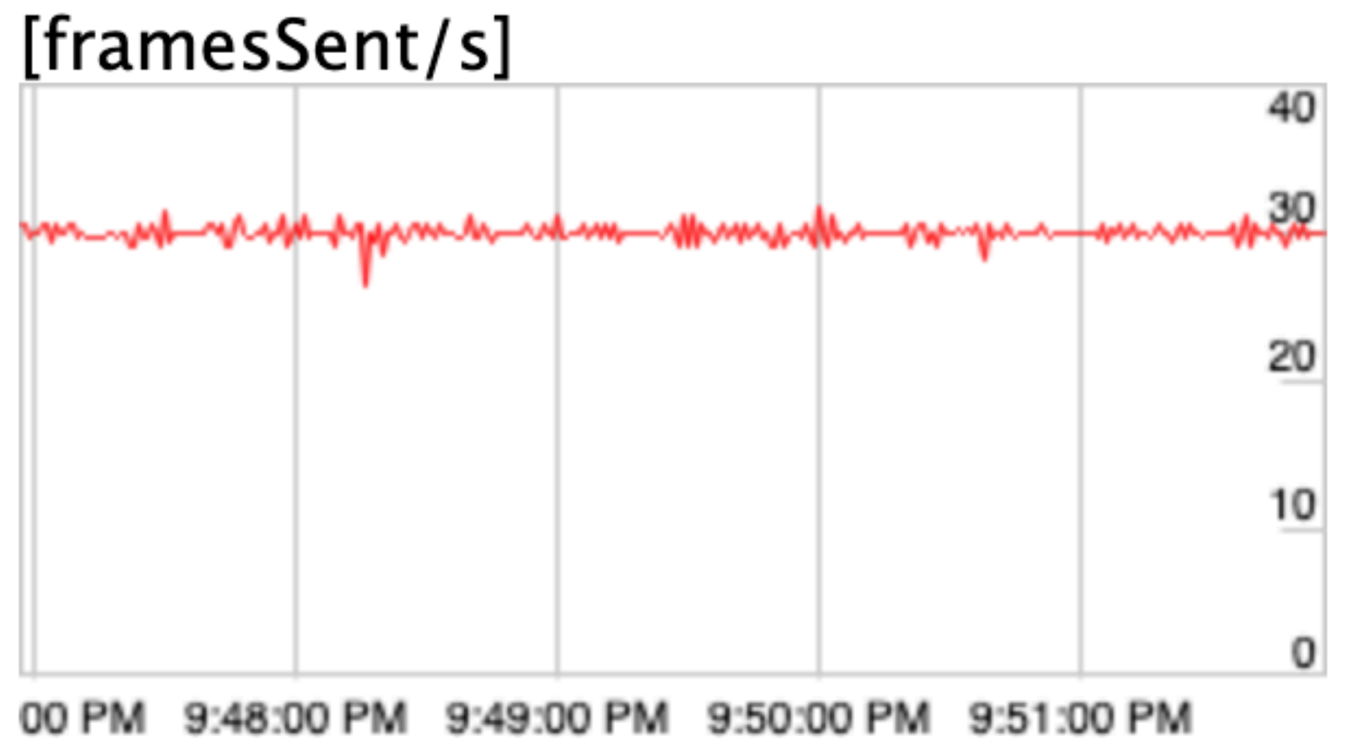




B —> A





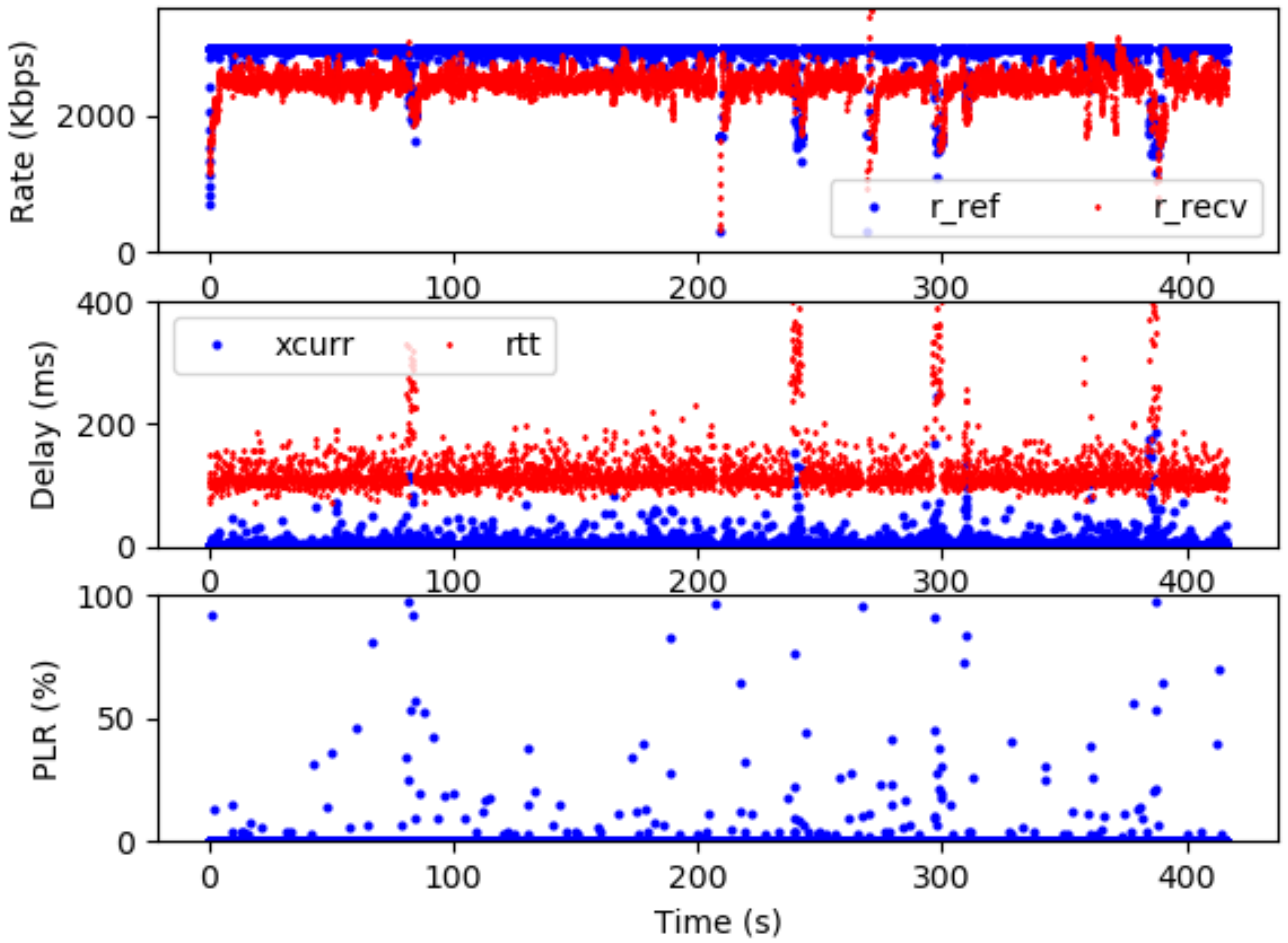


Scenario B.1: Remote Session within US

Client A:
- Location: Austin, Texas
- Connection: enterprise Wi-Fi

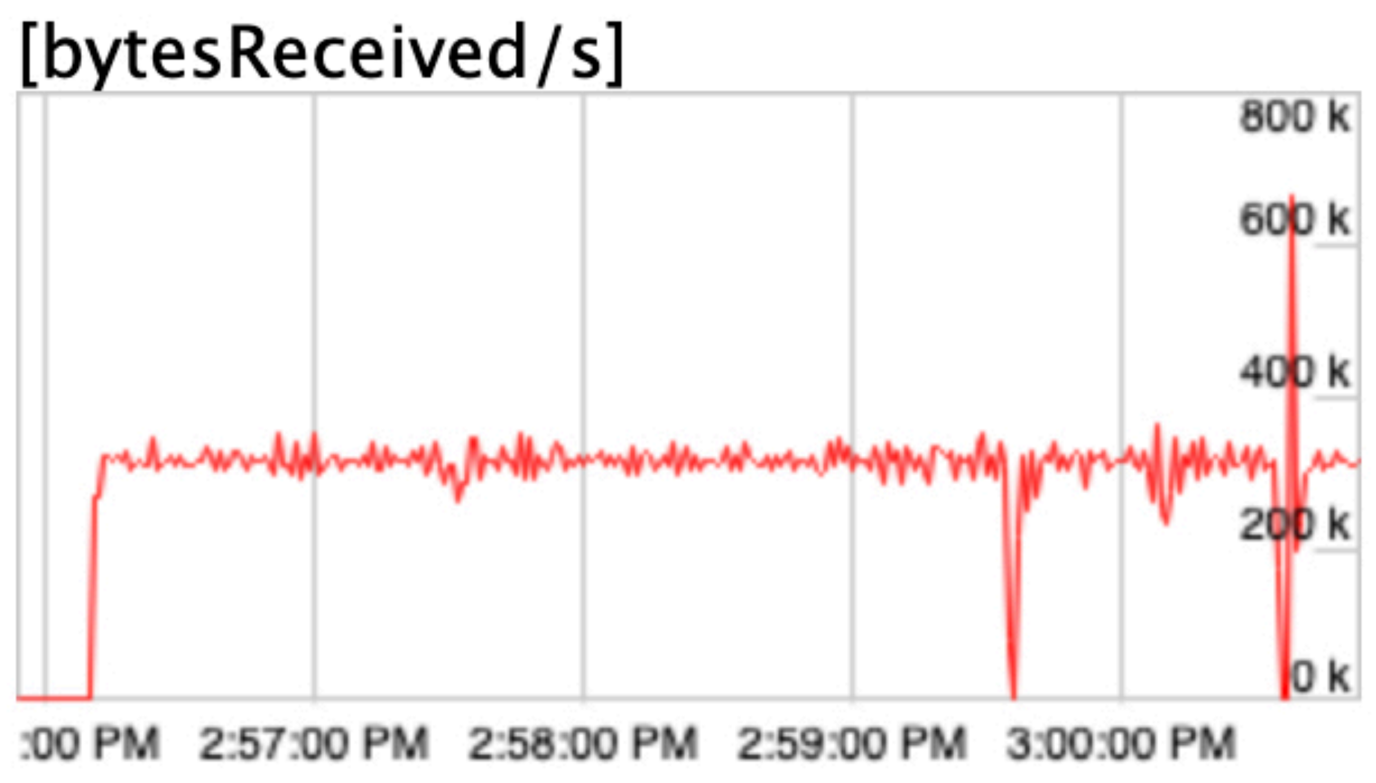
Client B:
- Location: San Jose, California
- Connection: enterprise Wi-Fi

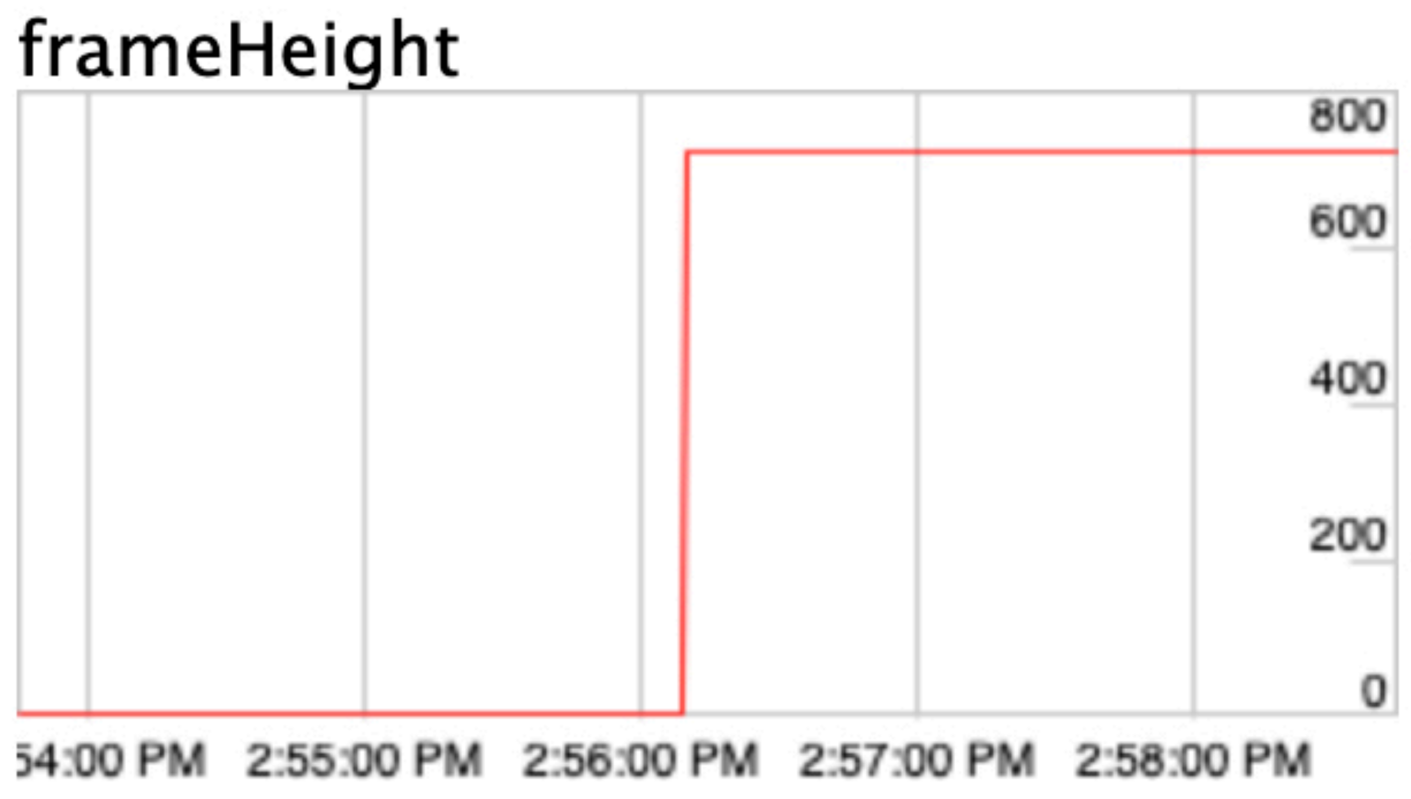
- Path Characteristic:
  - Baseline RTT: ~ 60ms
  - Max RTT: ~ 2.5s
  - Overall loss ratio: 2.2%

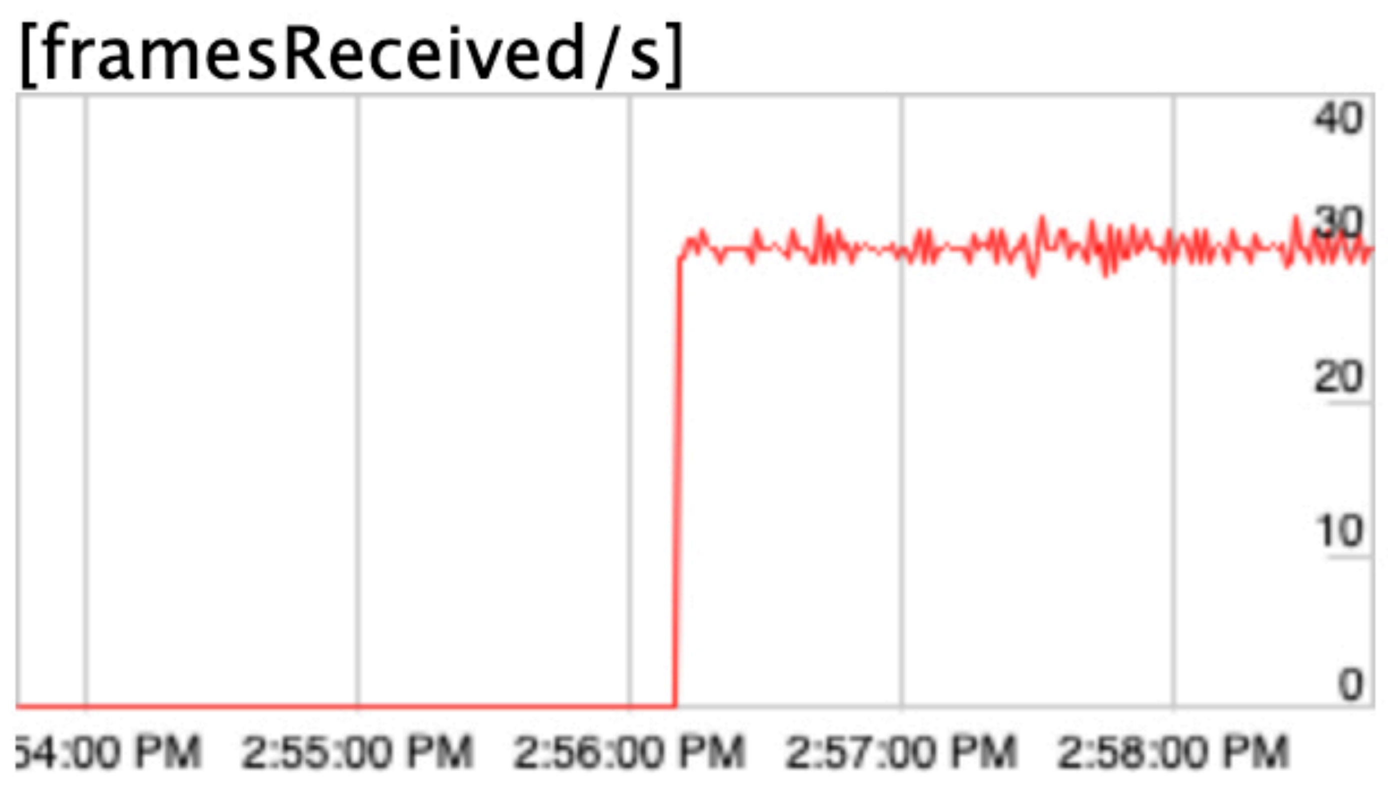


Scenario B.1: Screenshots from Chrome

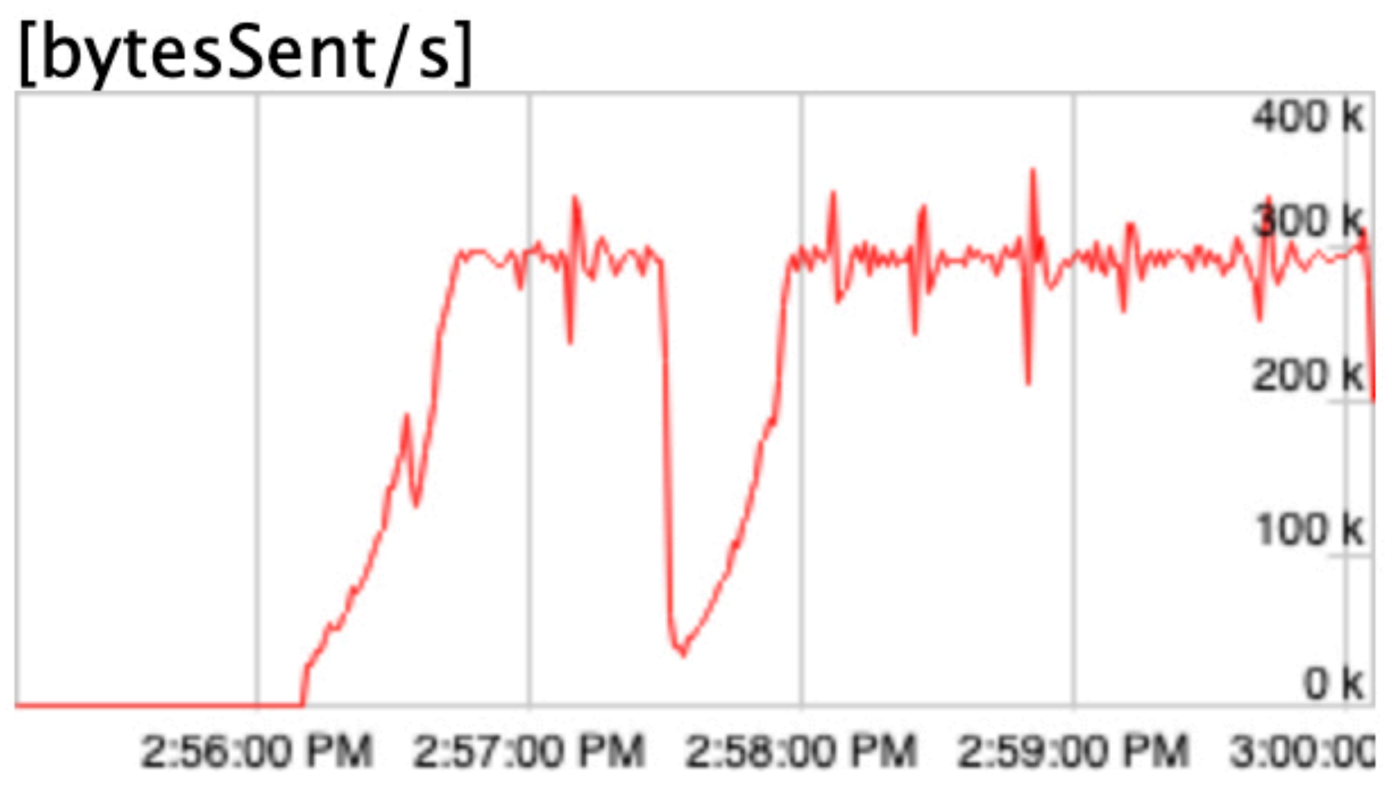
A —> B

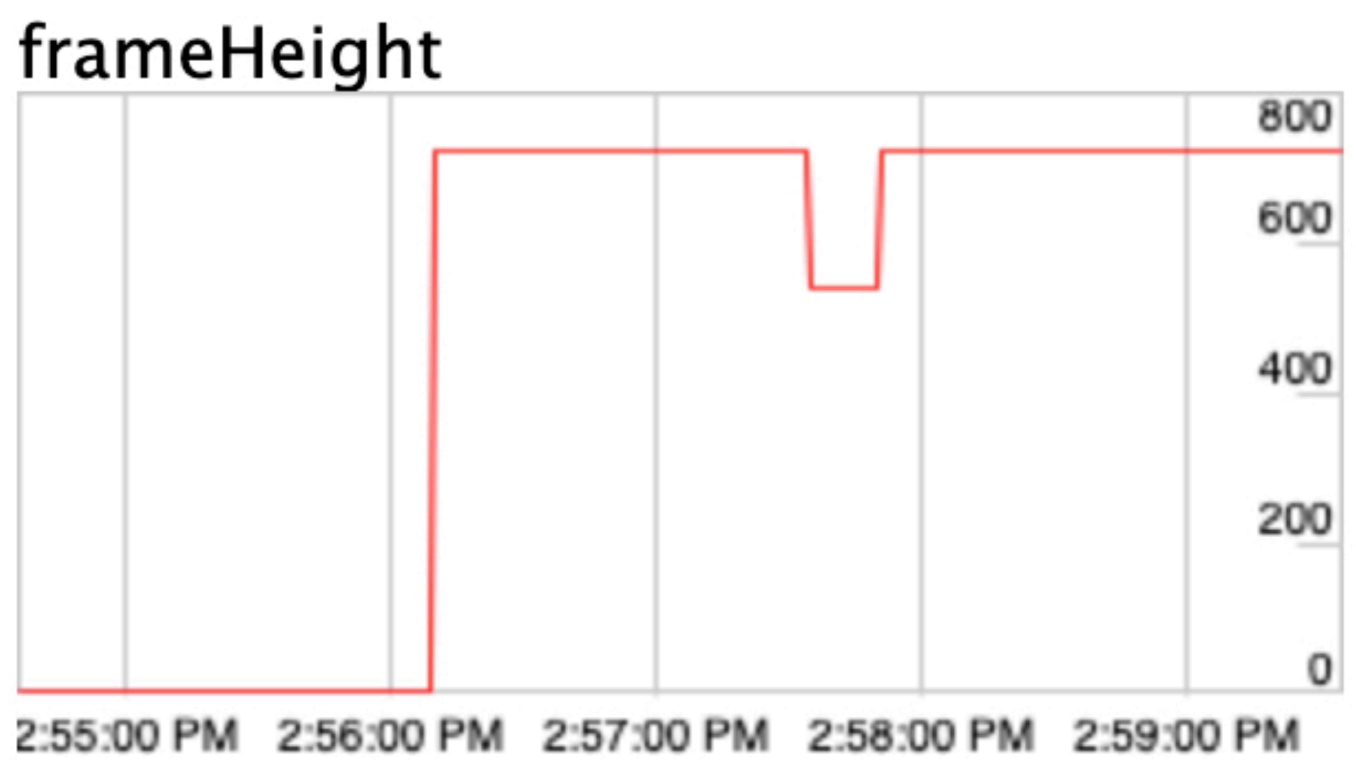


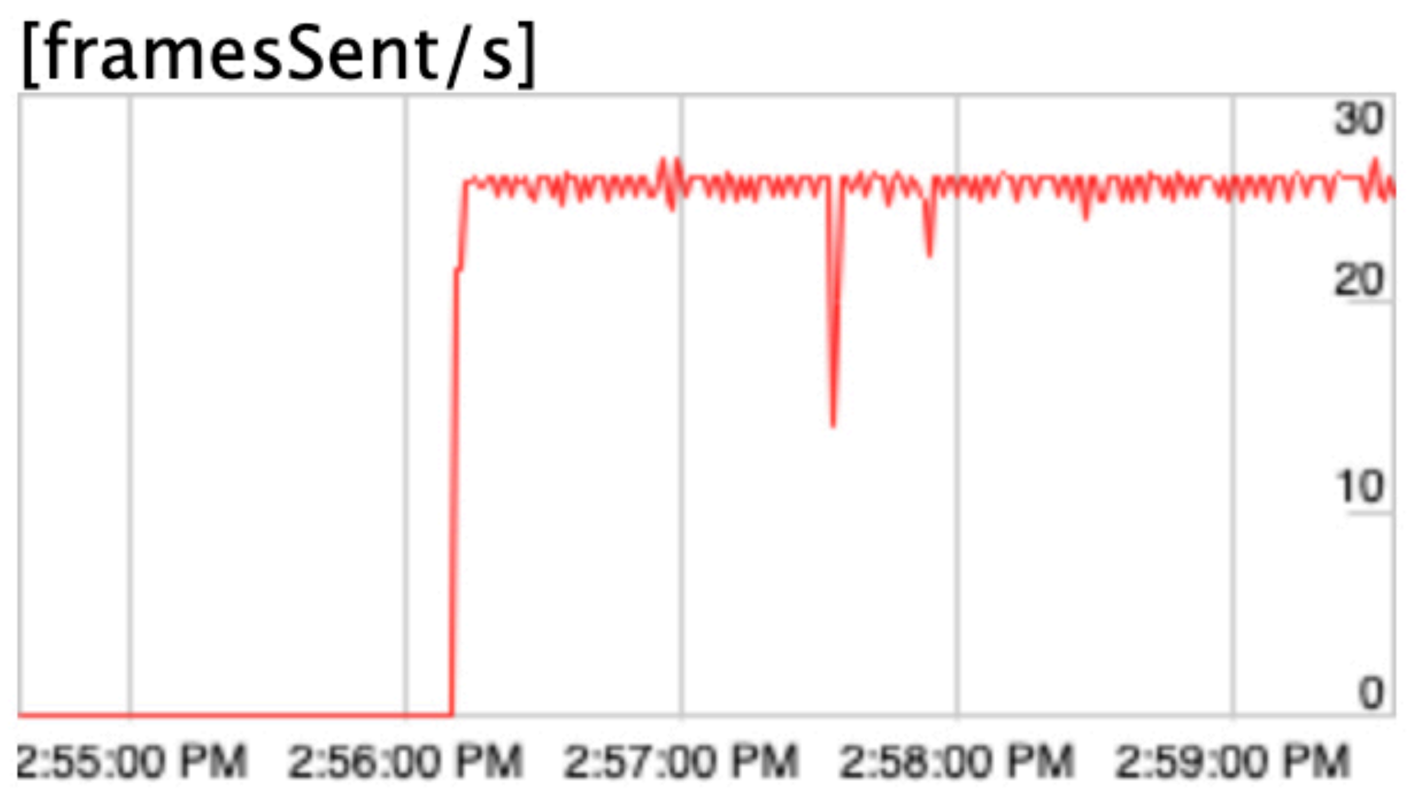




B —> A





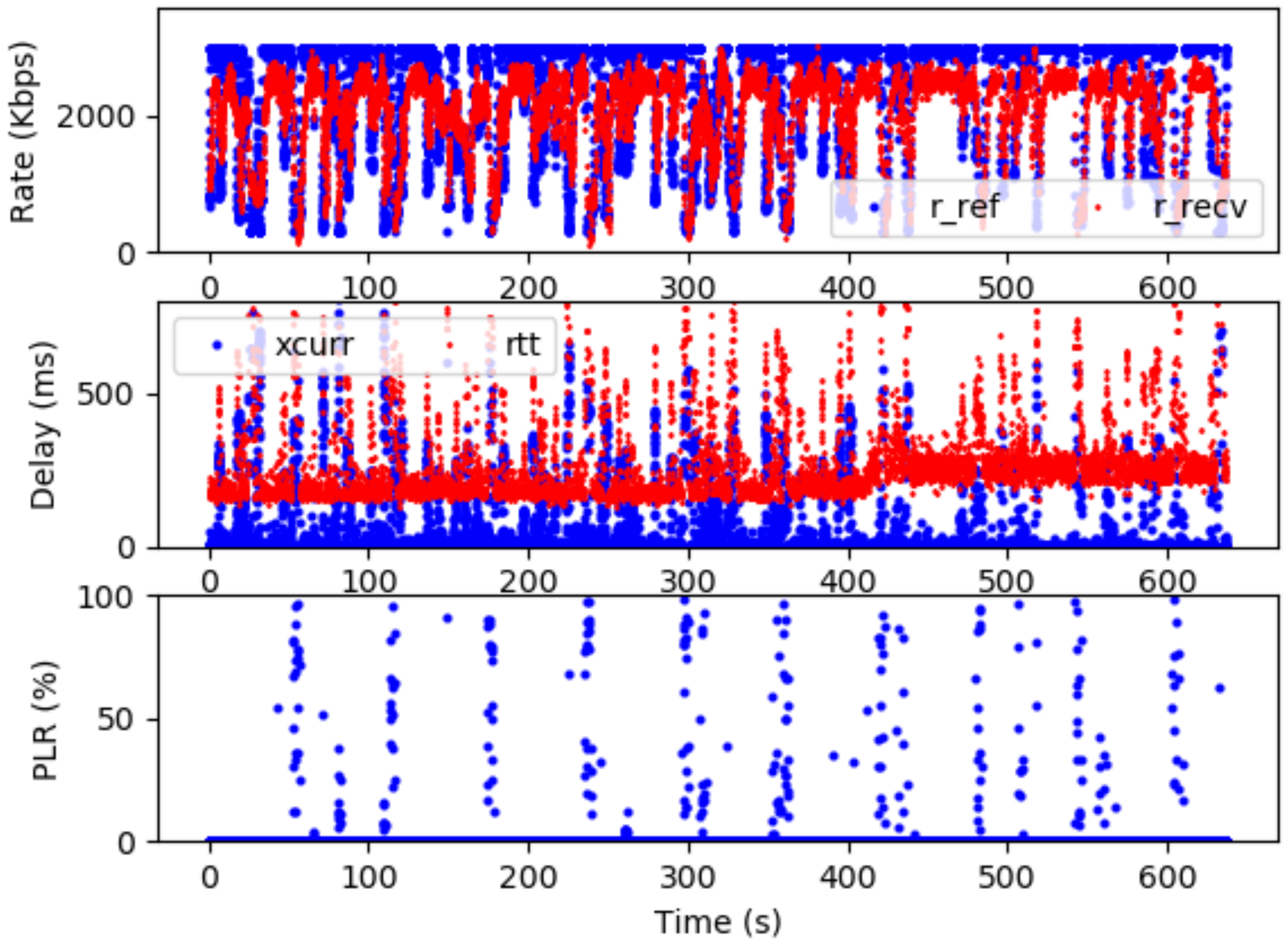


Scenario B.2: Remote Session within US

Client A:
- Location: Austin, Texas
- Connection: enterprise Wi-Fi

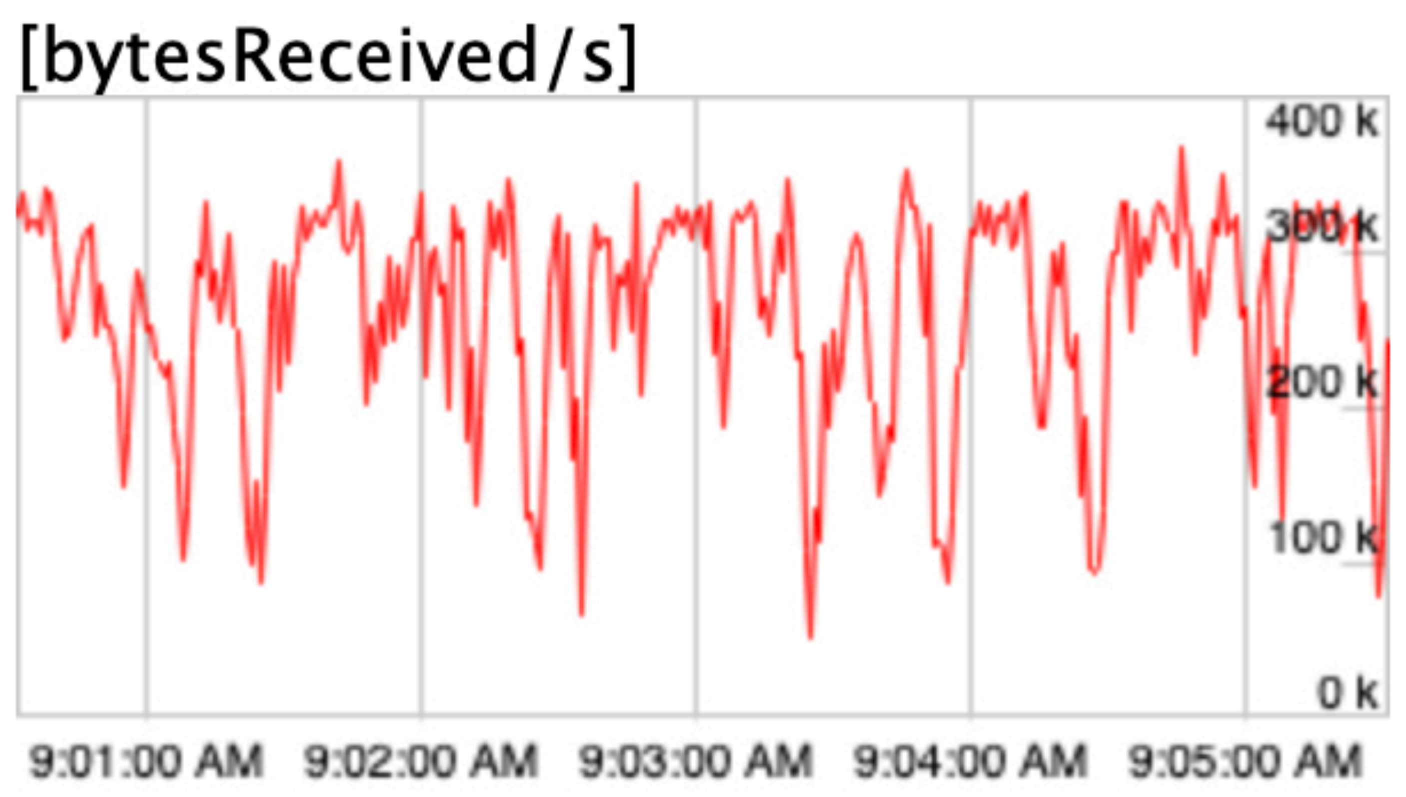
Client B:
- Location: San Jose, California
- Connection: home Wi-Fi

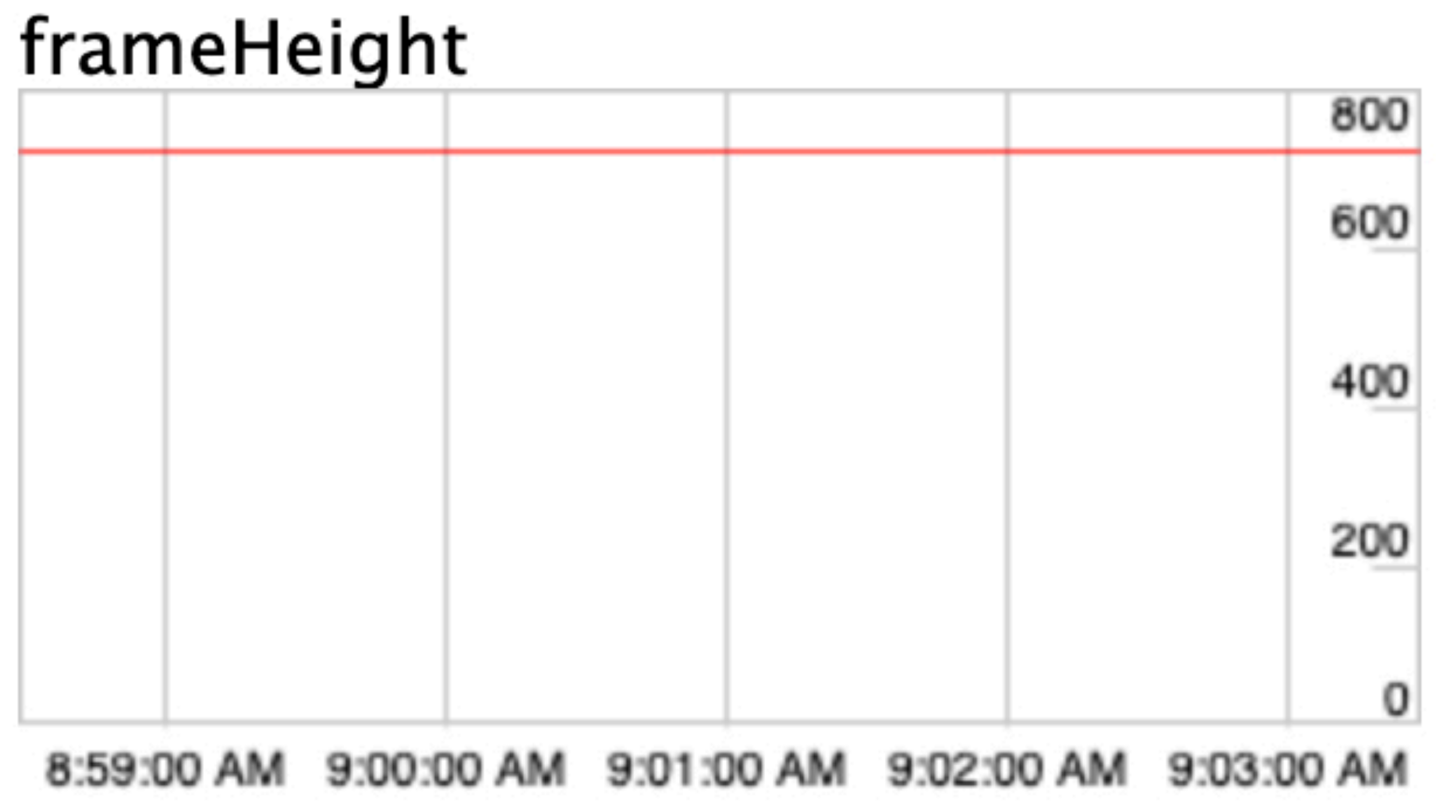
- Path Characteristics:
  - Baseline RTT: ~ 110ms
  - Max RTT: 1.9 sec
  - Overall loss ratio: 3.0%

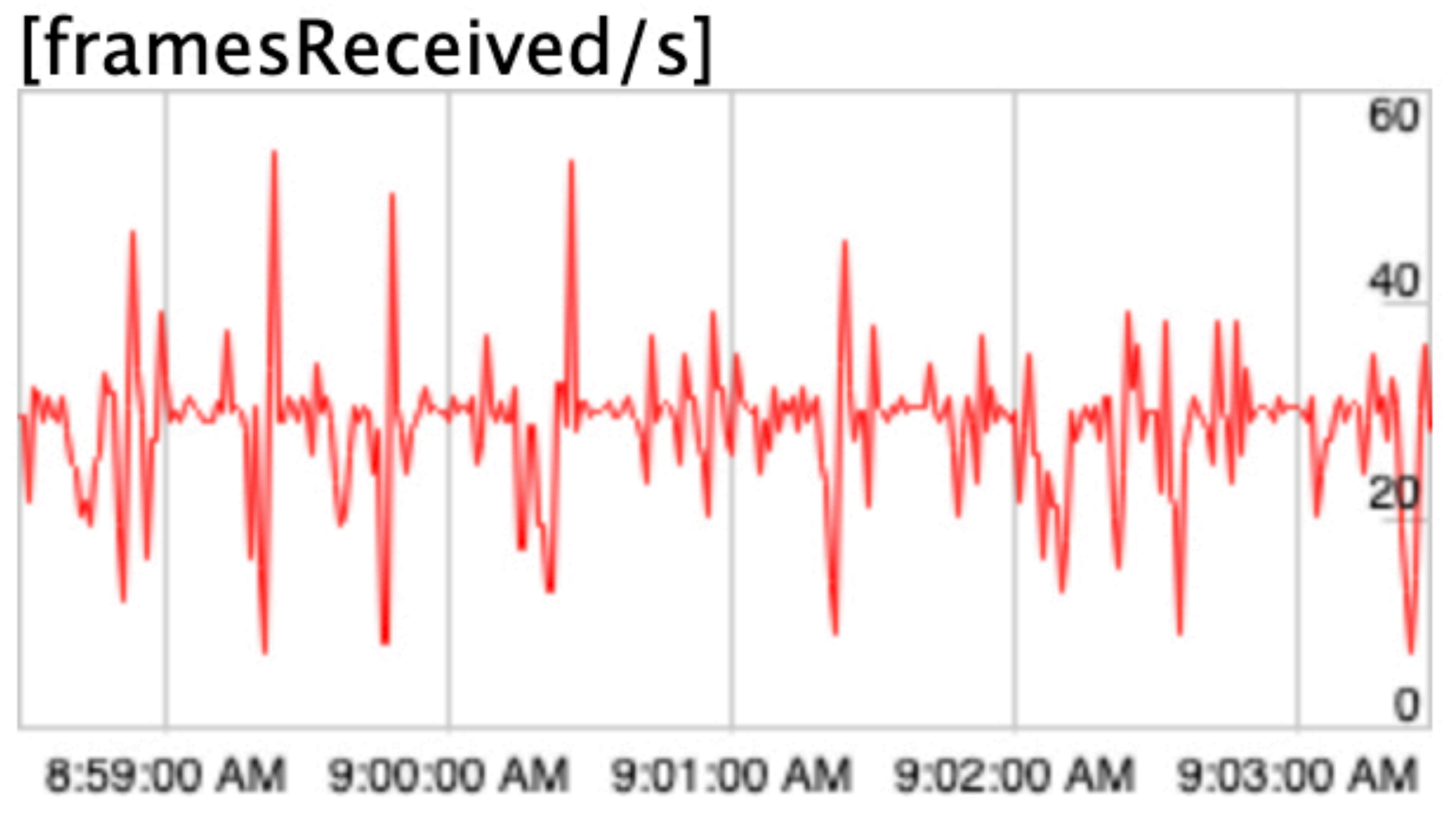


Scenario B.2: Screenshots from Chrome

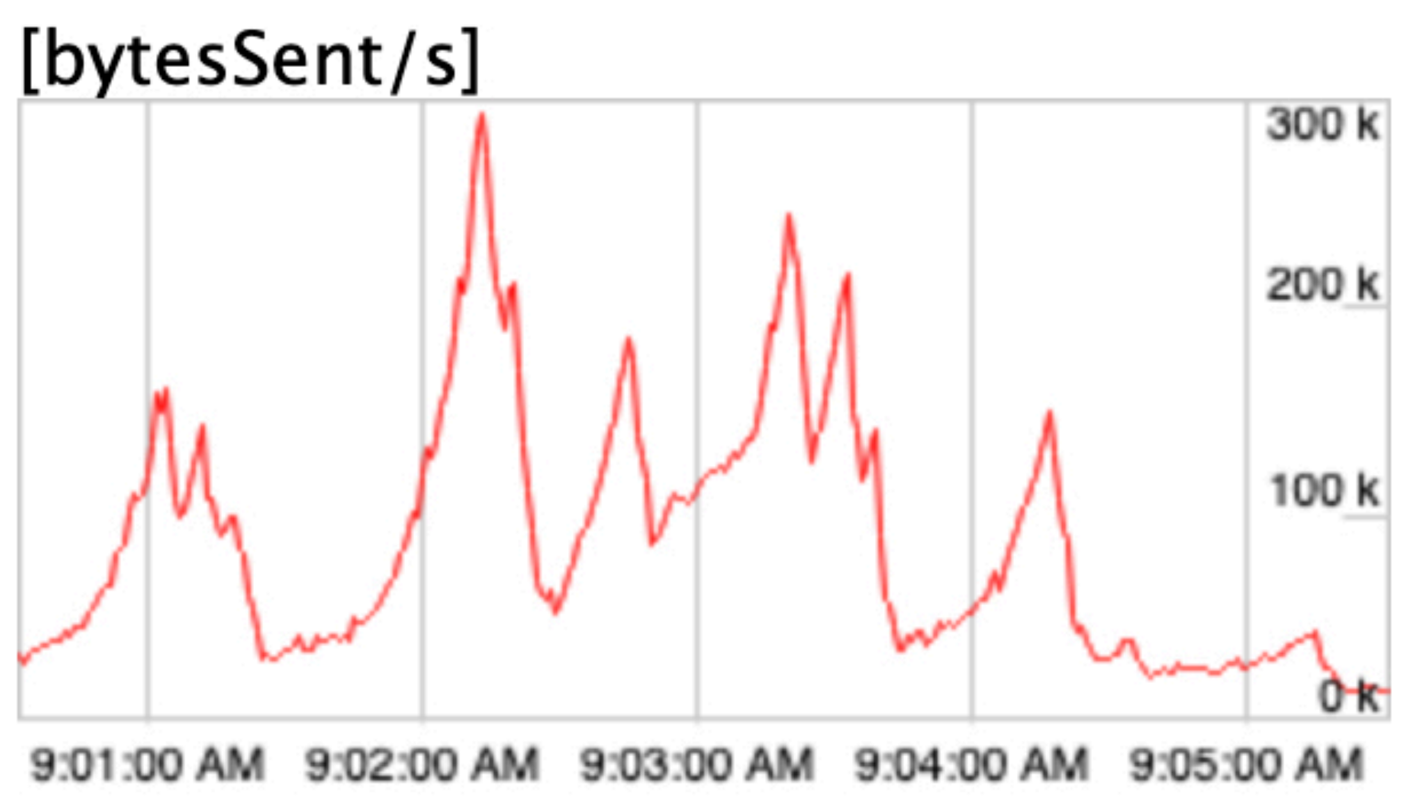
A —> B

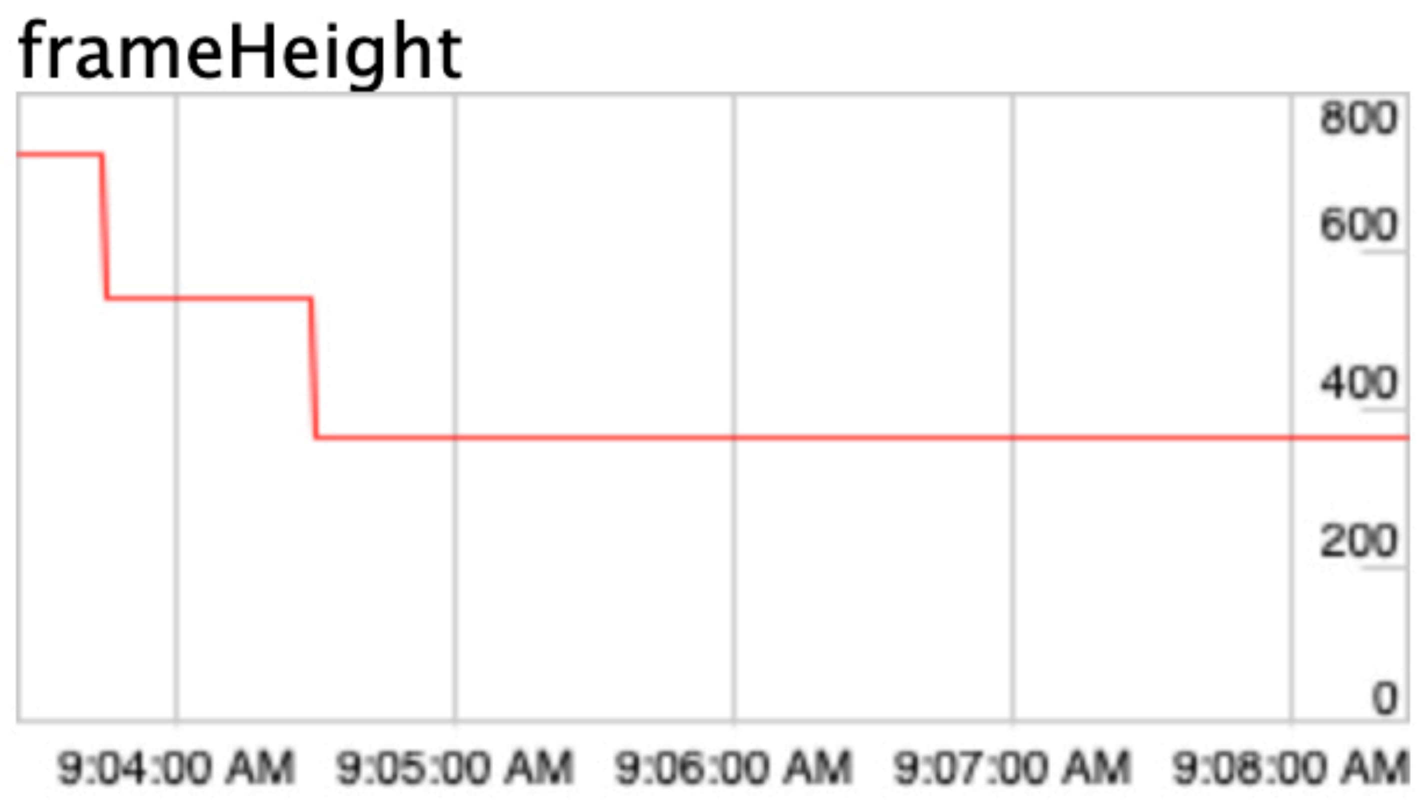


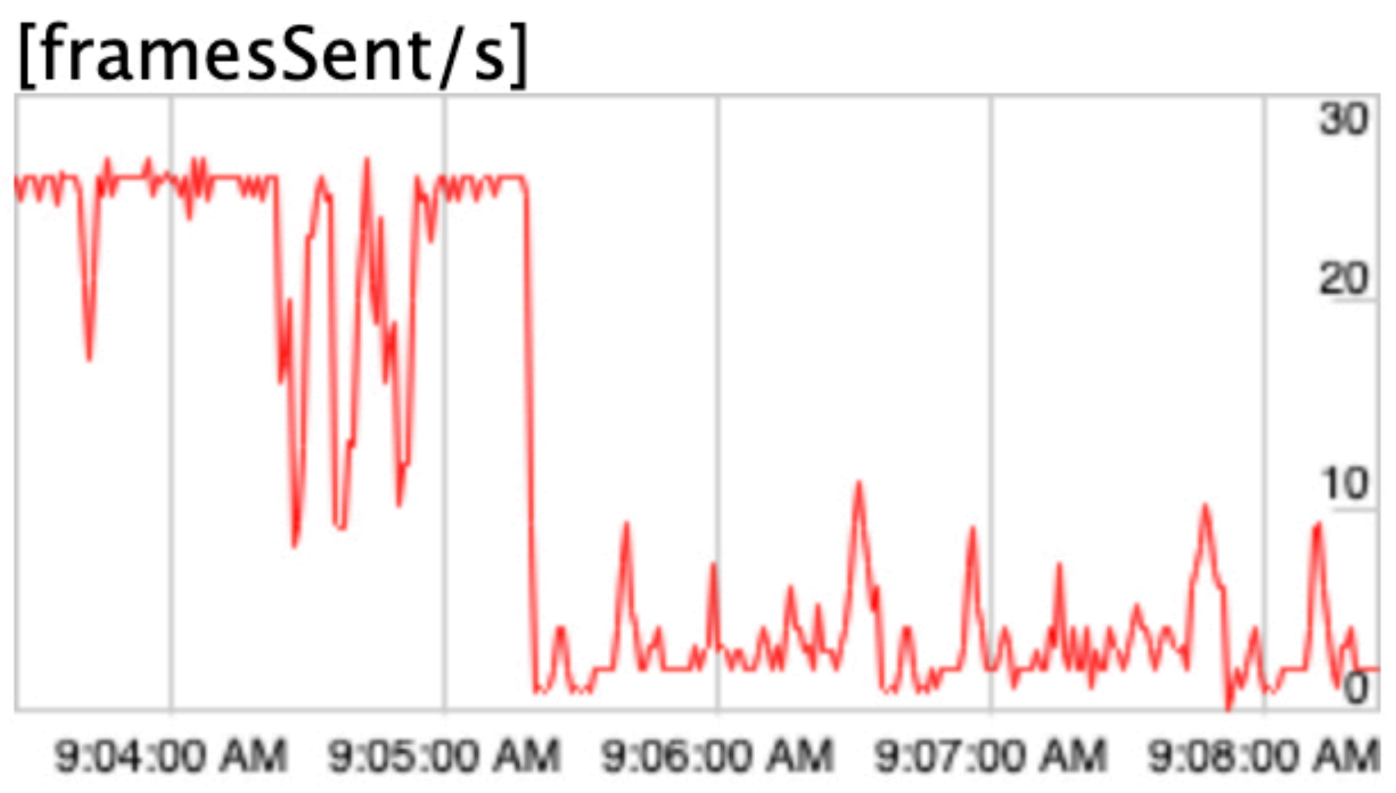




B —> A





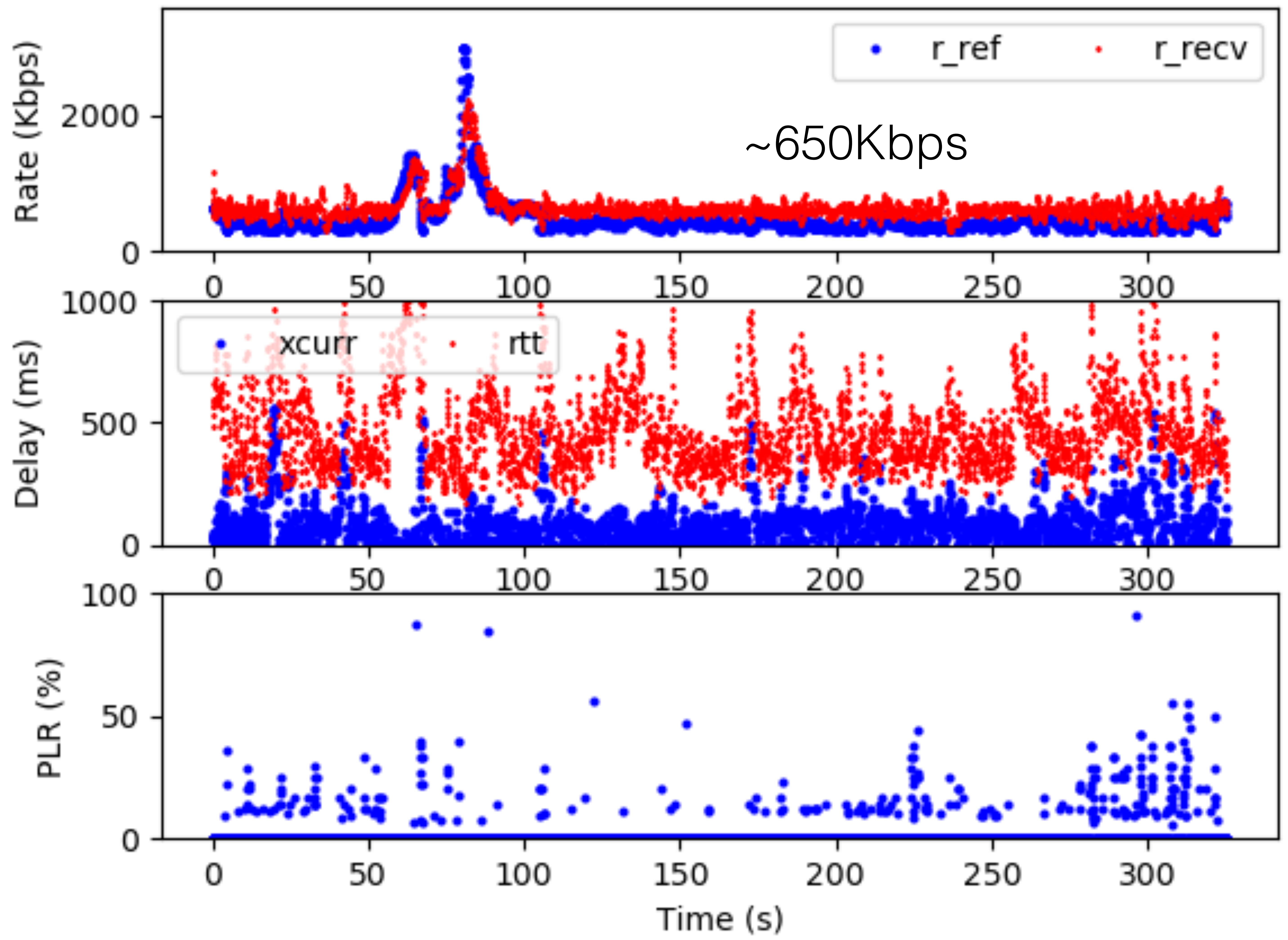


Scenario B.3: Remote Session within US

Client A:
- Location: Austin, Texas
- Connection: enterprise Wi-Fi

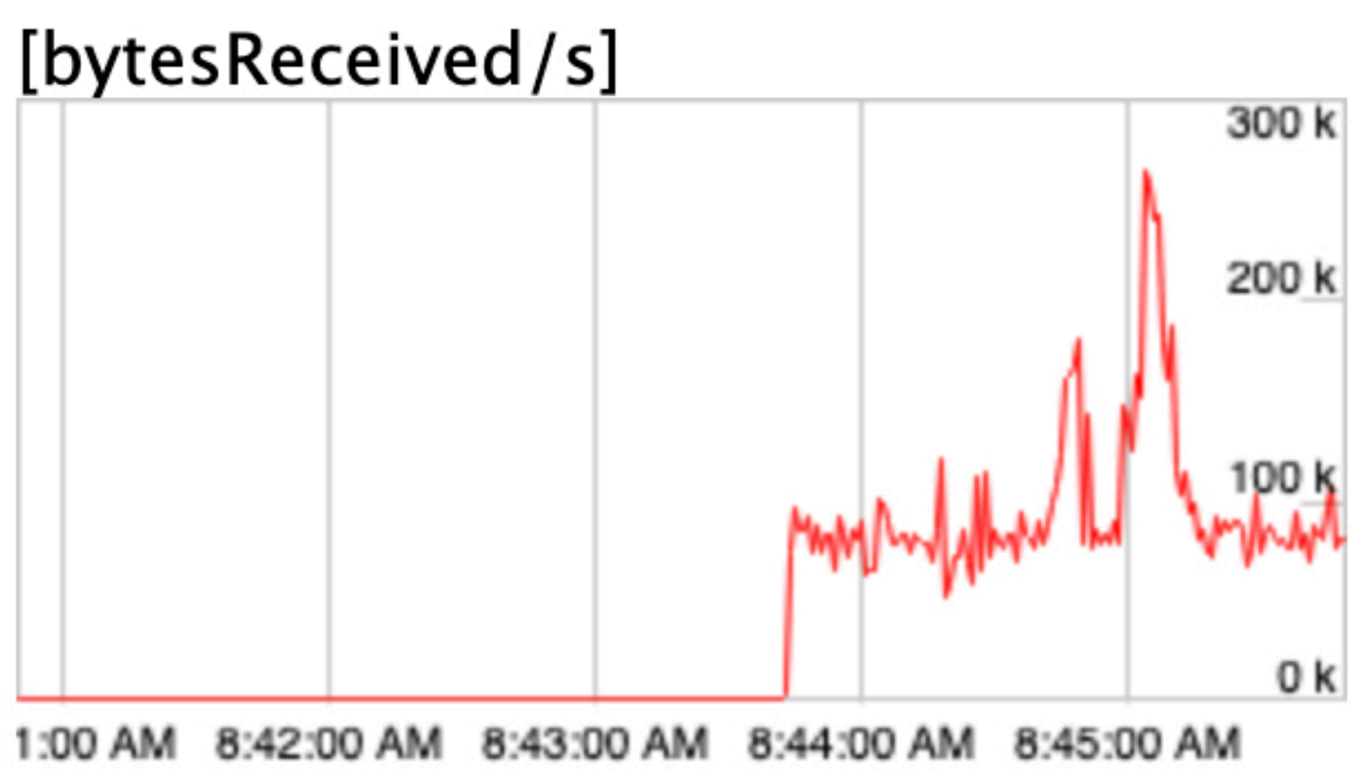
Client B:
- Location: San Jose, California
- Connection: home Wi-Fi
- Background Bit-Torrent traffic

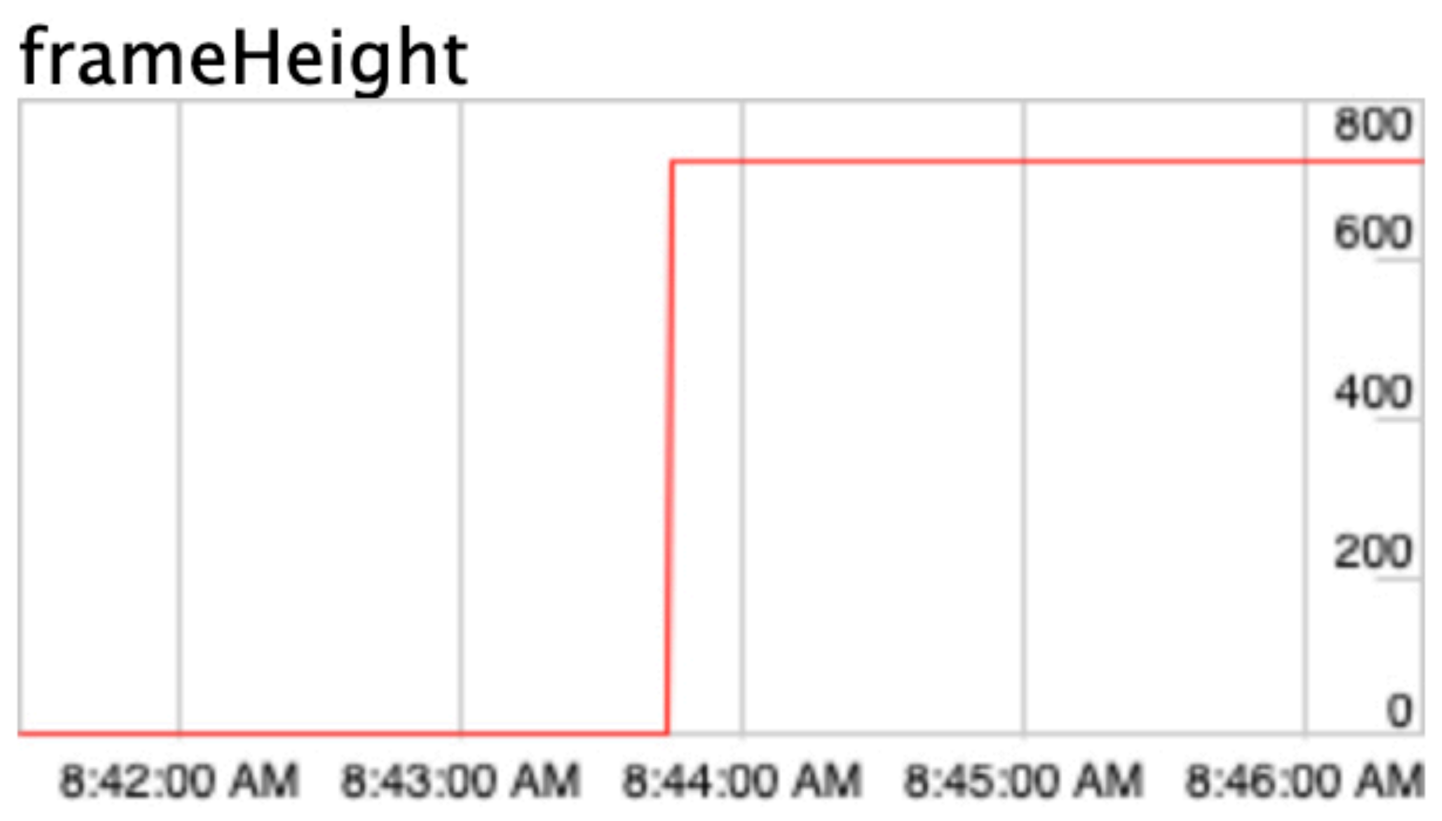
- Path Characteristics:
  - Baseline RTT: ~ 125ms
  - Max RTT: 1.5 sec
  - Overall loss ratio: 2.0%

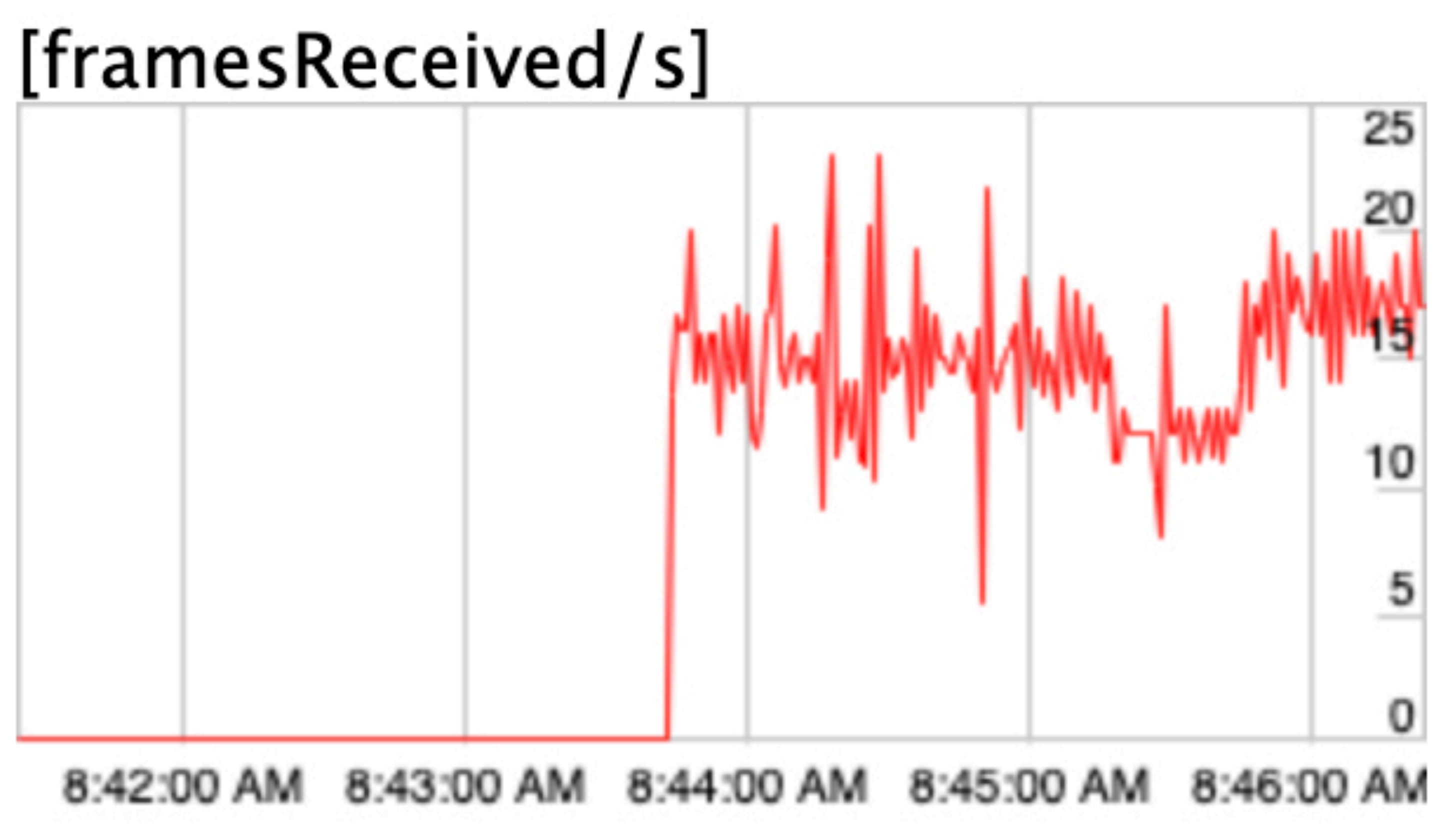


Scenario B.3: Screenshots from Chrome

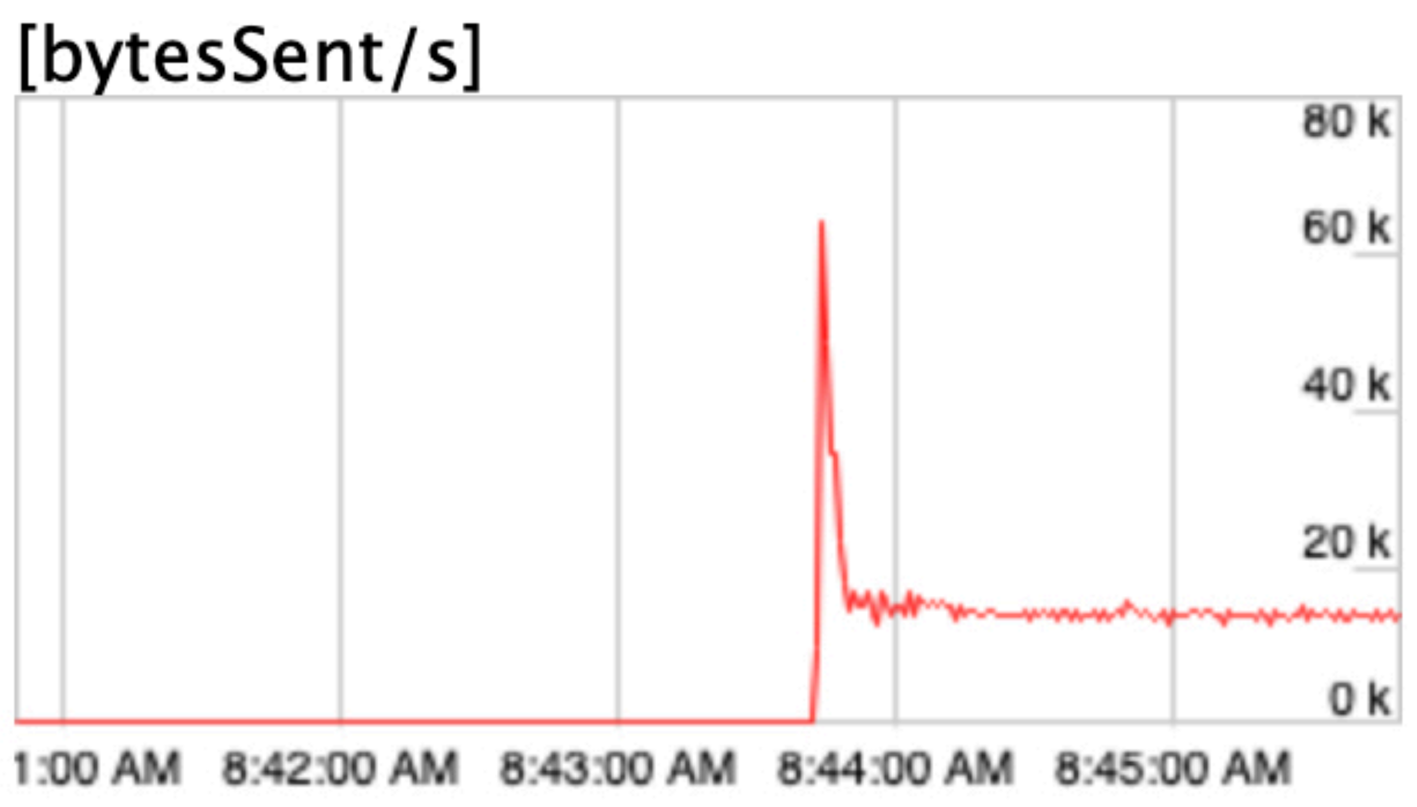
A —> B

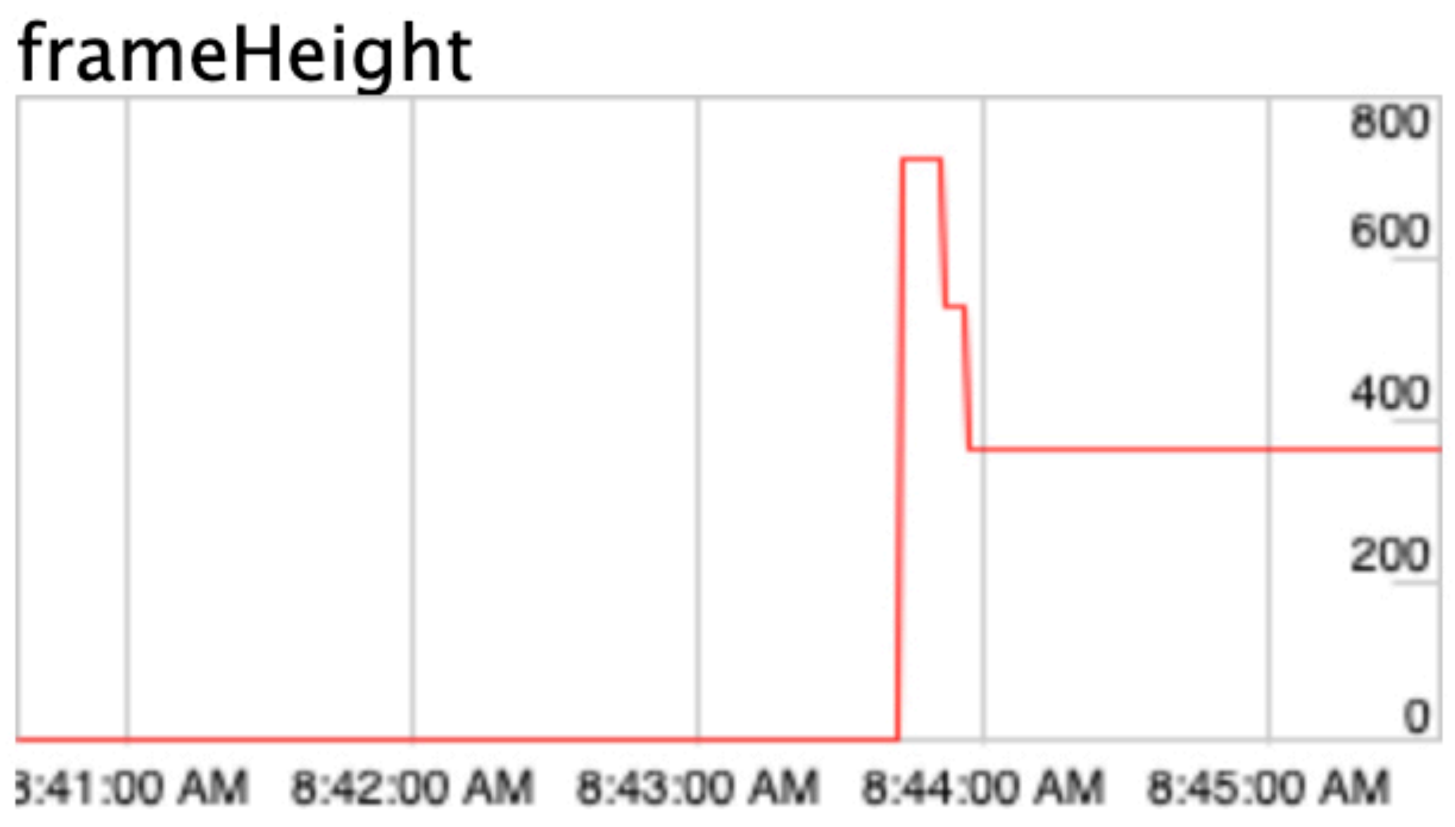


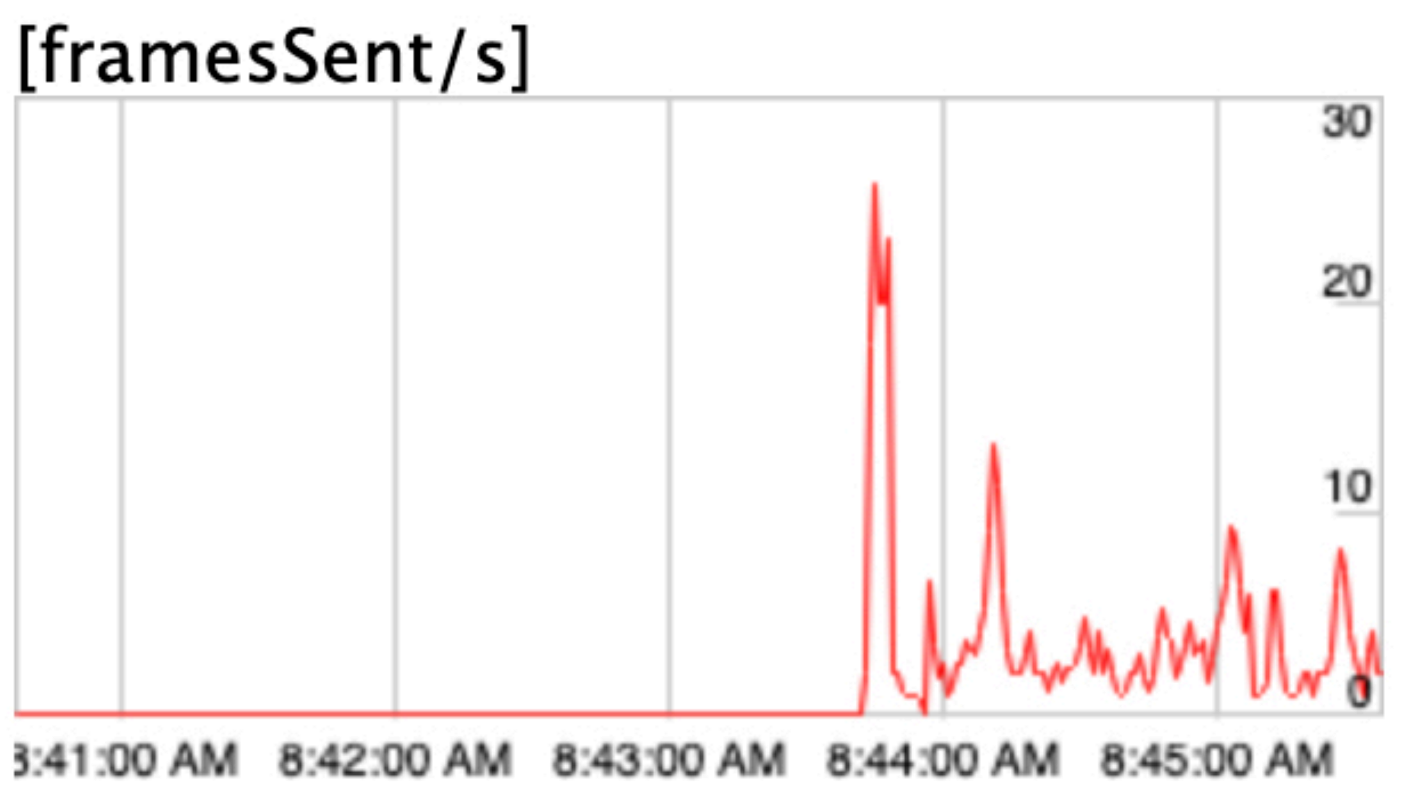




B —> A





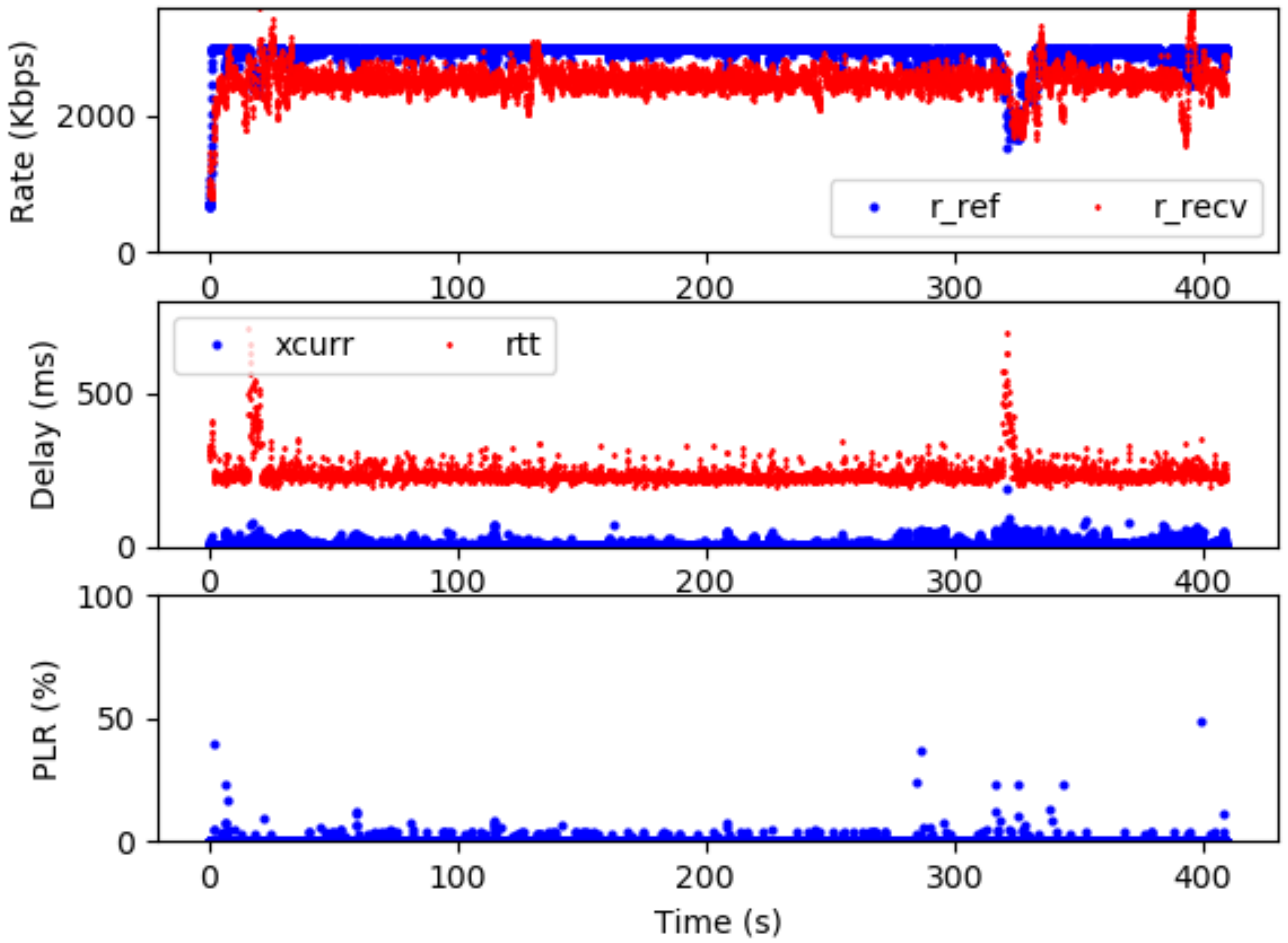


Scenario C.1: Remote Session across Atlantic

Client A:
- Location: Austin, Texas, USA
- Connection: enterprise Wi-Fi

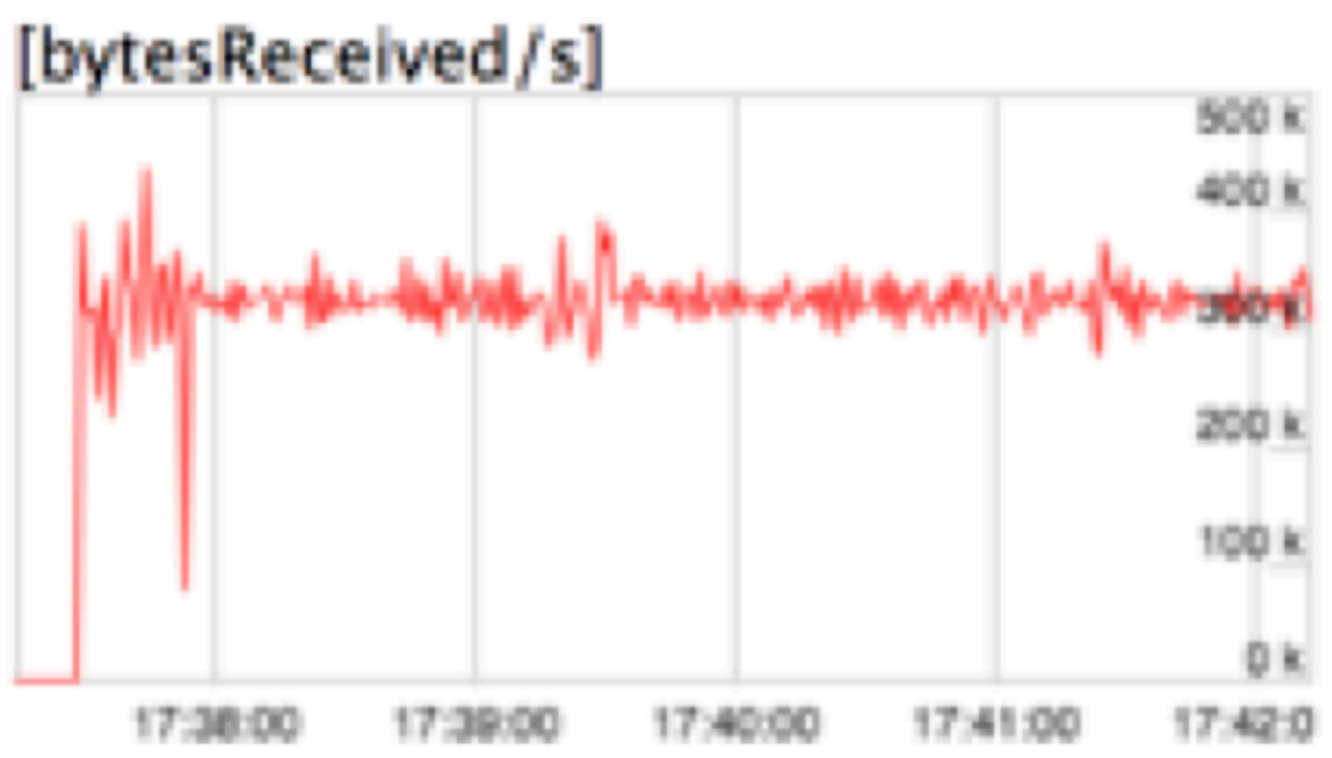
Client B:
- Location: Lausanne, Switzerlan
- Connection: enterprise Ethernet

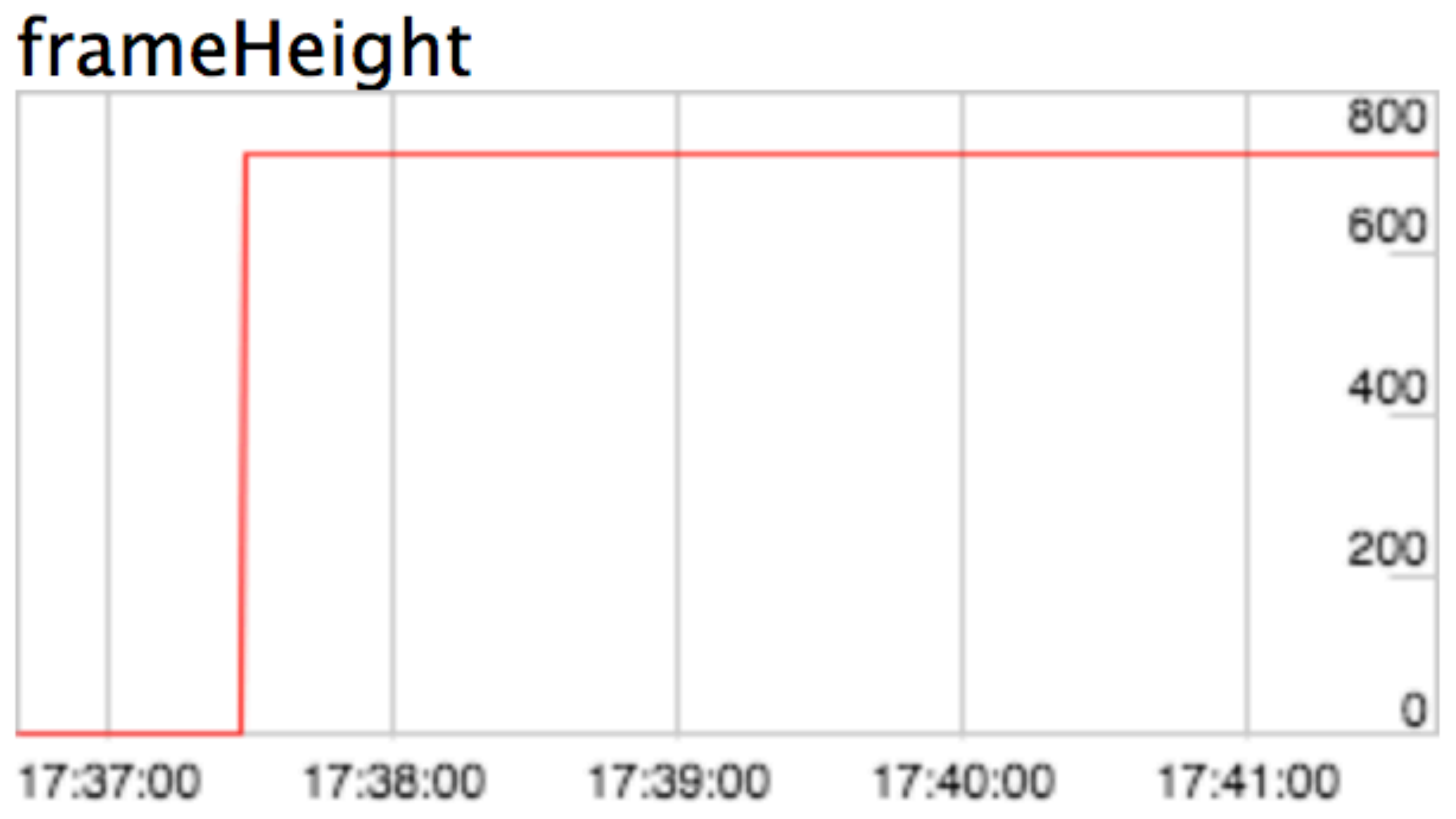
- Path Characteristics:
  - Baseline RTT: ~180ms
  - Max RTT: ~ 7.1s
  - Overall loss ratio: 0.2%

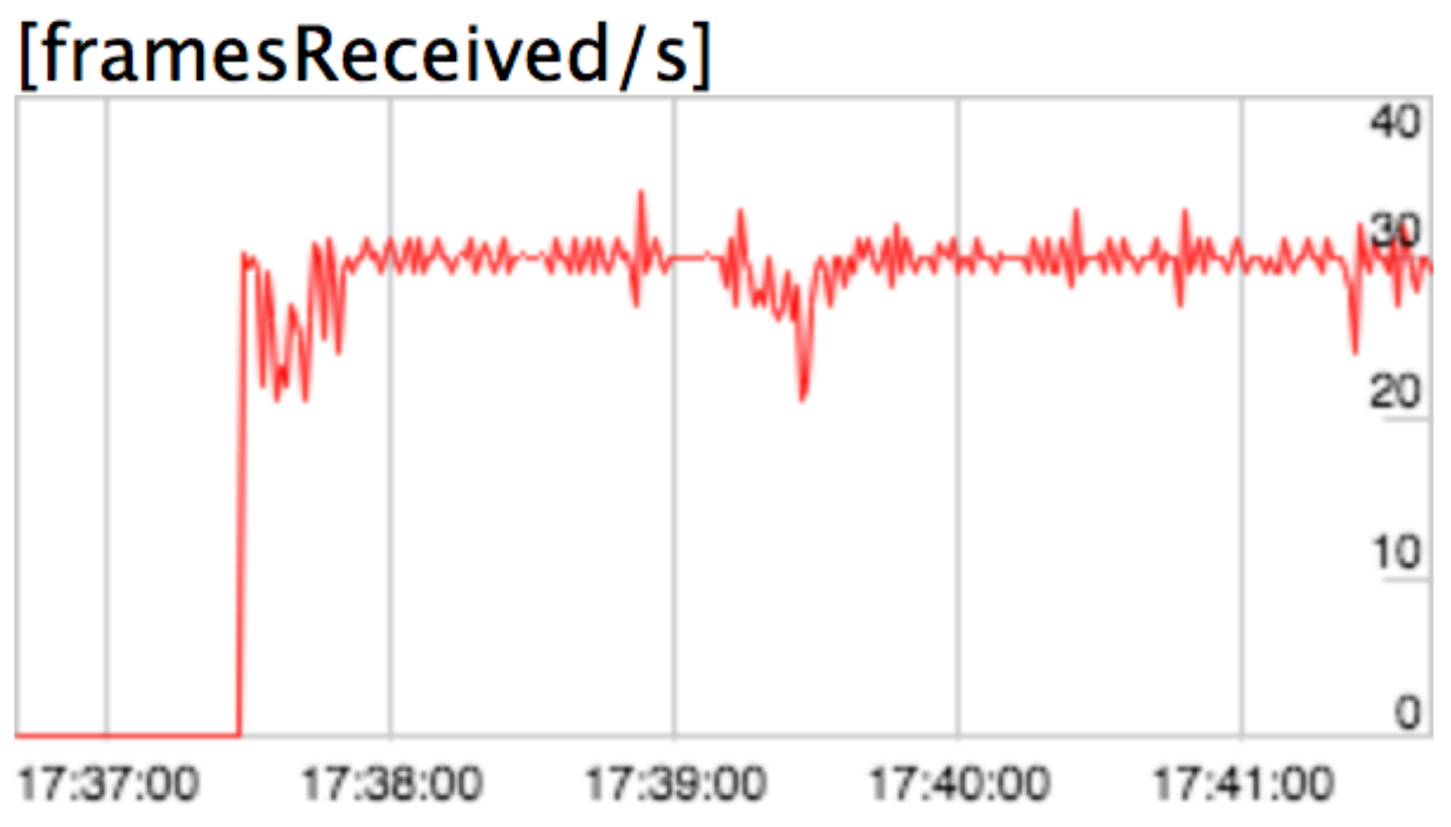


Scenario C.1: Screenshots from Chrome

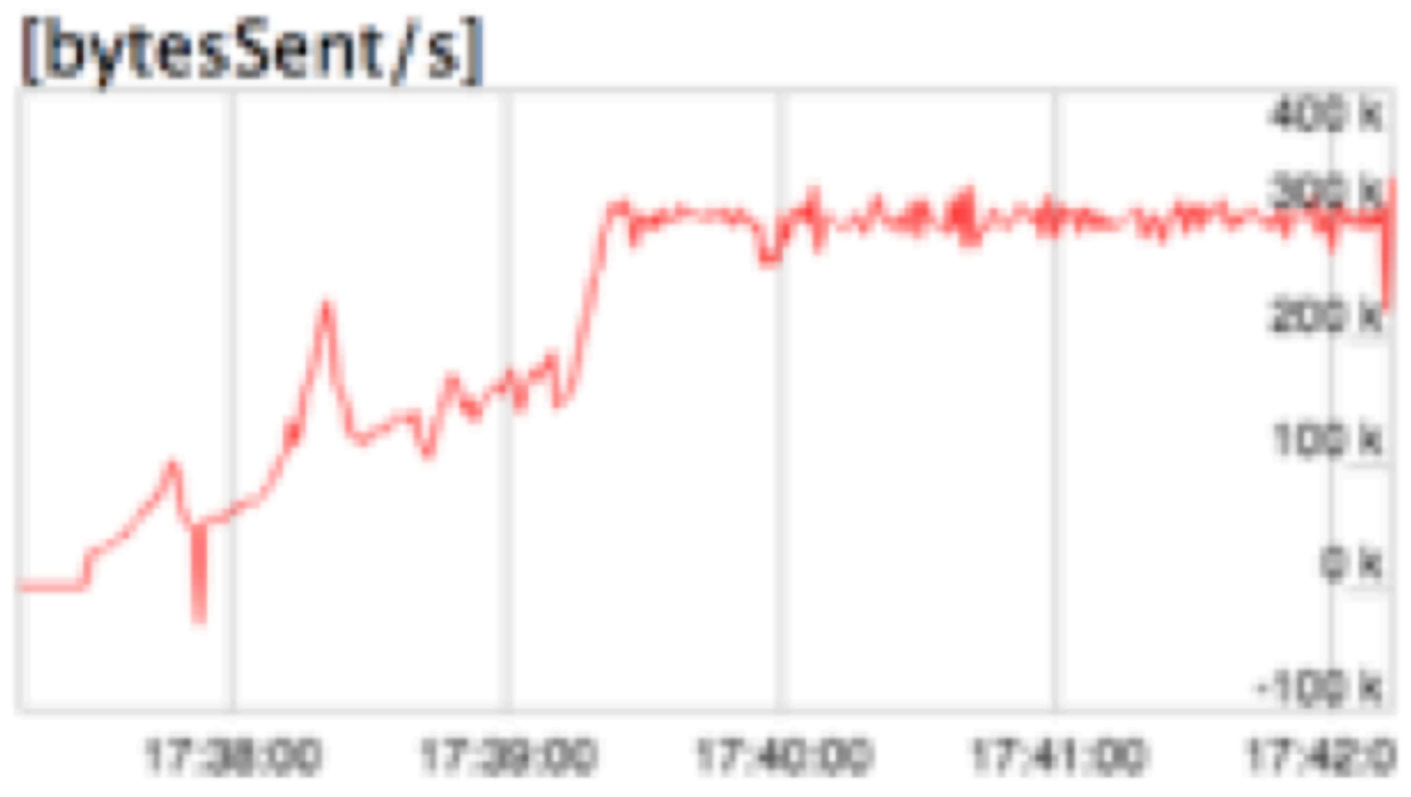
A —> B

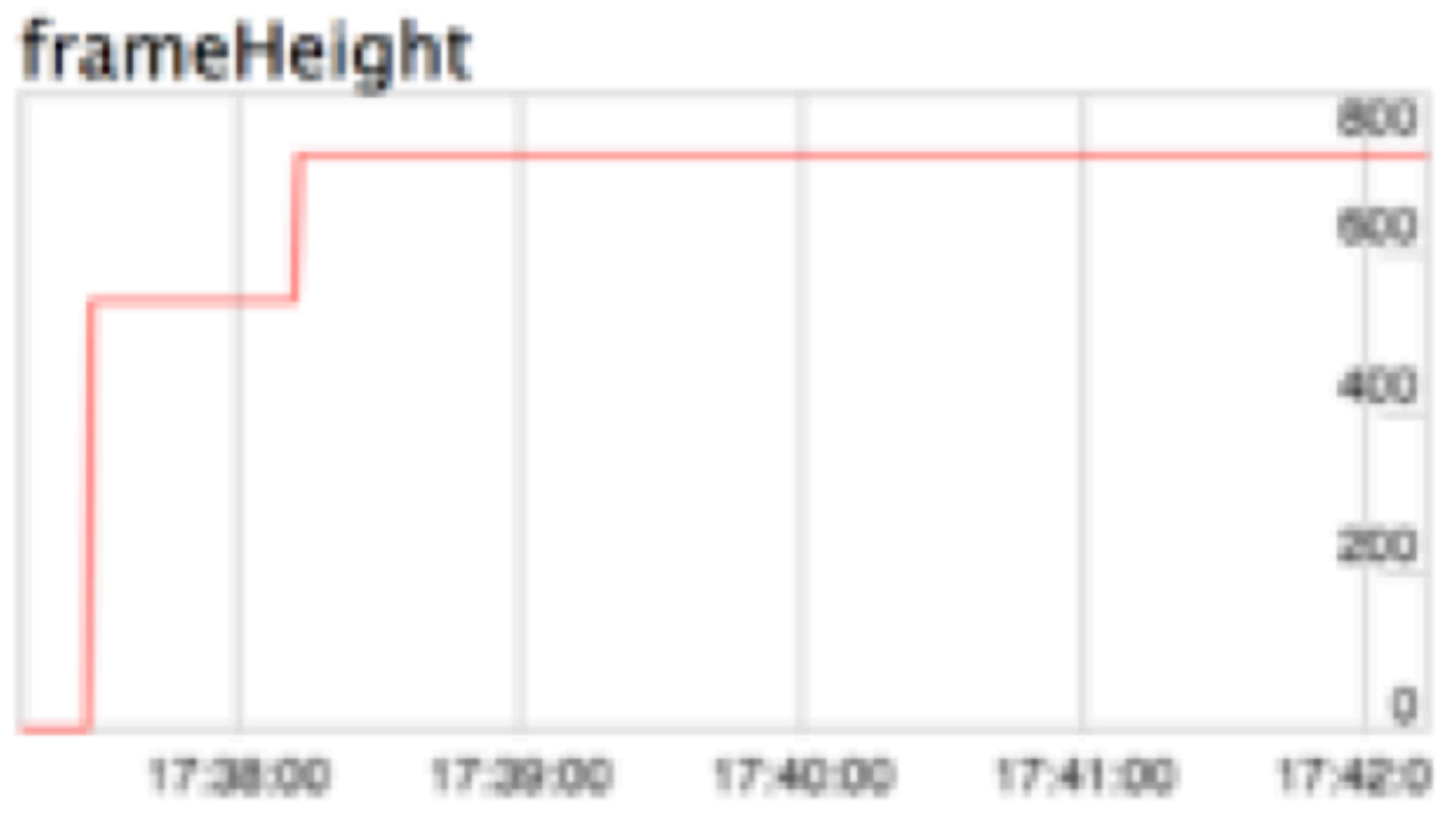


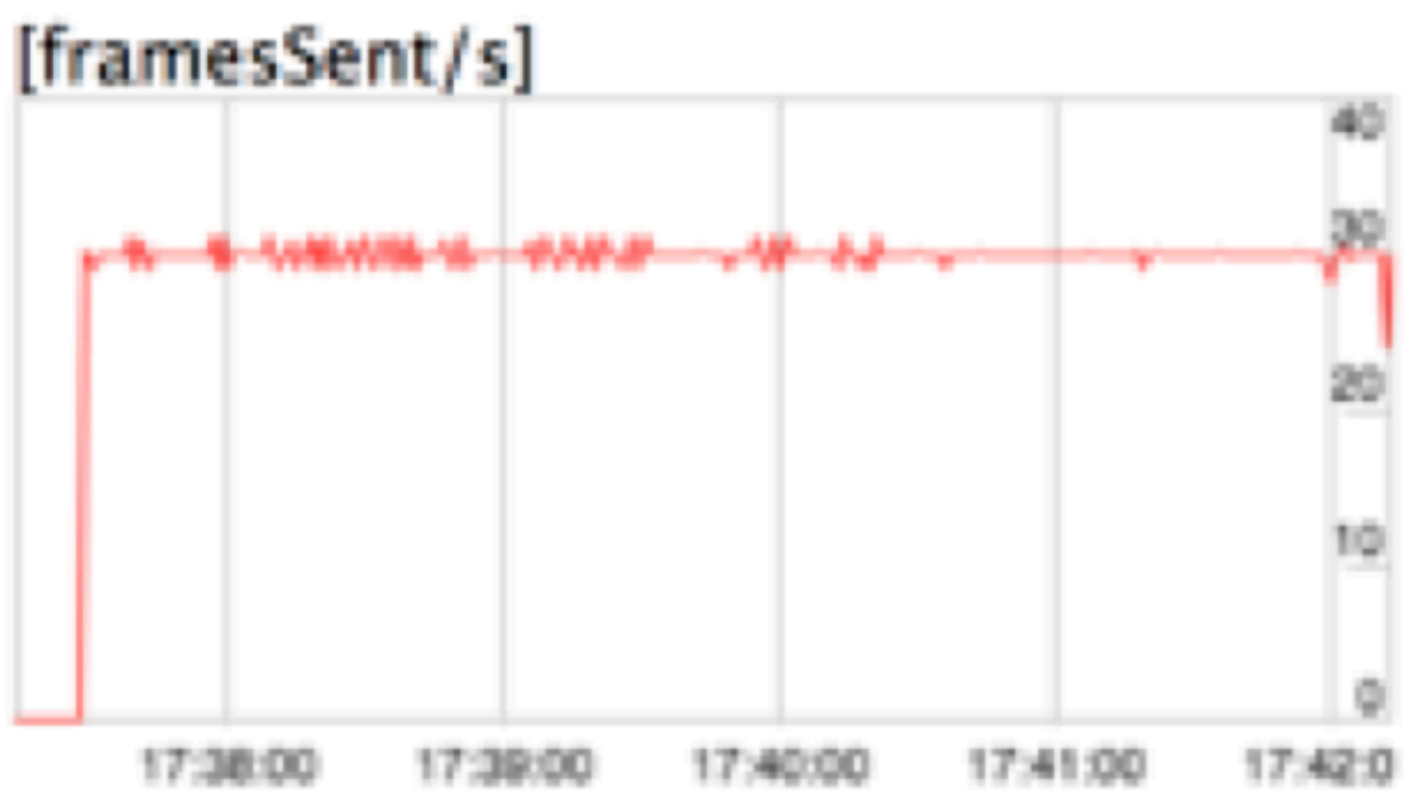




B —> A







Scenario C.2: Remote Session across Atlantic

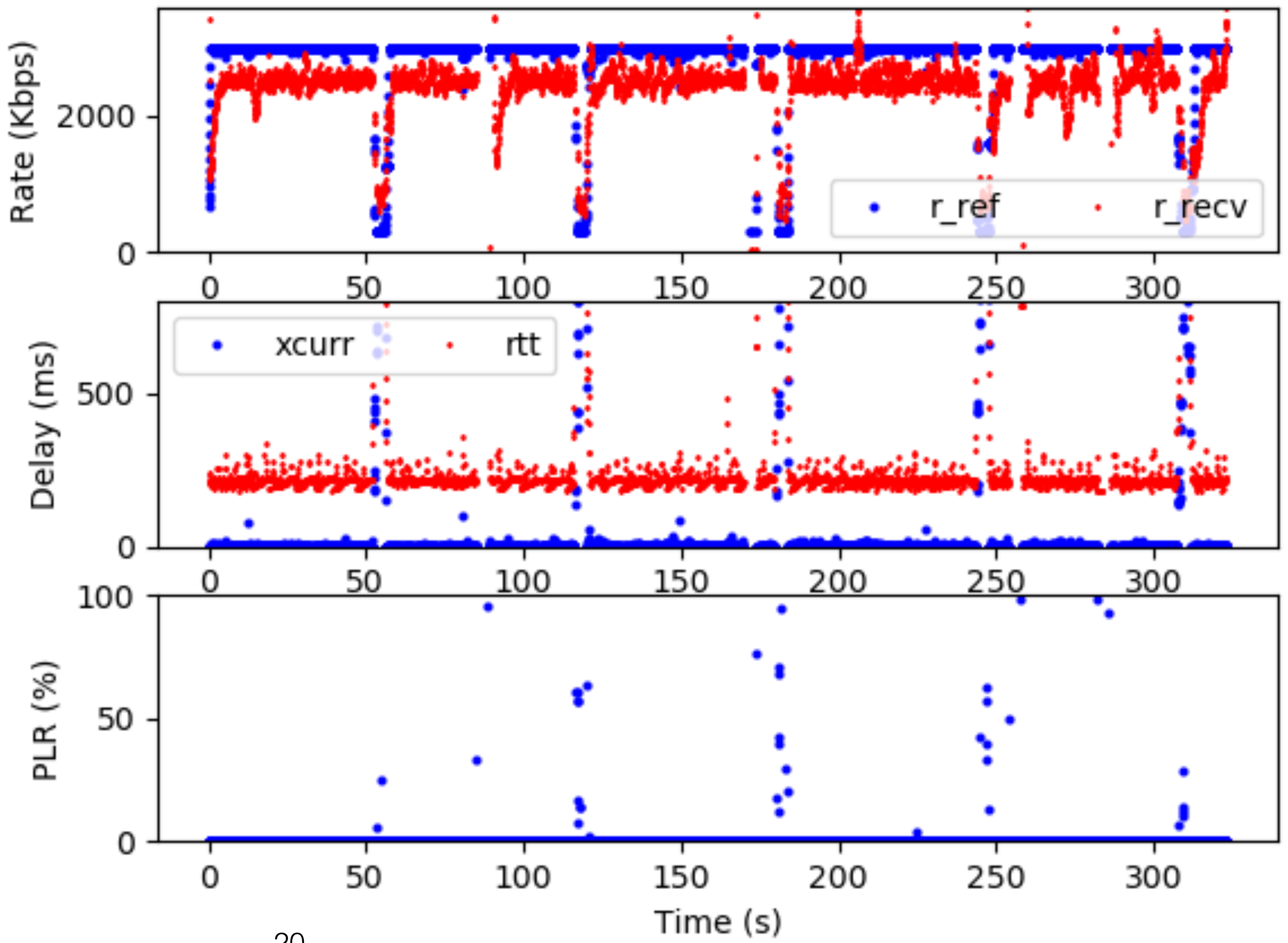
Client A:

- Location: Austin, Texas, USA
- Connection: enterprise Wi-Fi

Client B:

- Location: Lausanne, Switzerland
- Connection: enterprise Wi-Fi

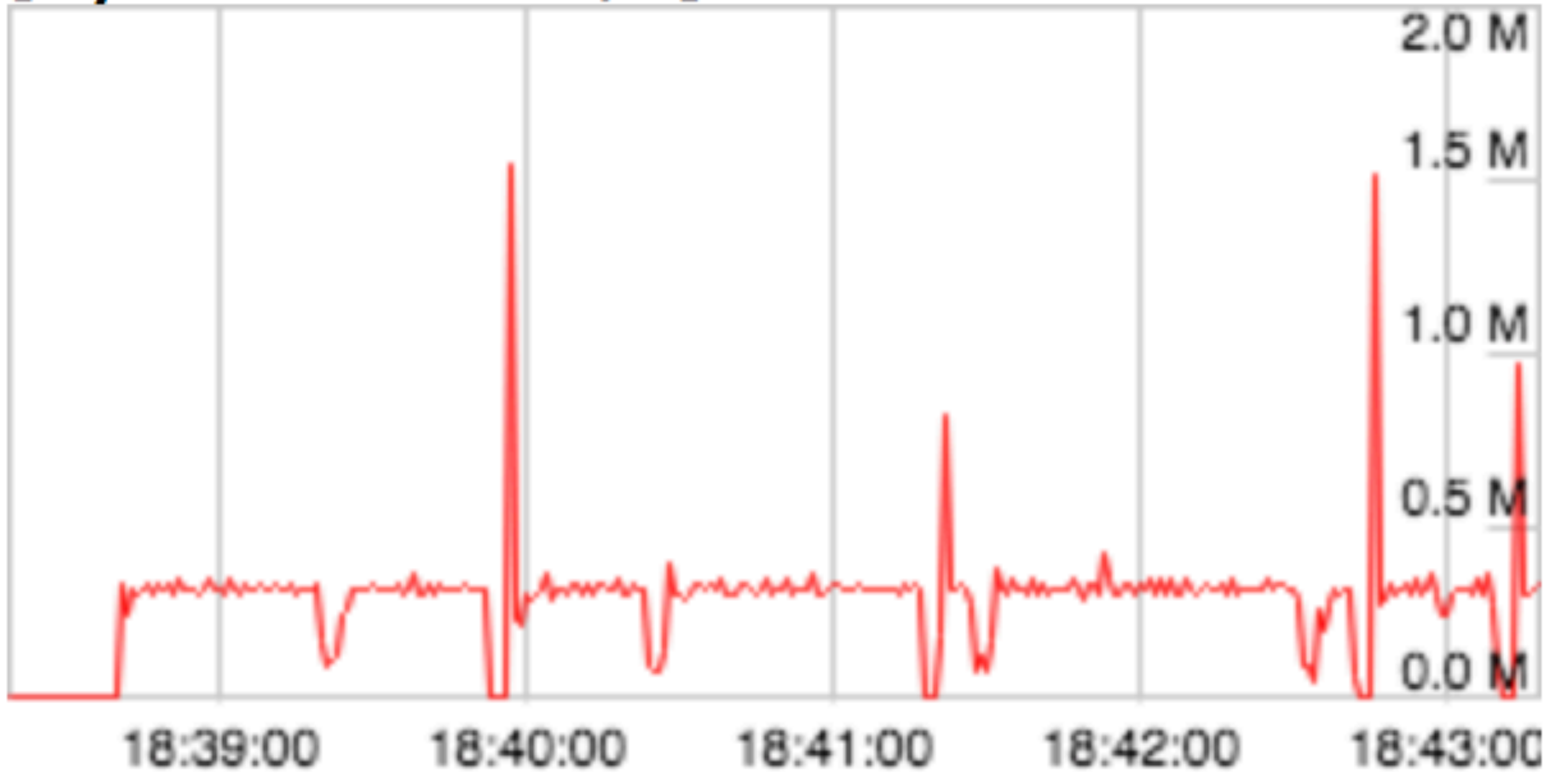
- Path Characteristics:
  - Baseline RTT: ~170ms
  - Max RTT: ~ 4.4s
  - Overall loss ratio: 1.5%



Scenario C.2: Screenshots from Chrome

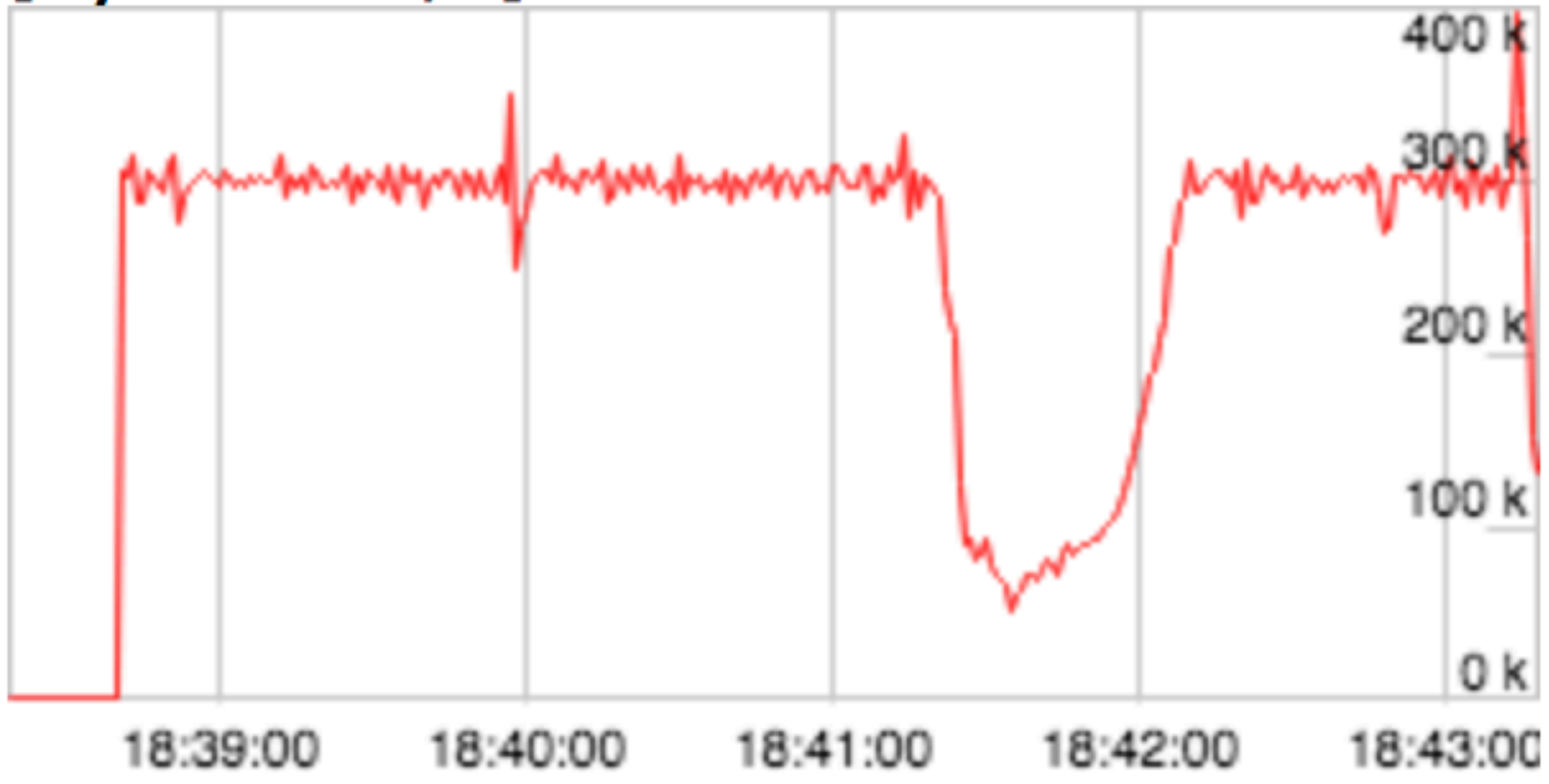
A —> B

[bytesReceived/s]



B —> A

[bytesSent/s]



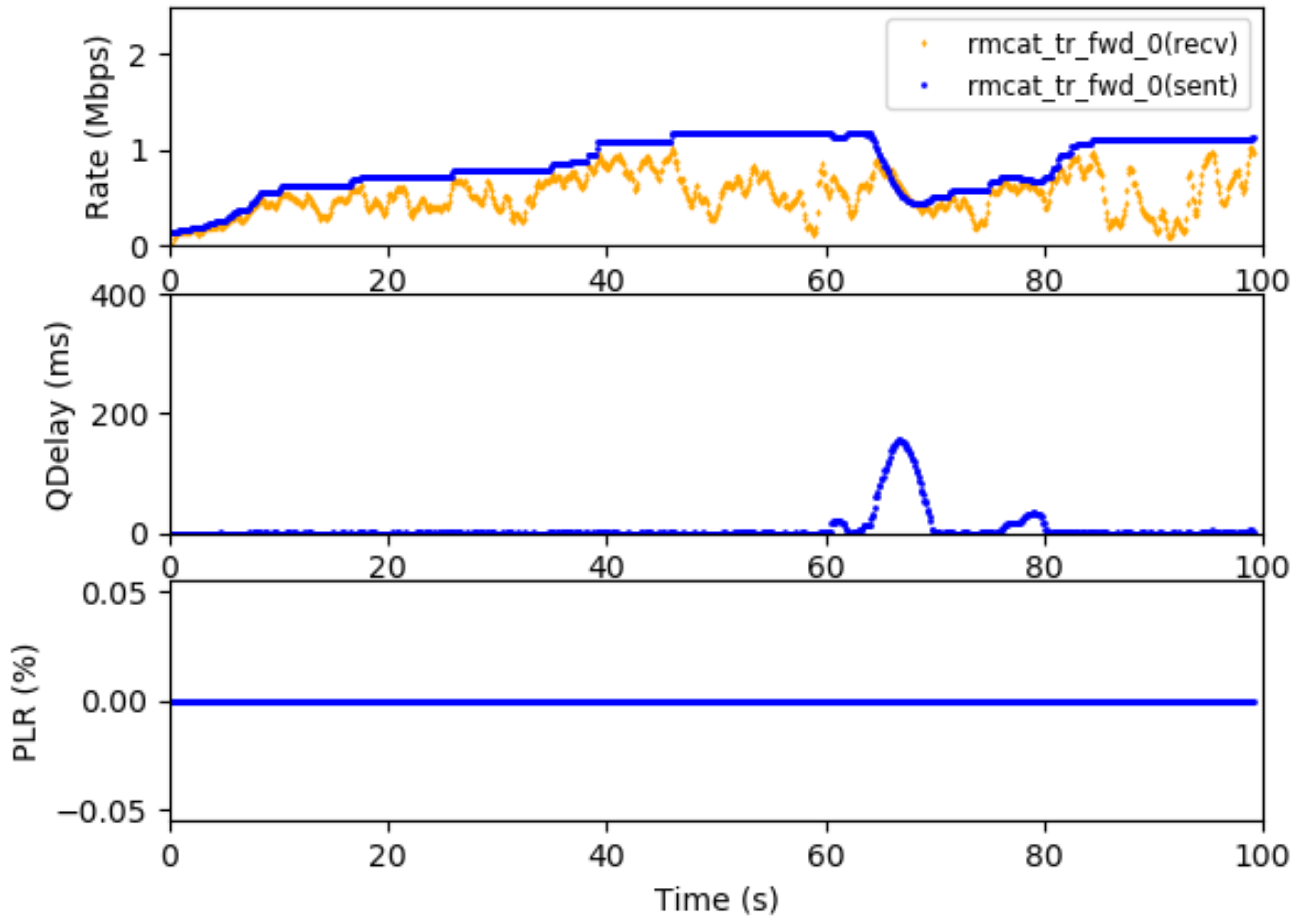
Observations and Next Steps

- Performance over connections with sufficient bandwidth:
    - Fast ramp up to maximum allowed rate, typically within a few seconds
    - Sending rate dips briefly due to occasional delay spikes but recovers quickly
    - Switches between accelerated ramp-up and gradual update modes due to frequent delay spikes
    - Remains stable over a wide range of path RTTs (e.g., up to 180ms in baseline RTT)
- Pending further investigations:
    - Performance over bandwidth-limited connections
    - Coexistence with TCP-like background traffic

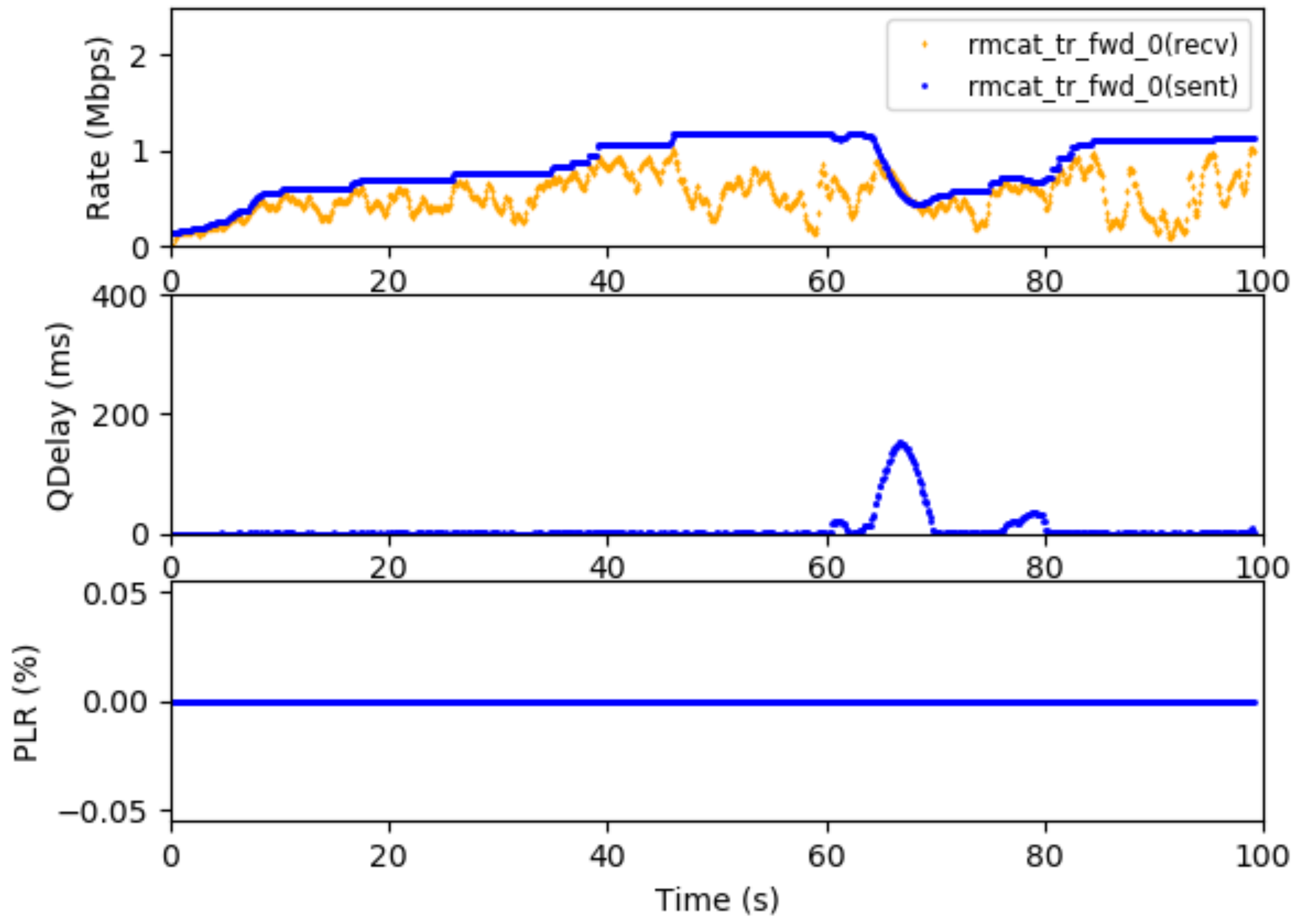
Backup Slides

Influence of Rate Shaping Buffer: TC5.1

BETA_S = 0.0 | BETA_V = 0.0

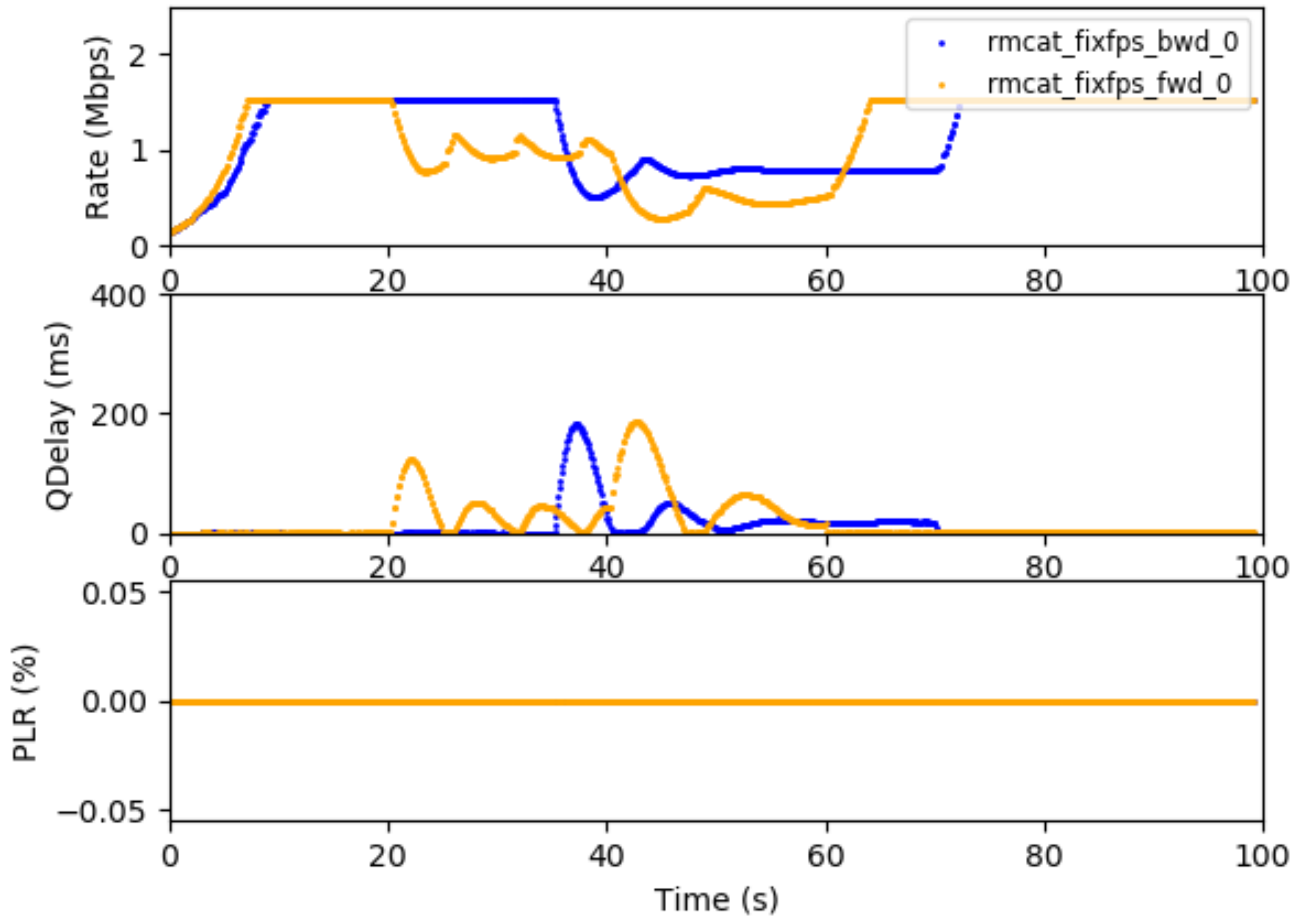


BETA_S = 0.1 | BETA_V = 0.1

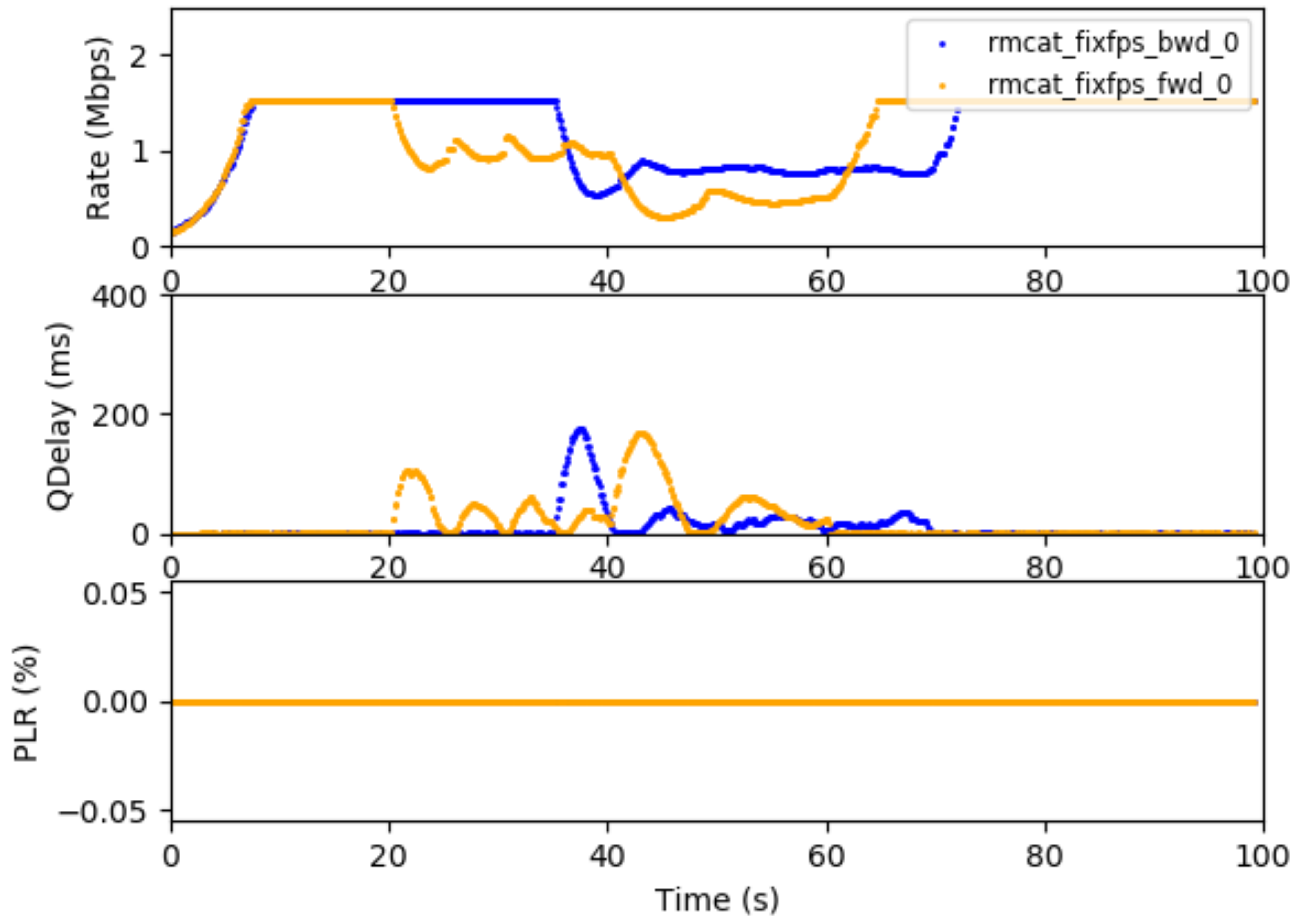


Influence of Rate Shaping Buffer: TC5.3

BETA_S = 0.0 I BETA_V = 0.0

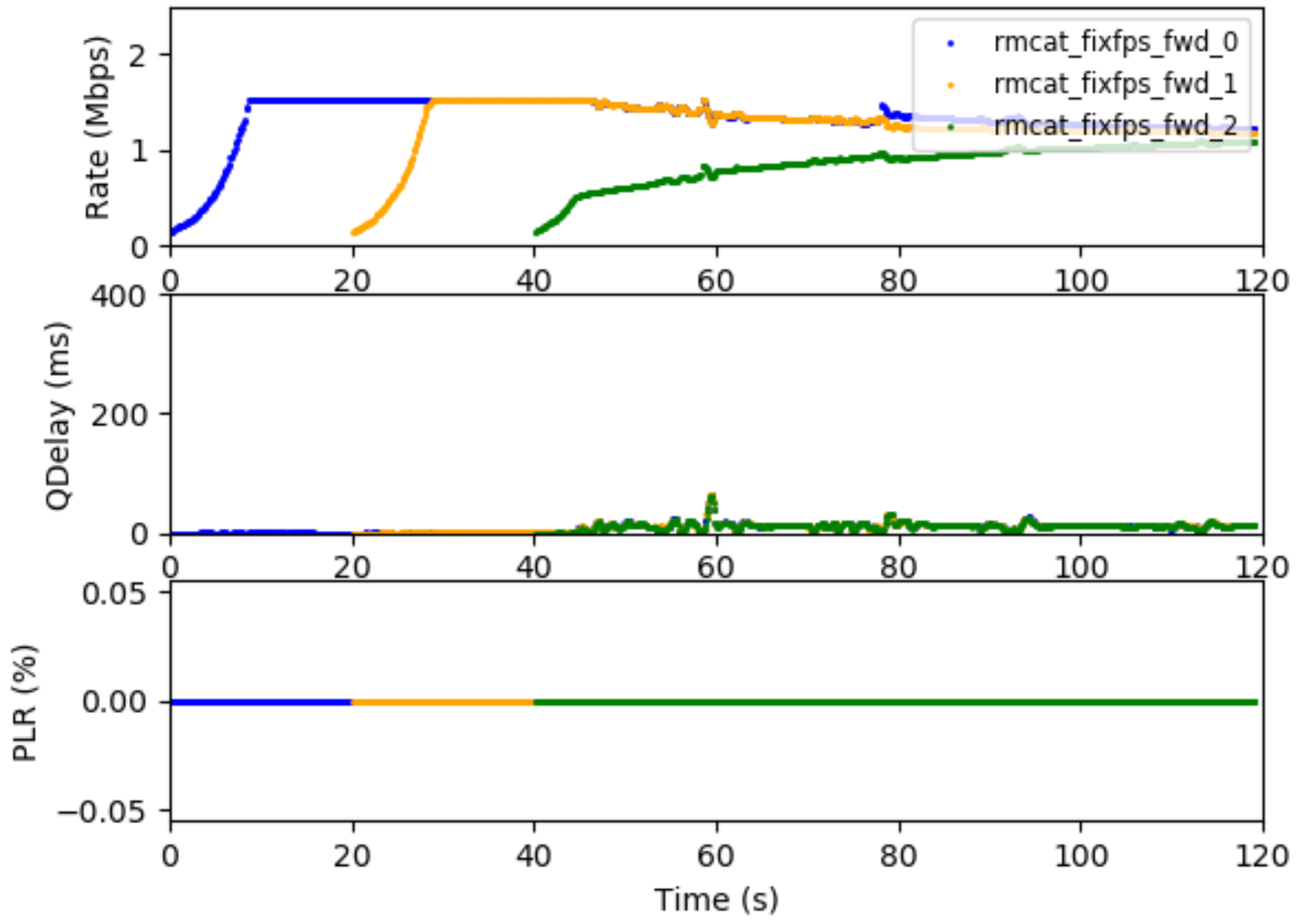


BETA_S = 0.1 I BETA_V = 0.1

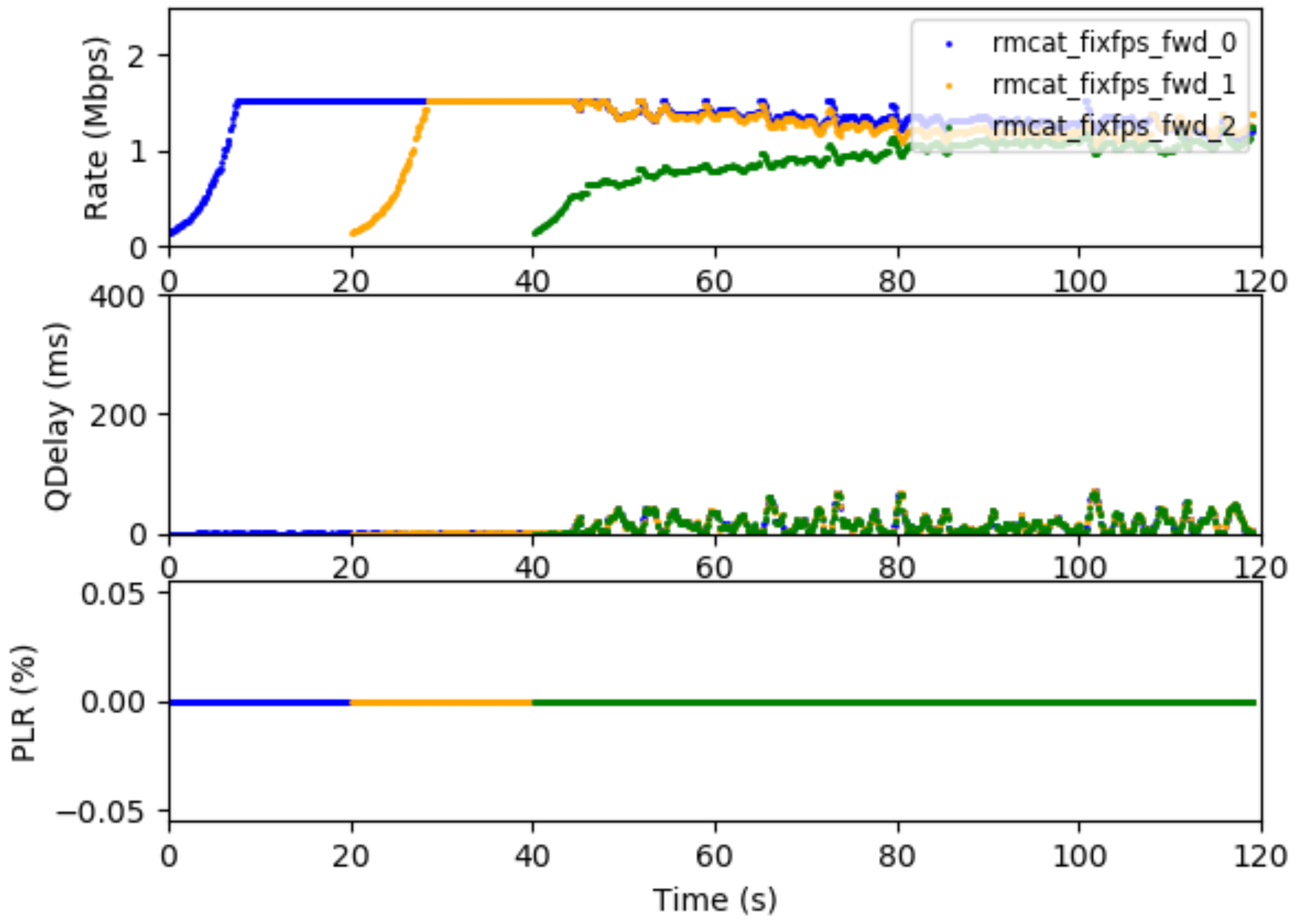


Influence of Rate Shaping Buffer: TC5.4

BETA_S = 0.0 | BETA_V = 0.0

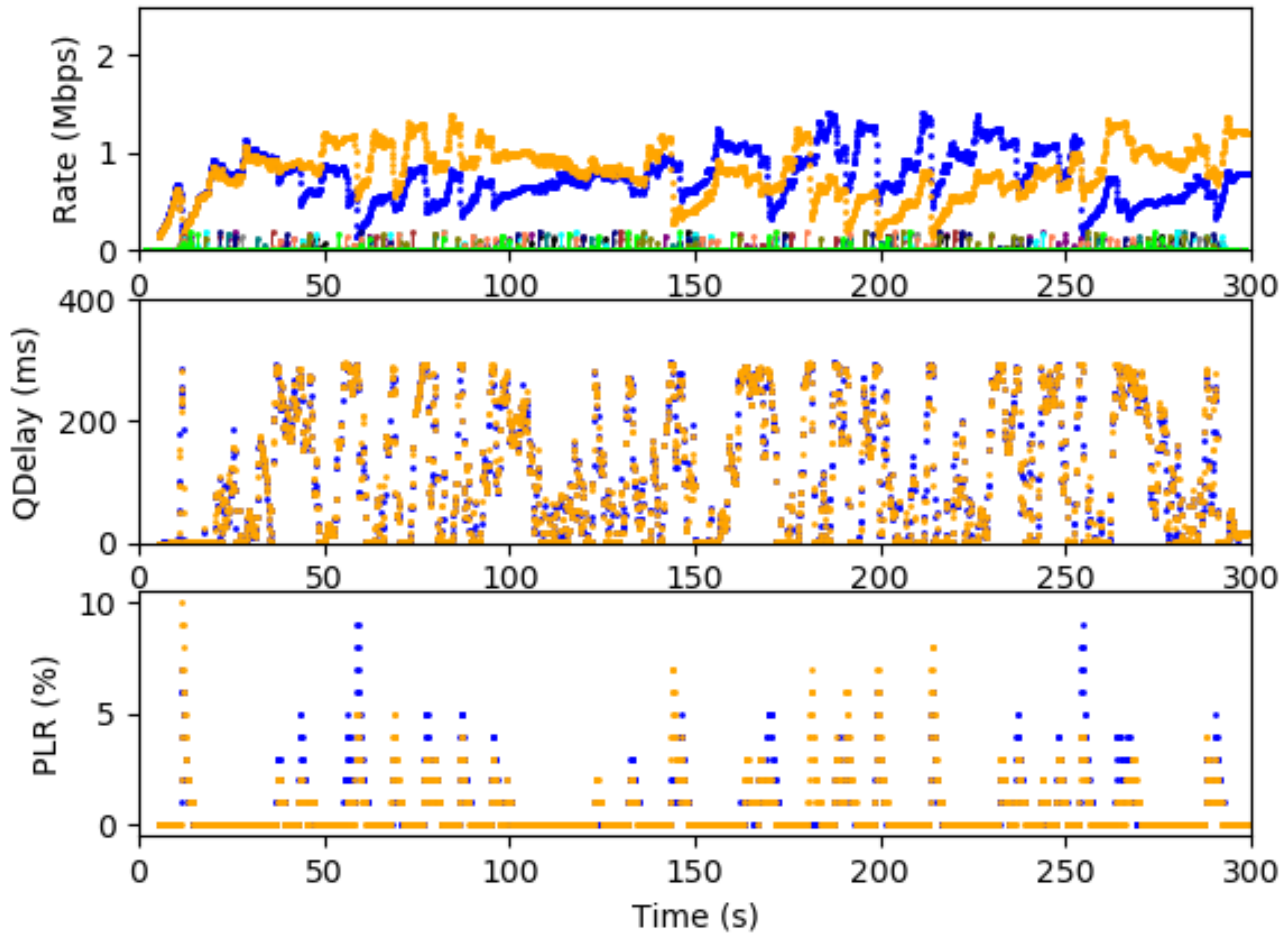


BETA_S = 0.1 | BETA_V = 0.1

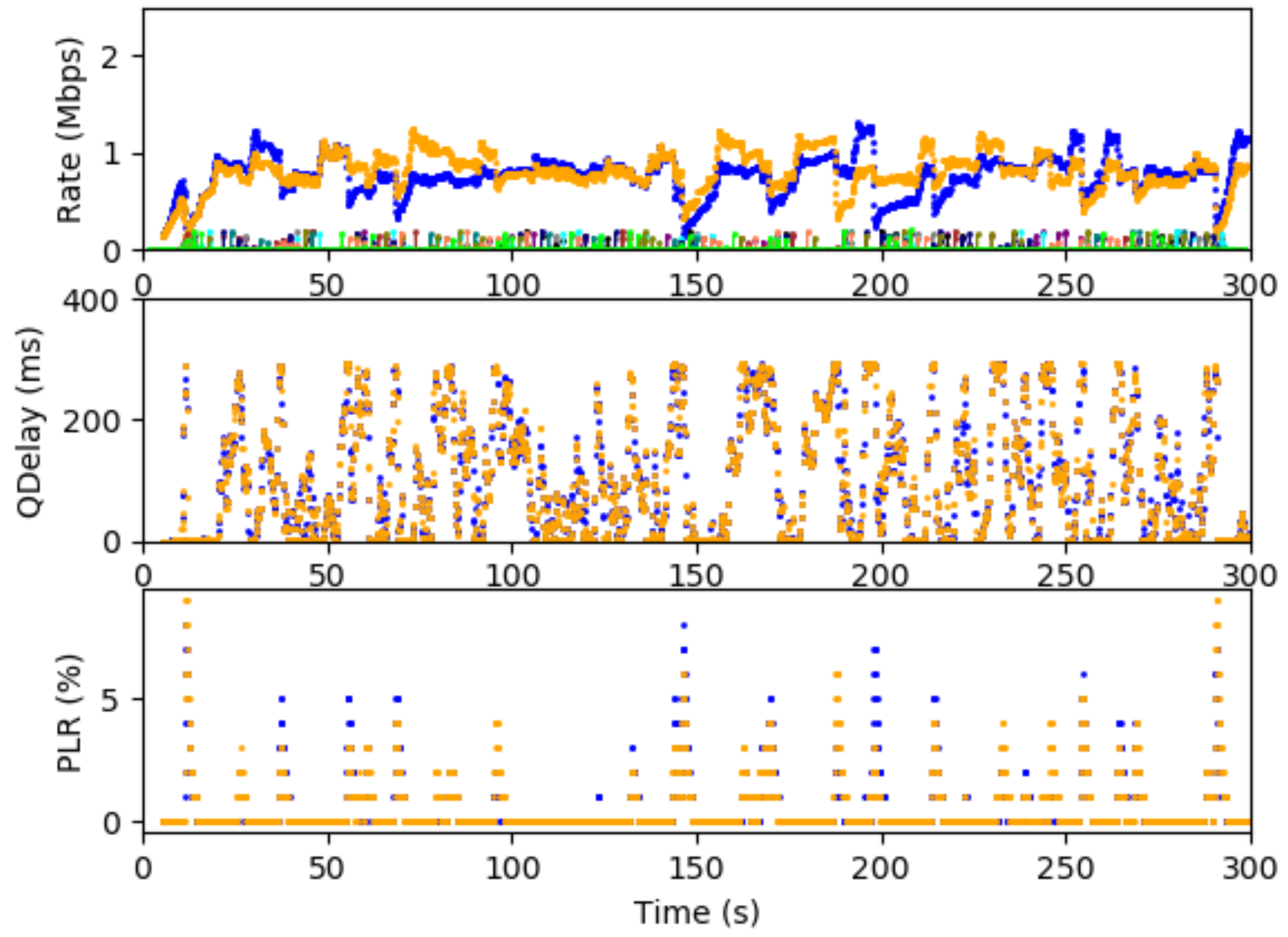


Influence of Rate Shaping Buffer: TC5.7

BETA_S = 0.0 | BETA_V = 0.0

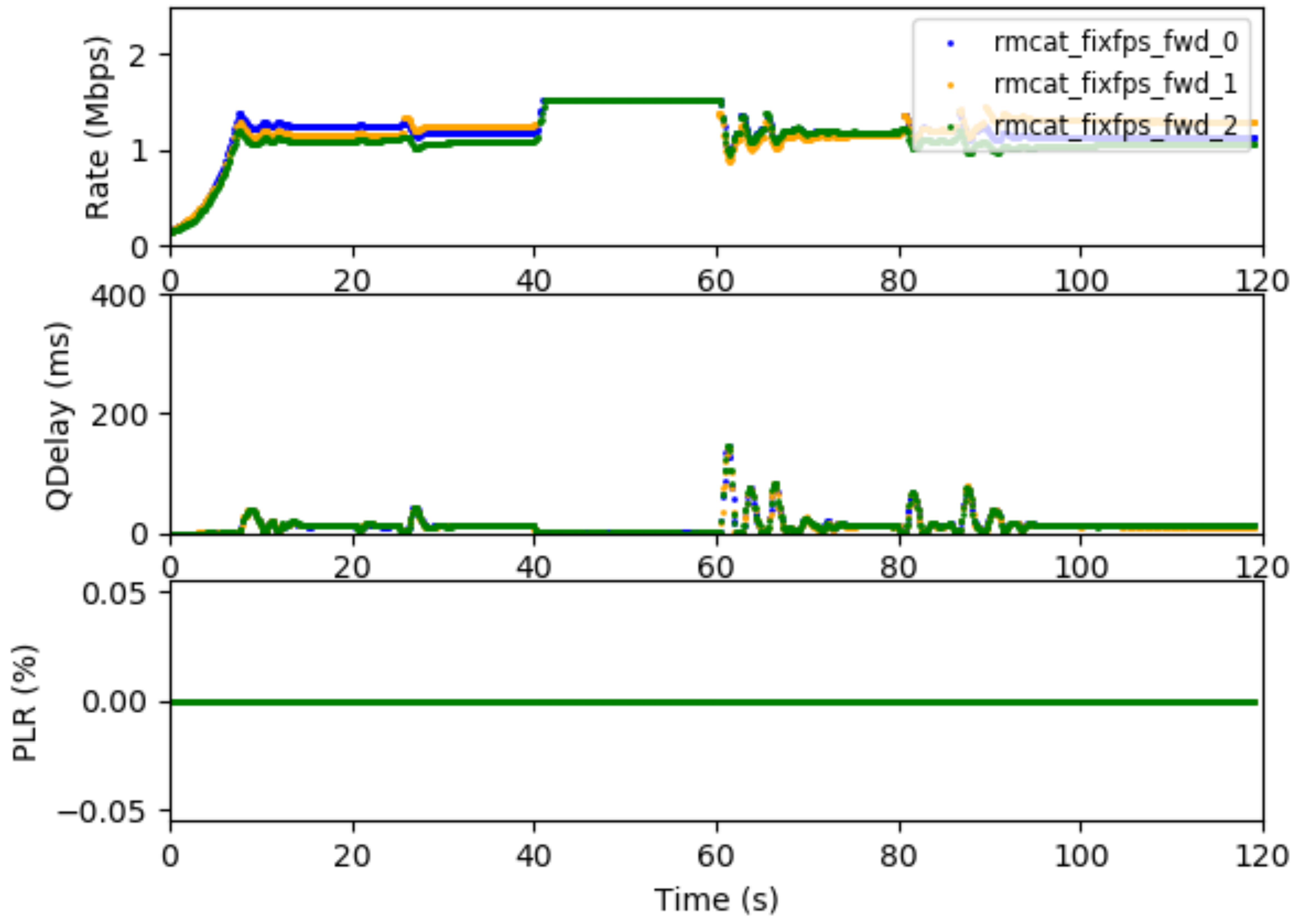


BETA_S = 0.1 | BETA_V = 0.1

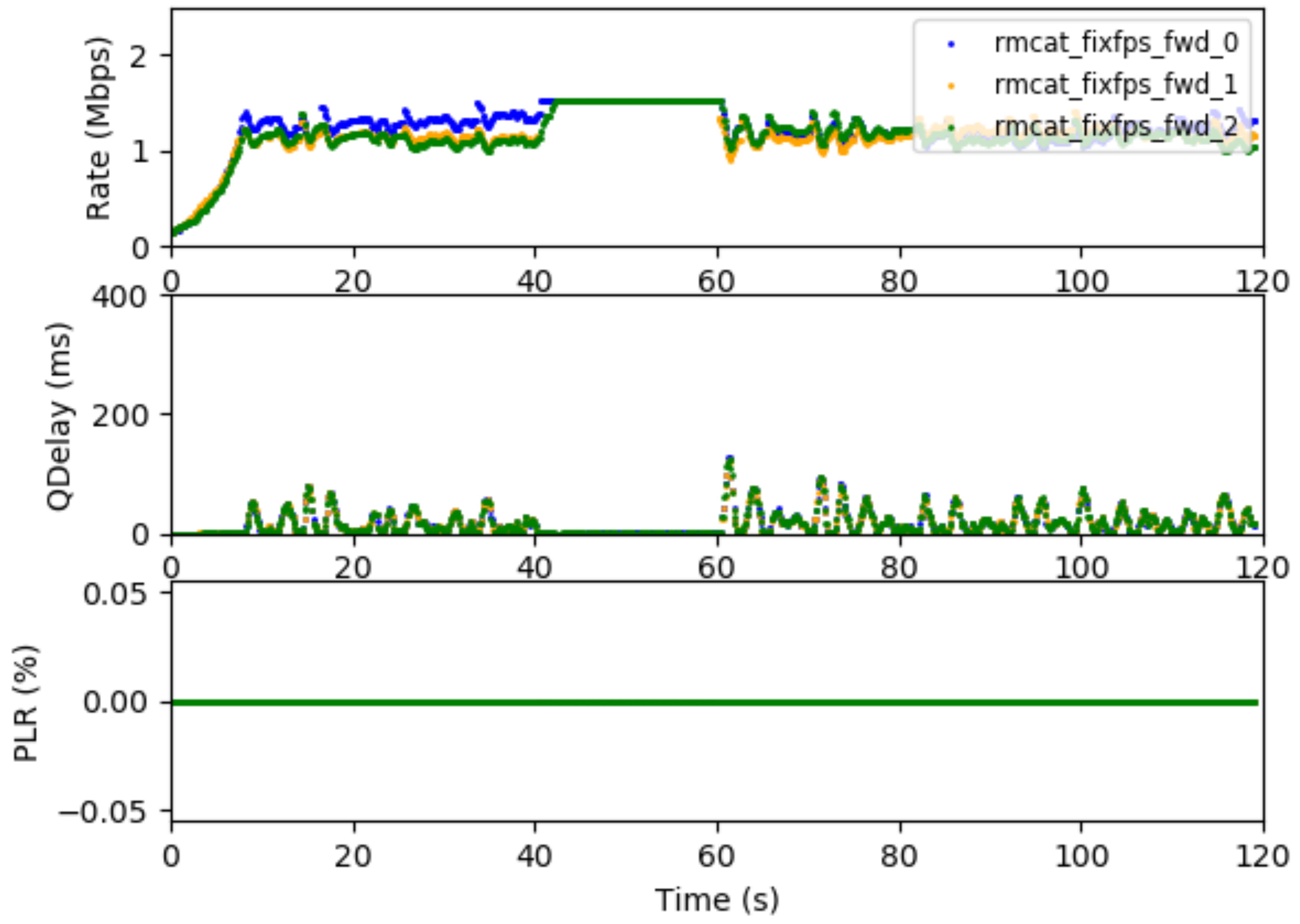


Influence of Rate Shaping Buffer: TC5.8

BETA_S = 0.0 | BETA_V = 0.0



BETA_S = 0.1 | BETA_V = 0.1


Rate (Mbps)
QDelay (ms)
PLR (%)
Time (s)
rmcat_fixfps_fwd_0
rmcat_fixfps_fwd_1
rmcat_fixfps_fwd_2