Insights into BBRv3's Performance and Behavior by Experimental Evaluation

Roland Bless[1)], Lukas Lihotzki[2)], Martina Zitterbart[1)]

[1)] Institute of Telematics, KIT

[2)] former Master student at KIT

Experimental Evaluation – Our Contributions

- Real hardware (no emulation, no virtualization)
- Bottleneck data rates 100 Mbit/s – 10 Gbit/s
- Traced BBR internal variables → root cause analysis

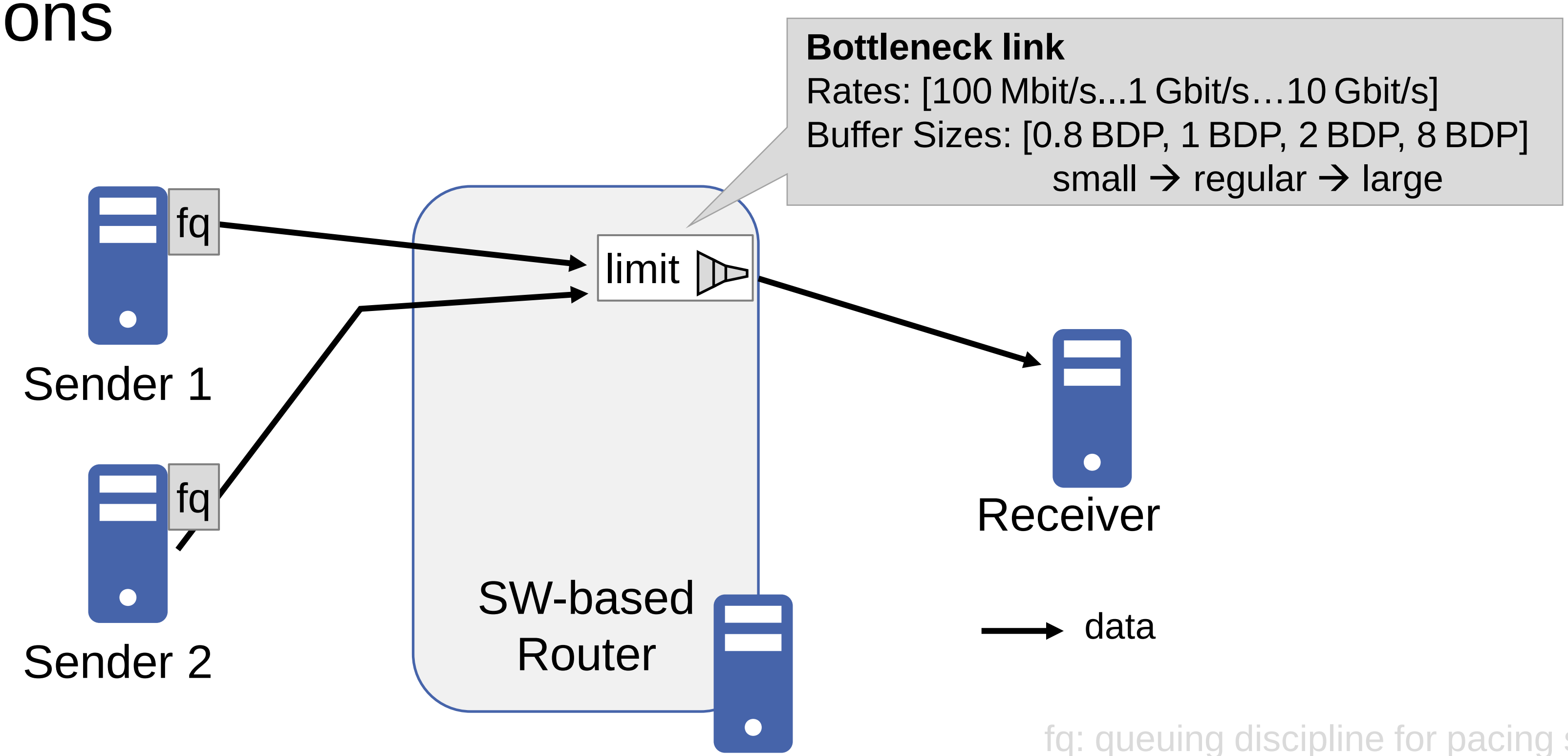
1) Self-Induced Queuing Delay — This talk

2) Delay-Jitter — See paper

3) Short-flows using Real-World Traffic — This talk

Testbed Setup

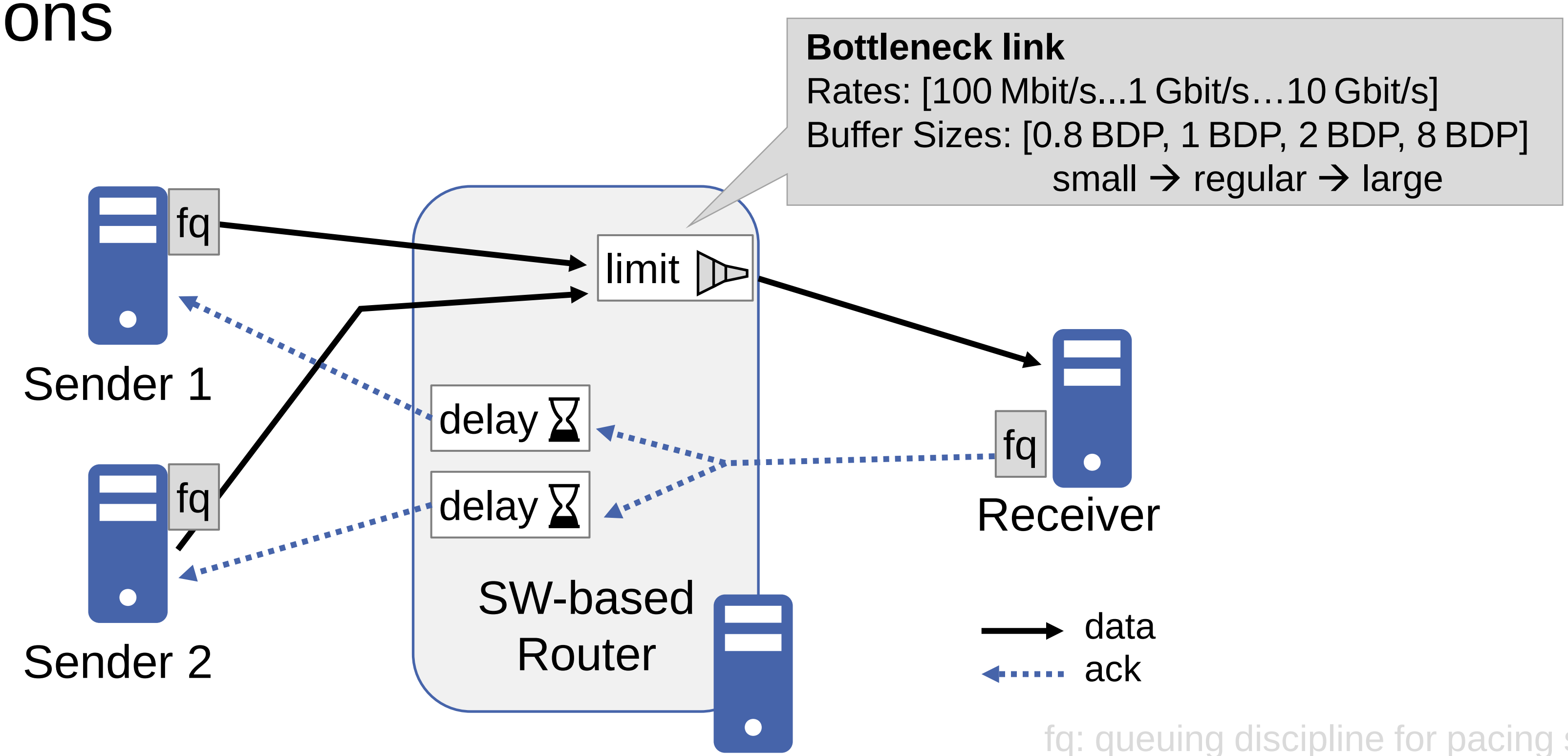
- Linux-based testbed with 100 Mbit/s – 10 Gbit/s
- Using Google's BBRv3 implementation
- 60s runs, 30 repetitions



fq: queuing discipline for pacing support

Testbed Setup

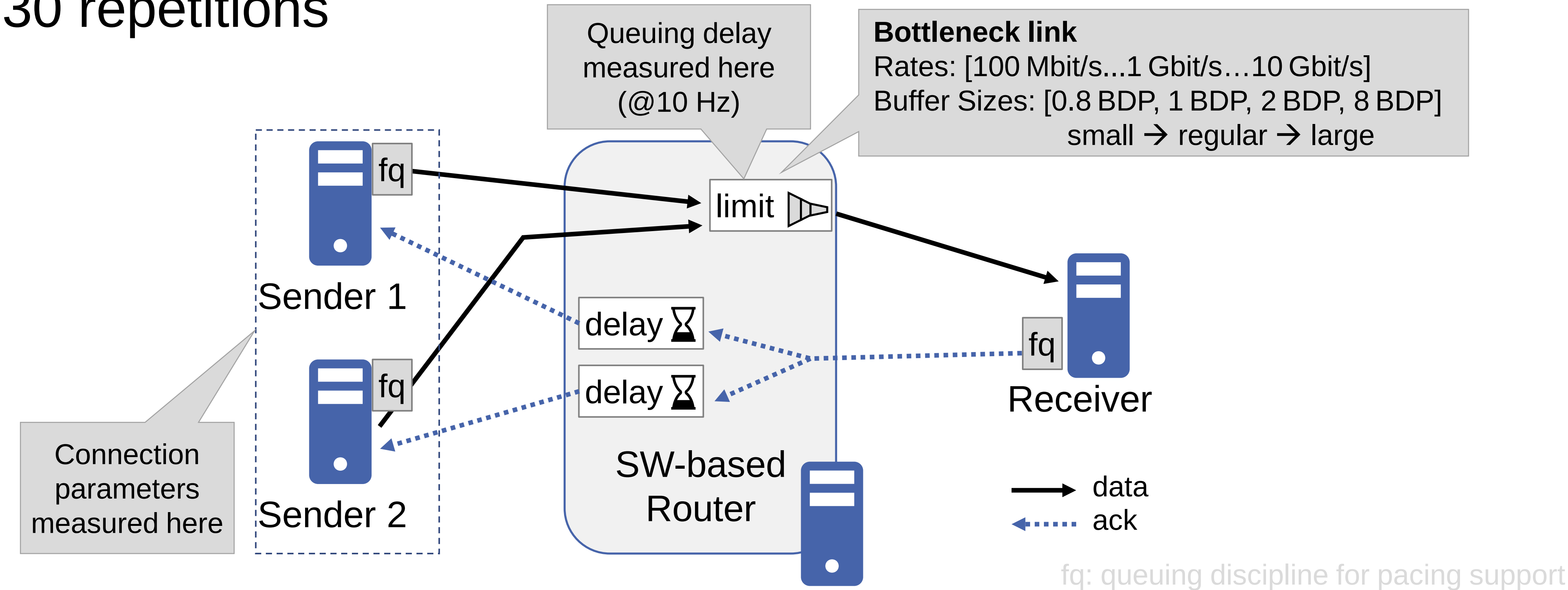
- Linux-based testbed with 100 Mbit/s – 10 Gbit/s
- Using Google's BBRv3 implementation
- 60s runs, 30 repetitions



fq: queuing discipline for pacing support

Testbed Setup

- Linux-based testbed with 100 Mbit/s – 10 Gbit/s
- Using Google's BBRv3 implementation
- 60s runs, 30 repetitions



Experimental Results

Karlsruhe Institute of Technology

1) Self-Induced Queuing Delay

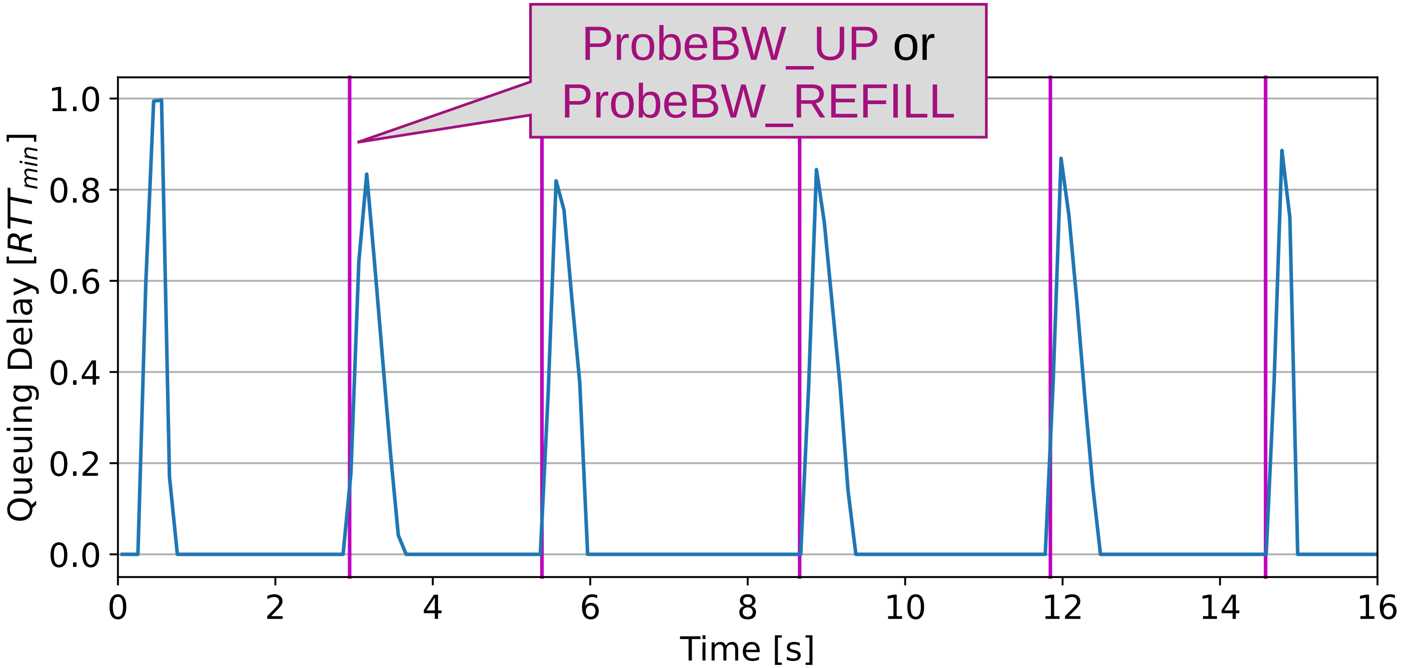
NOW

(traffic **not** limited by application)

2) Delay-Jitter

3) Short-flows using Real-World Traffic

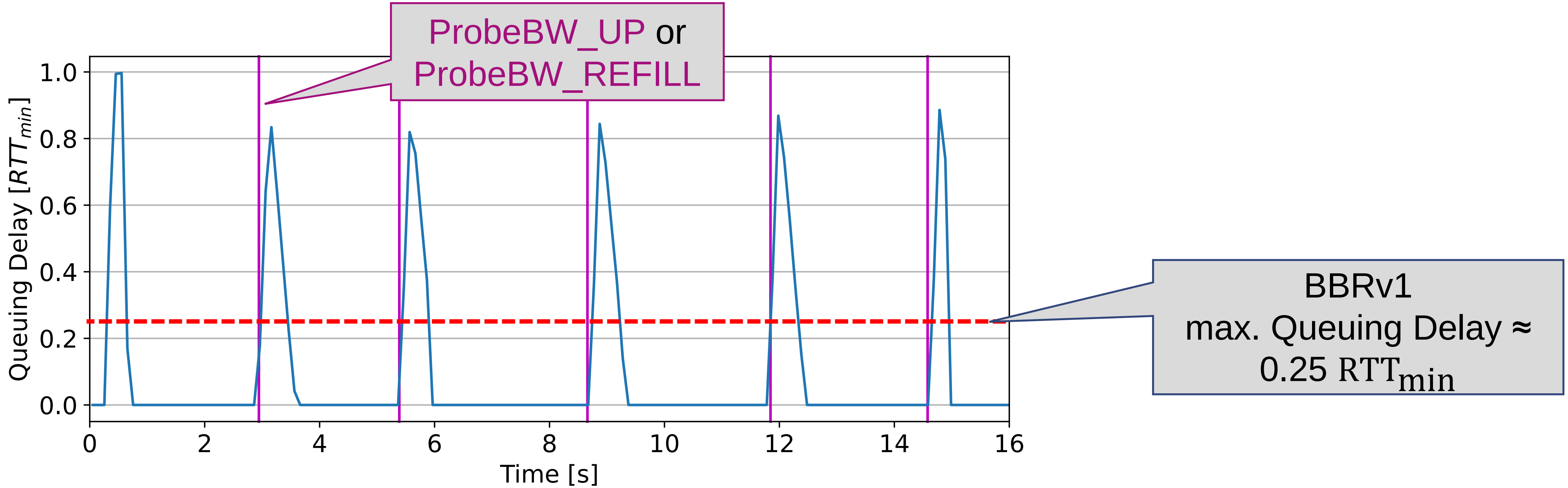
Self-induced Queuing Delay – Single Flow



100 Mbit/s, $\mathrm{RTT_{min}}$ = 50 ms, buffer size = 8 BDP

- 1st Cause: ProbeBW_UP Phase → estimated bandwidth · 1.25, for 3 RTTs
- Self-induced queuing delay prolongs round-trips

Self-induced Queuing Delay – Single Flow

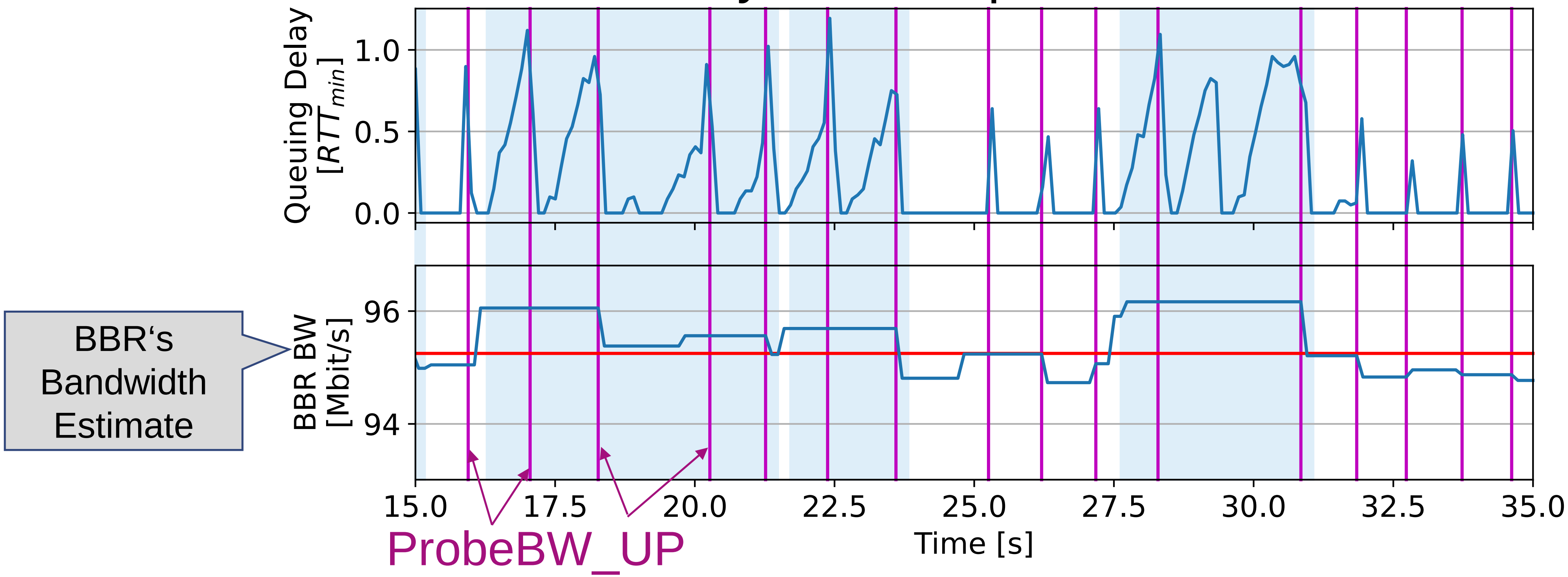


100 Mbit/s, RTT_{min} = 50 ms, buffer size = 8 BDP

- 1st Cause: ProbeBW_UP Phase → estimated bandwidth · 1.25, for 3 RTTs
- Self-induced queuing delay prolongs round-trips
- BBRv3 creates higher peaks – not present to that extent with BBRv1

Self-induced Queuing Delay – Single Flow

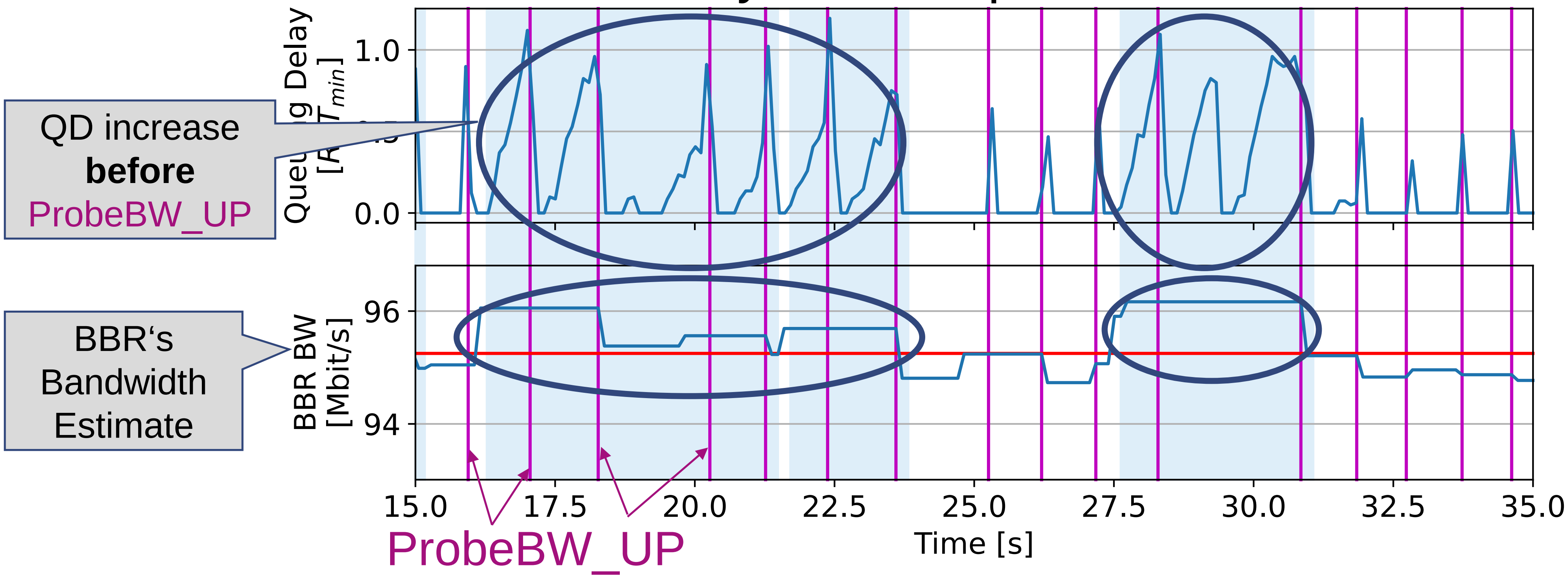
- 2^{nd} Cause: Bandwidth overestimation due to natural jitter
- Measurement noise in delivery rate samples



100 Mbit/s, $\mathrm{RTT}_{\mathrm{min}}$ = **50 ms**, buffer size = 8 BDP

Self-induced Queuing Delay – Single Flow

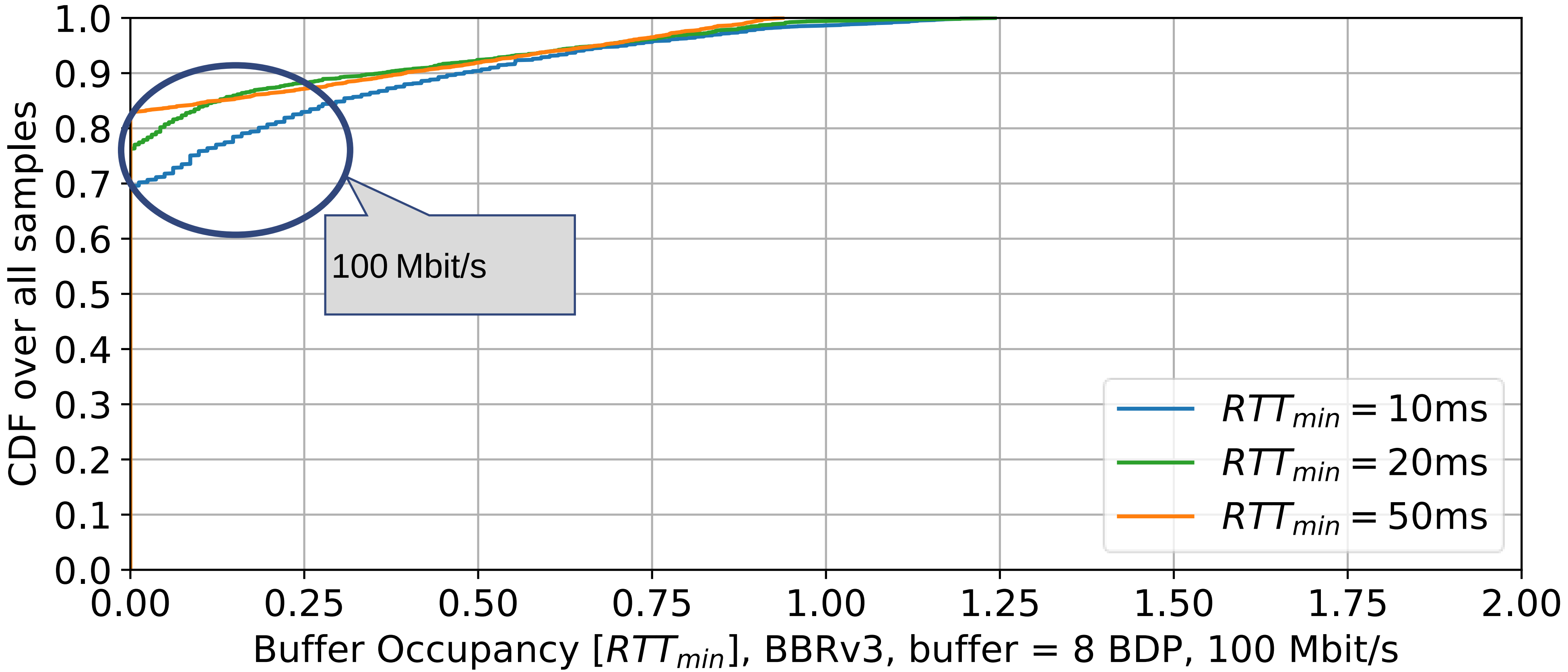
- 2nd Cause: Bandwidth overestimation due to natural jitter
- Measurement noise in delivery rate samples



100 Mbit/s, $\mathrm{RTT}_{\mathrm{min}}$ = **50 ms**, buffer size = 8 BDP

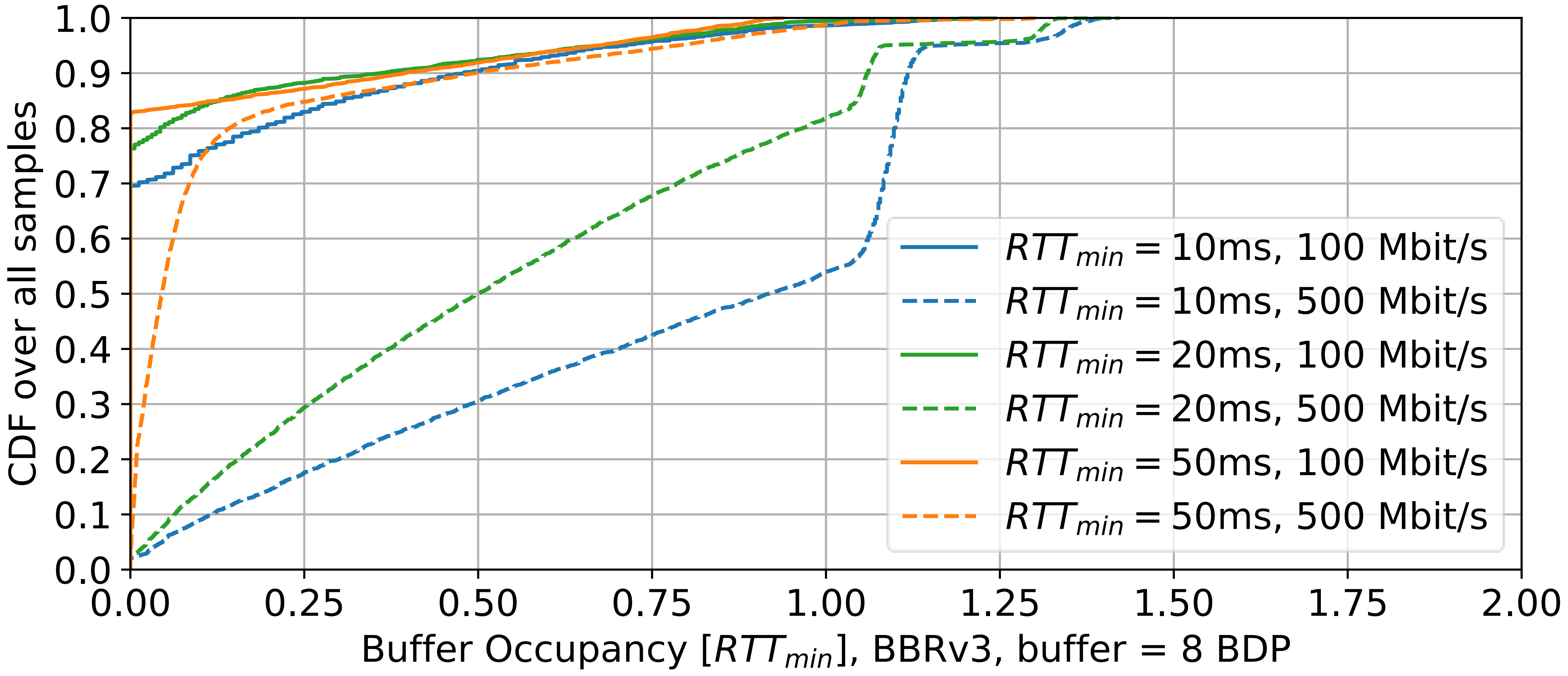
Self-induced Queuing Delay – Single Flow

- It gets worse at higher speeds and smaller $\mathrm{RTT}_{\mathrm{min}}$ values…



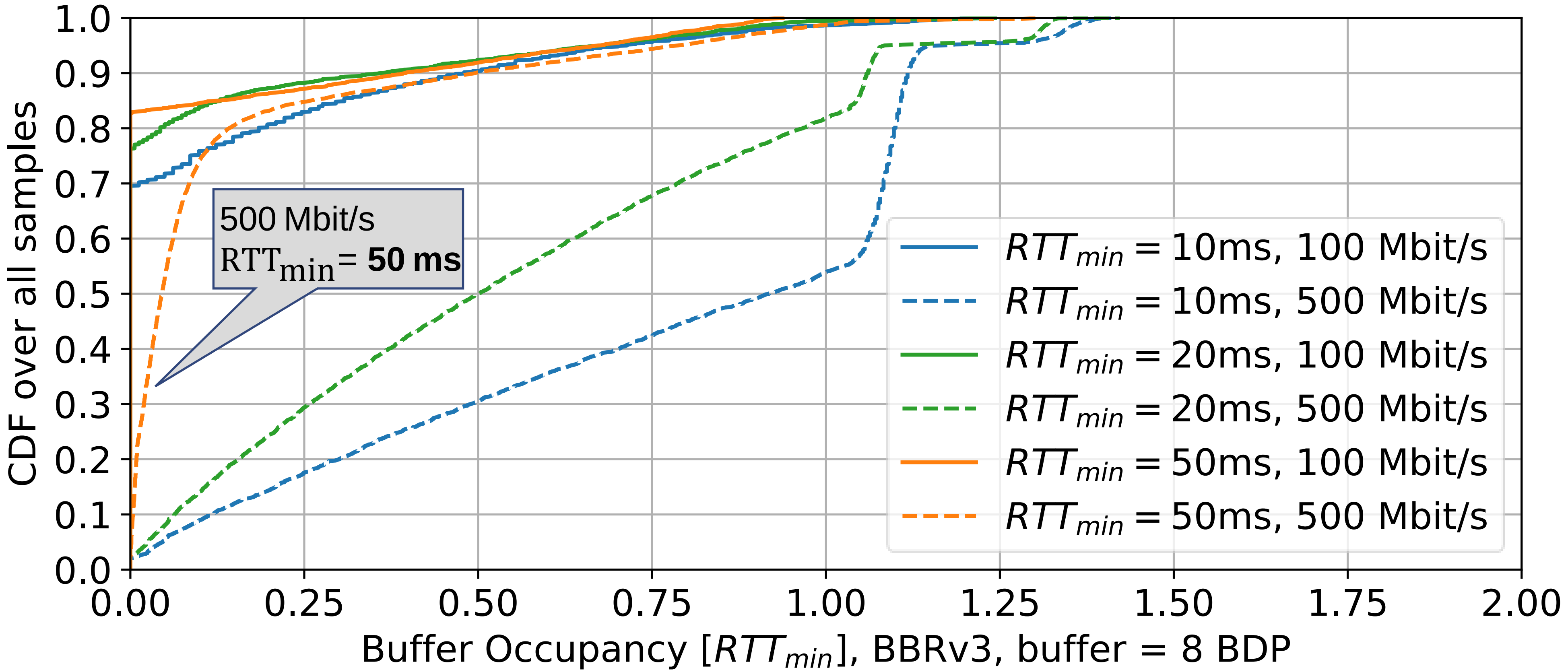
Self-induced Queuing Delay – Single Flow

- It gets worse at higher speeds and smaller $\mathrm{RTT_{min}}$ values…



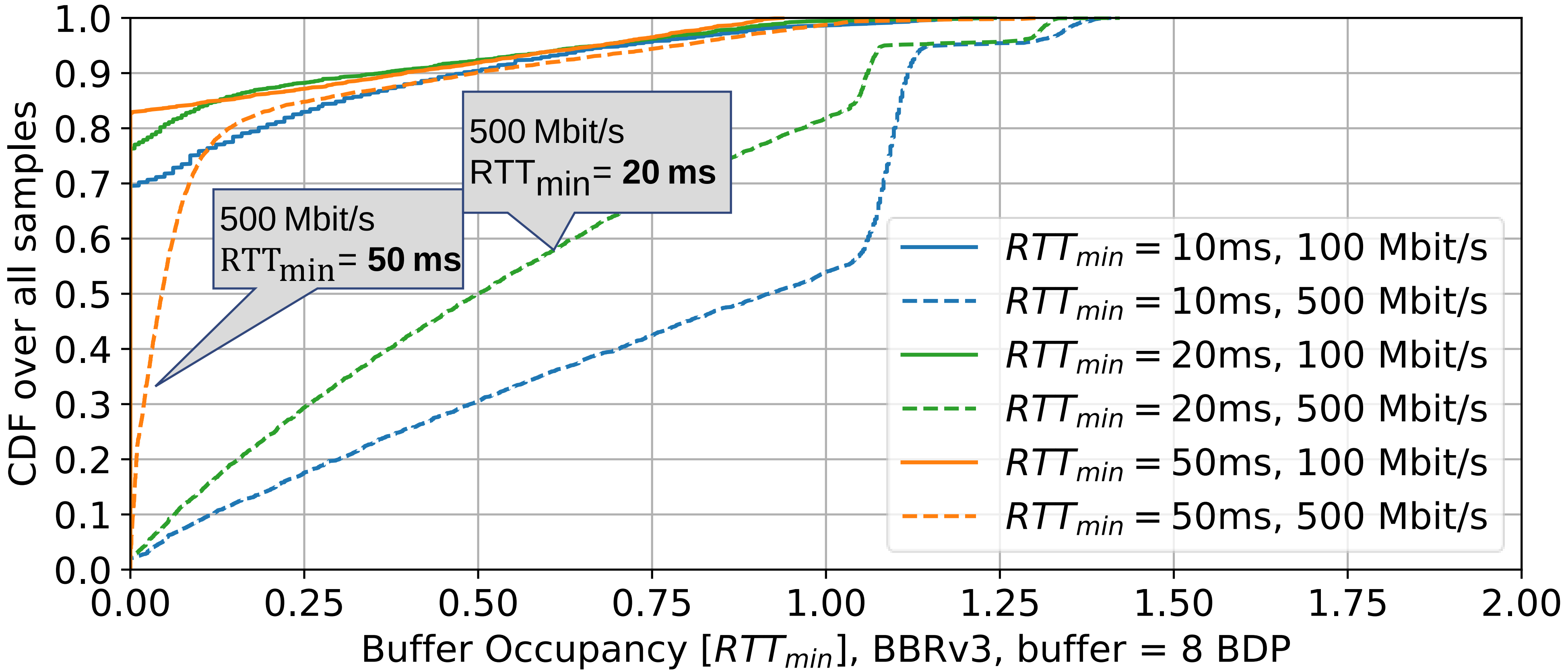
Self-induced Queuing Delay – Single Flow

- It gets worse at higher speeds and smaller $\mathrm{RTT}_{\mathrm{min}}$ values…



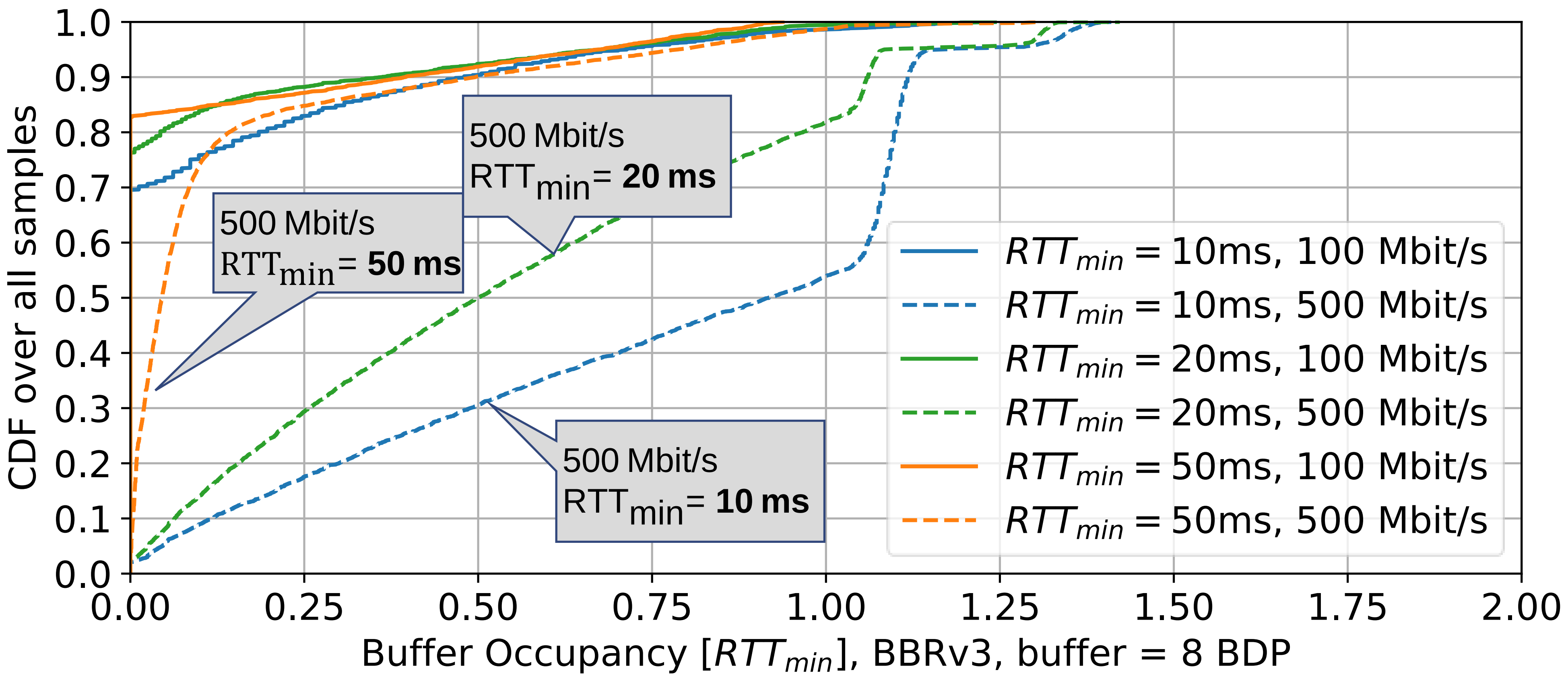
Self-induced Queuing Delay – Single Flow

- It gets worse at higher speeds and smaller $\mathrm{RTT_{min}}$ values…

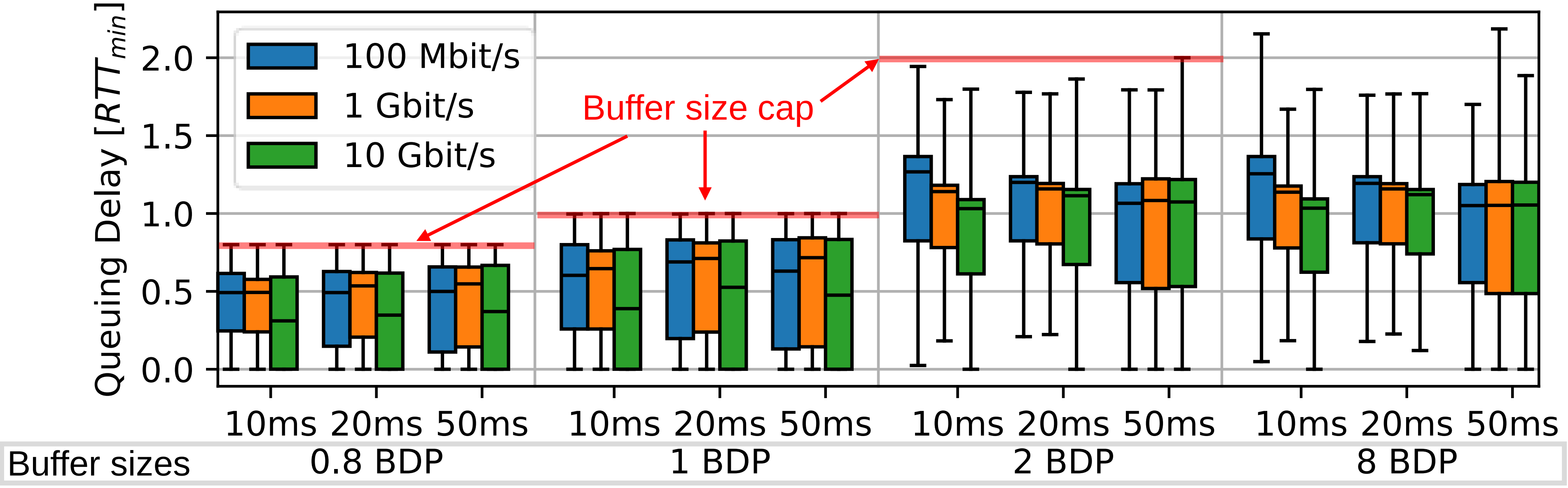


Self-induced Queuing Delay – Single Flow

- It gets worse at higher speeds and smaller $\mathrm{RTT}_{\mathrm{min}}$ values…



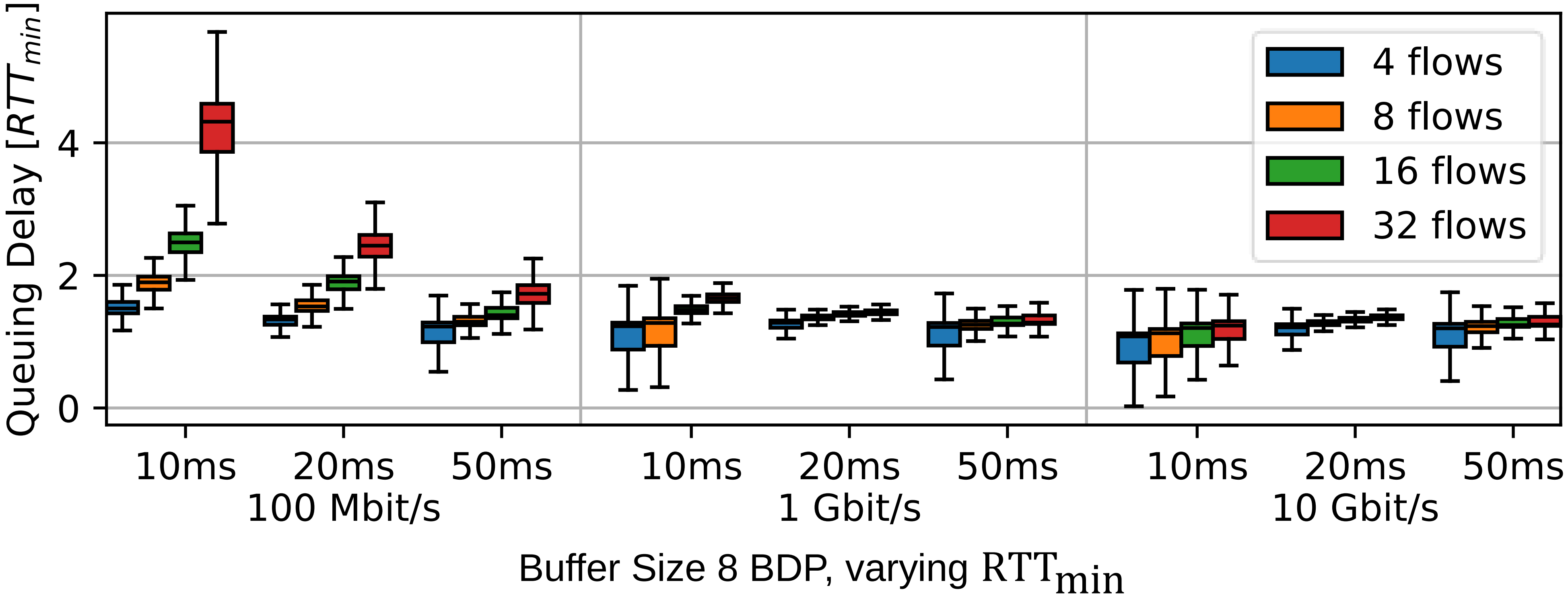
Self-induced Queuing Delay – Two Flows



- Queuing Delay around 1 $\mathrm{RTT}_{\mathrm{min}}$ for >50% of the time for larger buffers
- Cause: overestimation of available bandwidth in ProbeBW_UP (present since BBRv1)

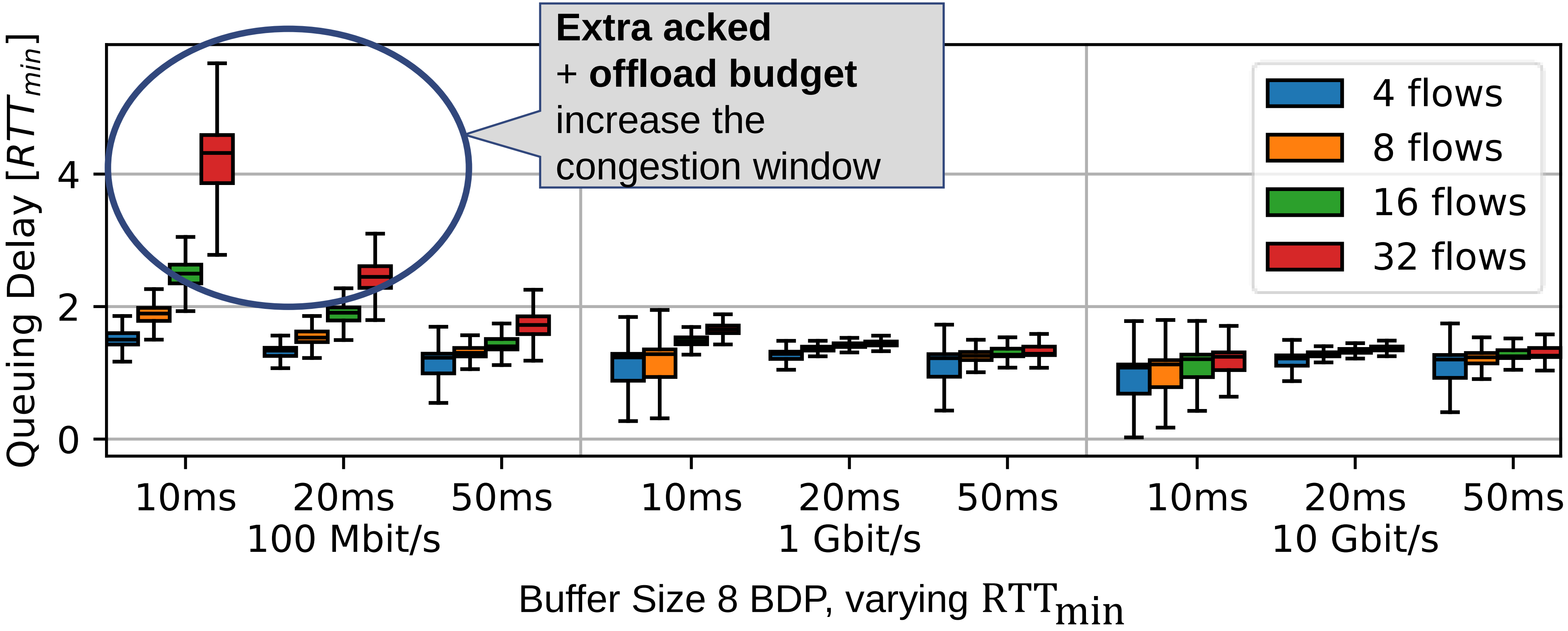
Self-induced Queuing Delay – Multiple Flows

- Gets a bit worse with increasing number of flows



Self-induced Queuing Delay – Multiple Flows

- Gets a bit worse with increasing number of flows



Experimental Results

Karlsruhe Institute of Technology

1) Self-Induced Queuing Delay

2) Delay-Jitter

See paper

3) Short-flows using Real-World Traffic

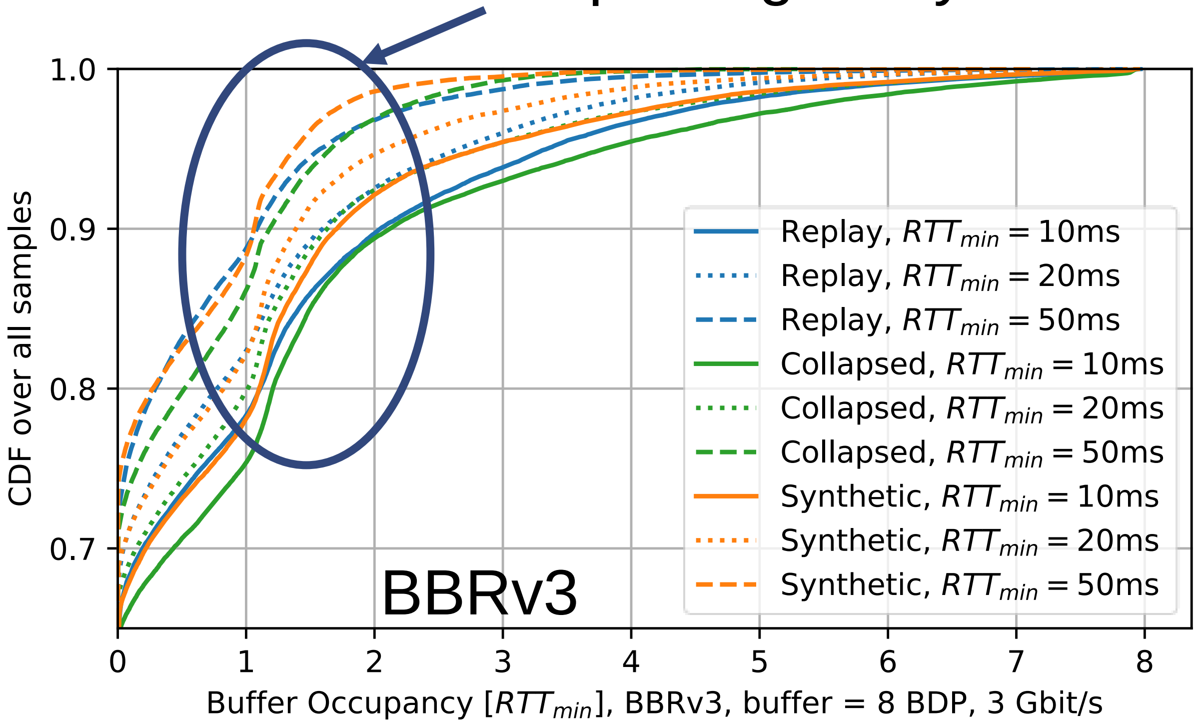
NOW

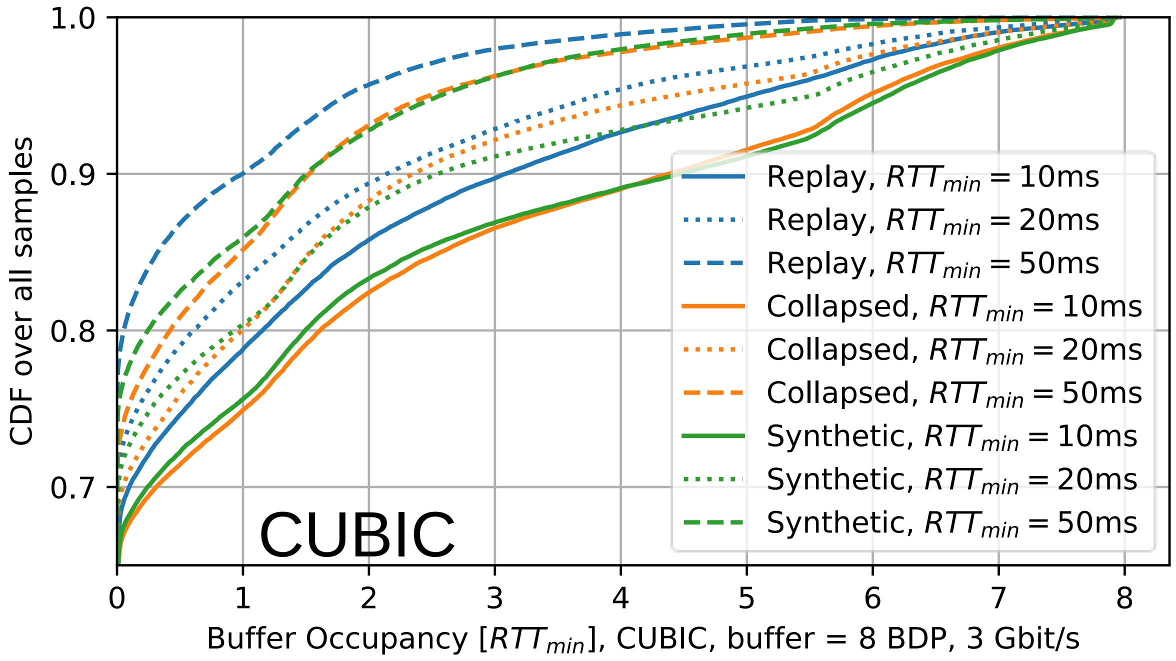
(traffic often limited by application)

Continuous application-unlimited flows → short application-limited flows?

Short Flows – Real World Trace

- 15min MAWI trace (2020/06/10, 10 Gbit/s link) → Replay at 3 Gbit/s
- BBR avoids to fill deep buffers completely
- BBR often limits queuing delay to 2 $\mathrm{RTT}_{\mathrm{min}}$





Conclusions

- We also confirmed other known fairness issues
  - RTT unfairness (large $\mathrm{RTT}_{\mathrm{min}}$ flows prevail), unfairness to CUBIC
  - Long convergence to fairness

1) Self-Induced Queuing Delay

- BBR creates noticeable self-inflicted queuing delay
  - around 1 $\mathrm{RTT}_{\mathrm{min}}$ for a single BBR flow
  - >1 $\mathrm{RTT}_{\mathrm{min}}$ more than 50% of the time for multiple flows

2) Delay-Jitter

- see paper: BBR is not as jitter sensitive as predicted by other research

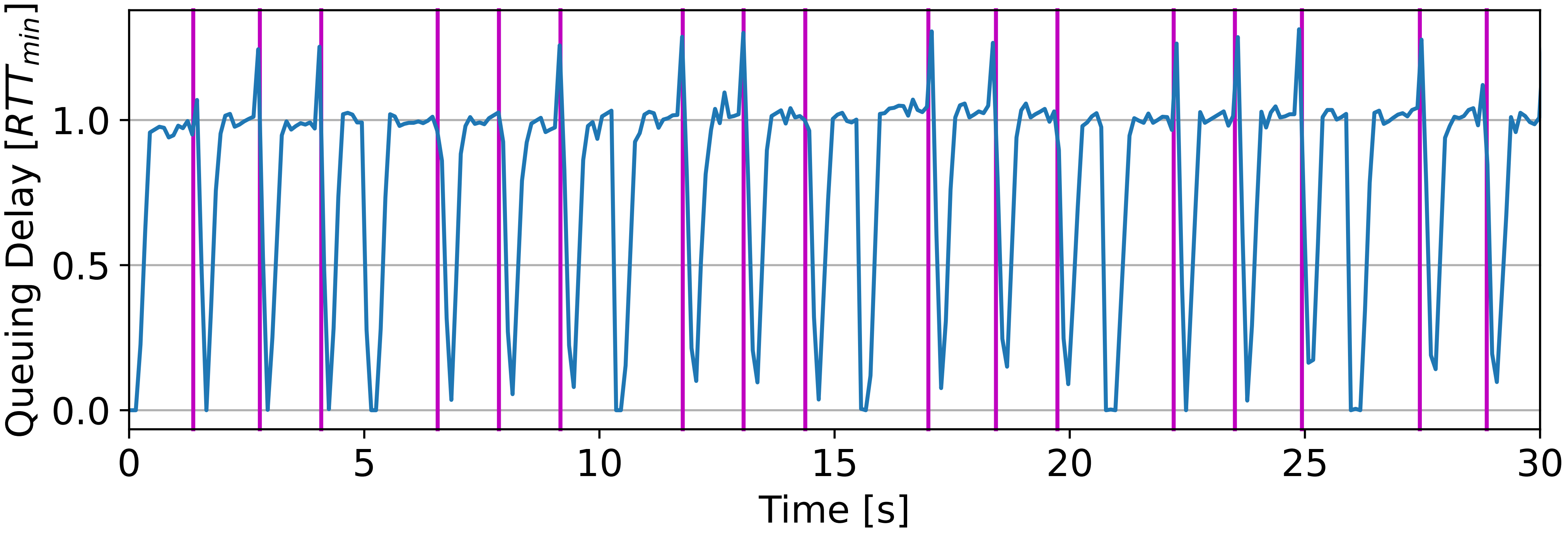
3) Short-flows using Real-World Traffic

- BBR shows slightly improved performance over CUBIC for short-lived flows
- BBR avoids completely filling deep buffers

Backup Slides

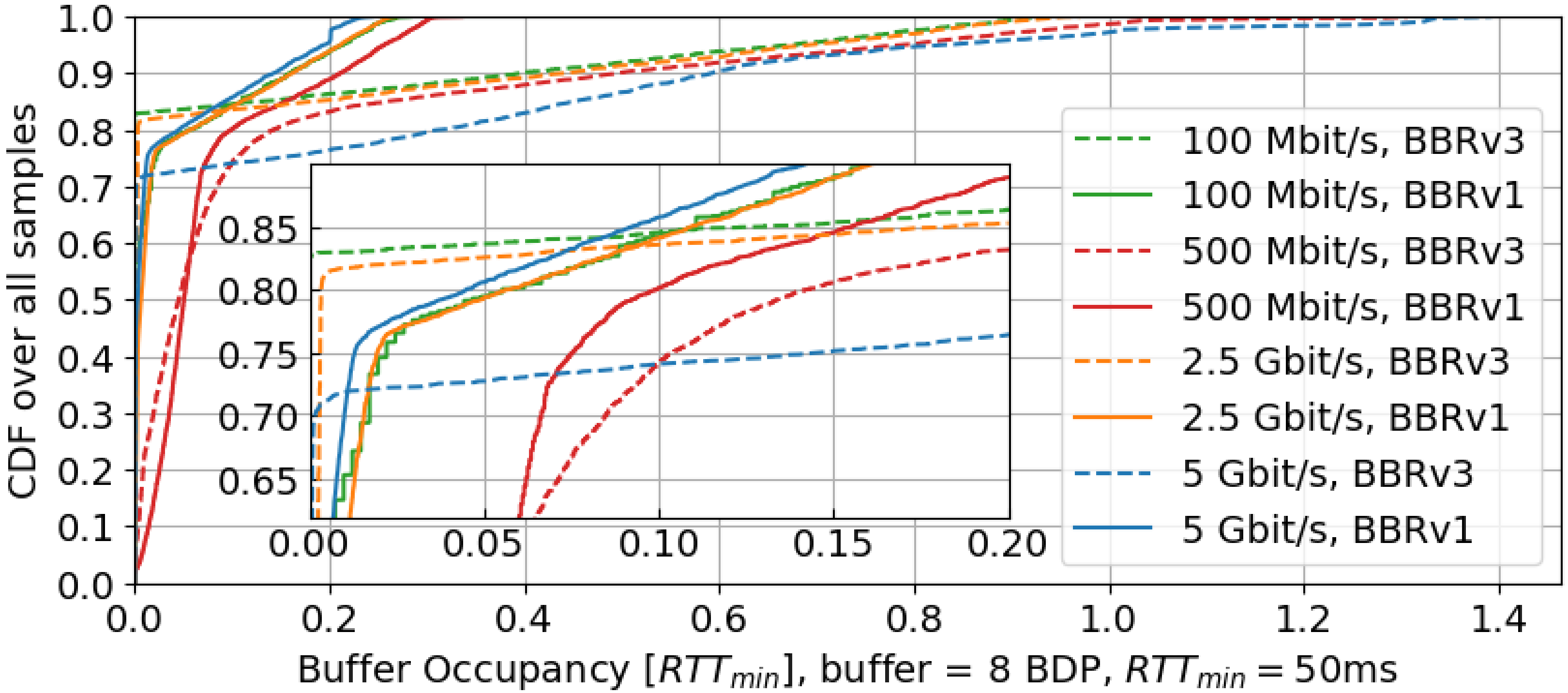
Self-induced Queuing Delay – Single Flow

- It gets worse at higher speeds and smaller $\mathrm{RTT}_{\mathrm{min}}$ values…



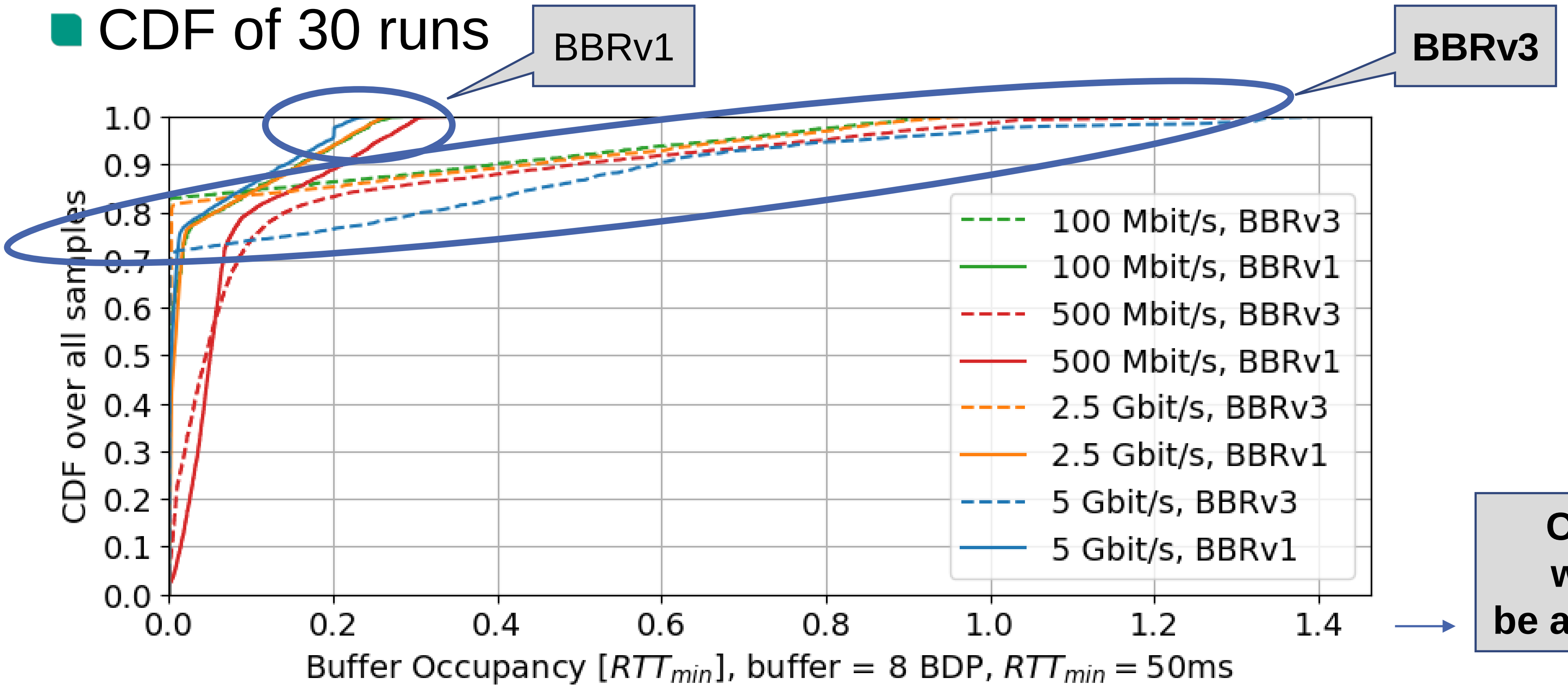
1 Gbit/s, $\mathrm{RTT}_{\mathrm{min}}$ = **10 ms**, buffer size = 8 BDP

Self-induced Queuing Delay – Single Flow



Self-induced Queuing Delay – Single Flow

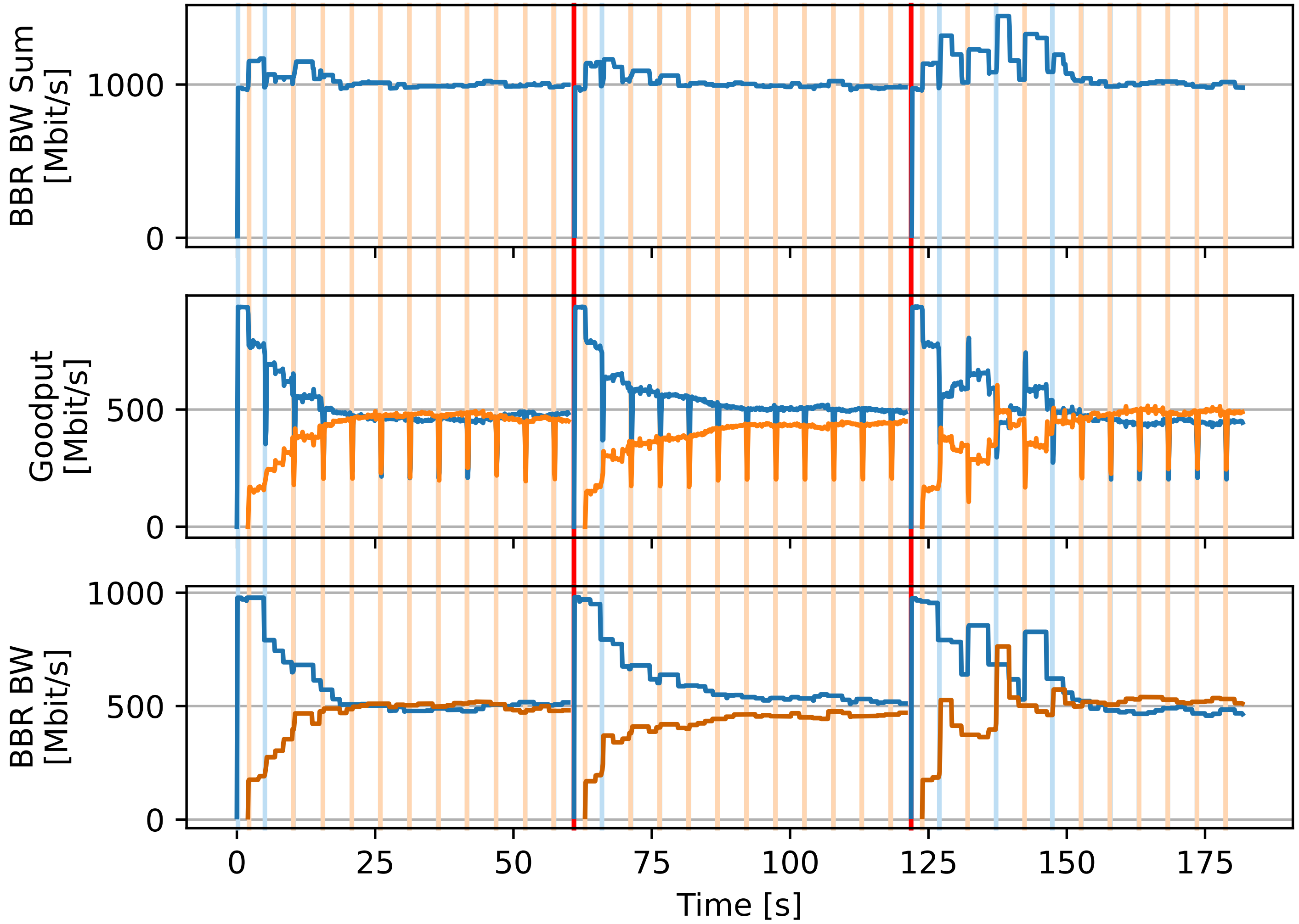
- CDF of 30 runs



CUBIC would be at 8 BDP! → 8

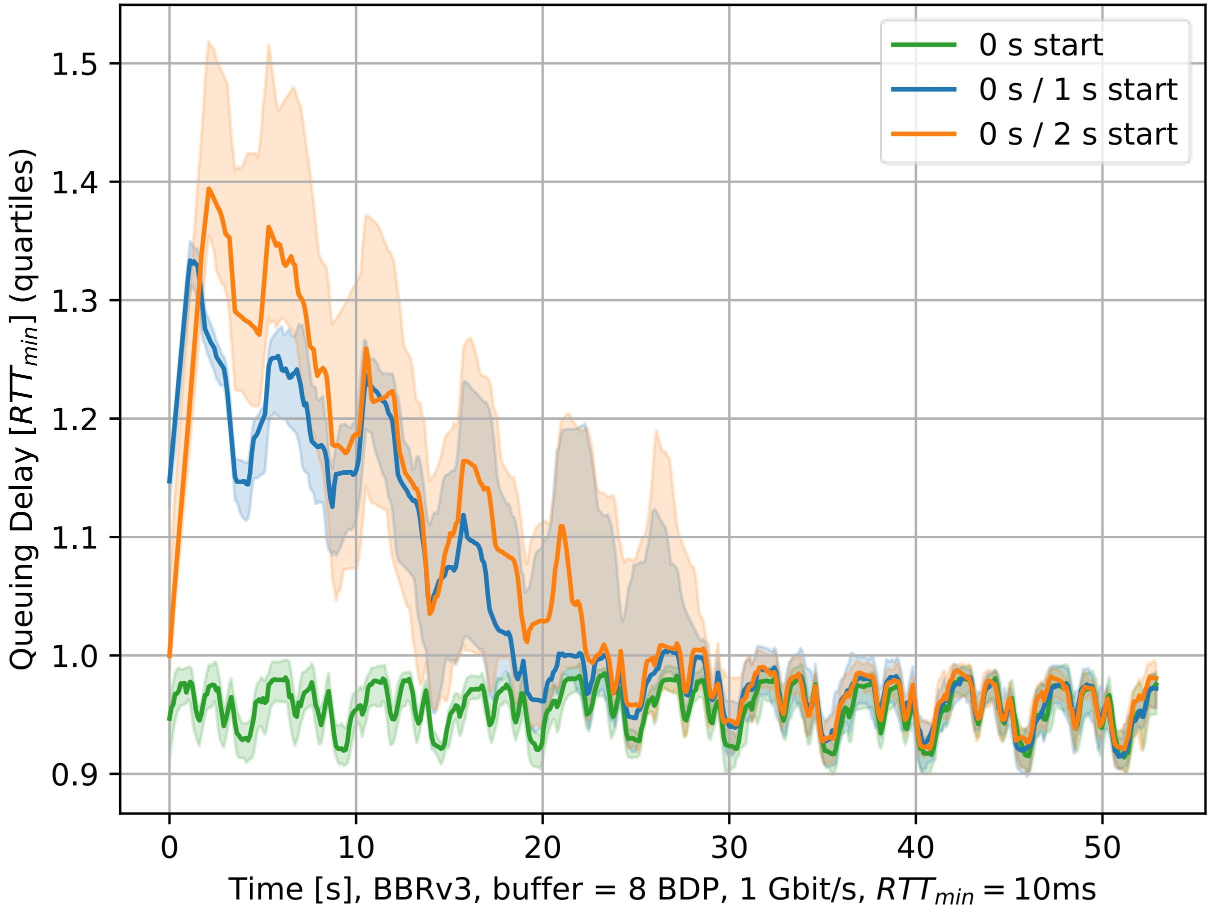
- BBRv3 creates higher peaks in queuing delay

Bandwidth Overestimation

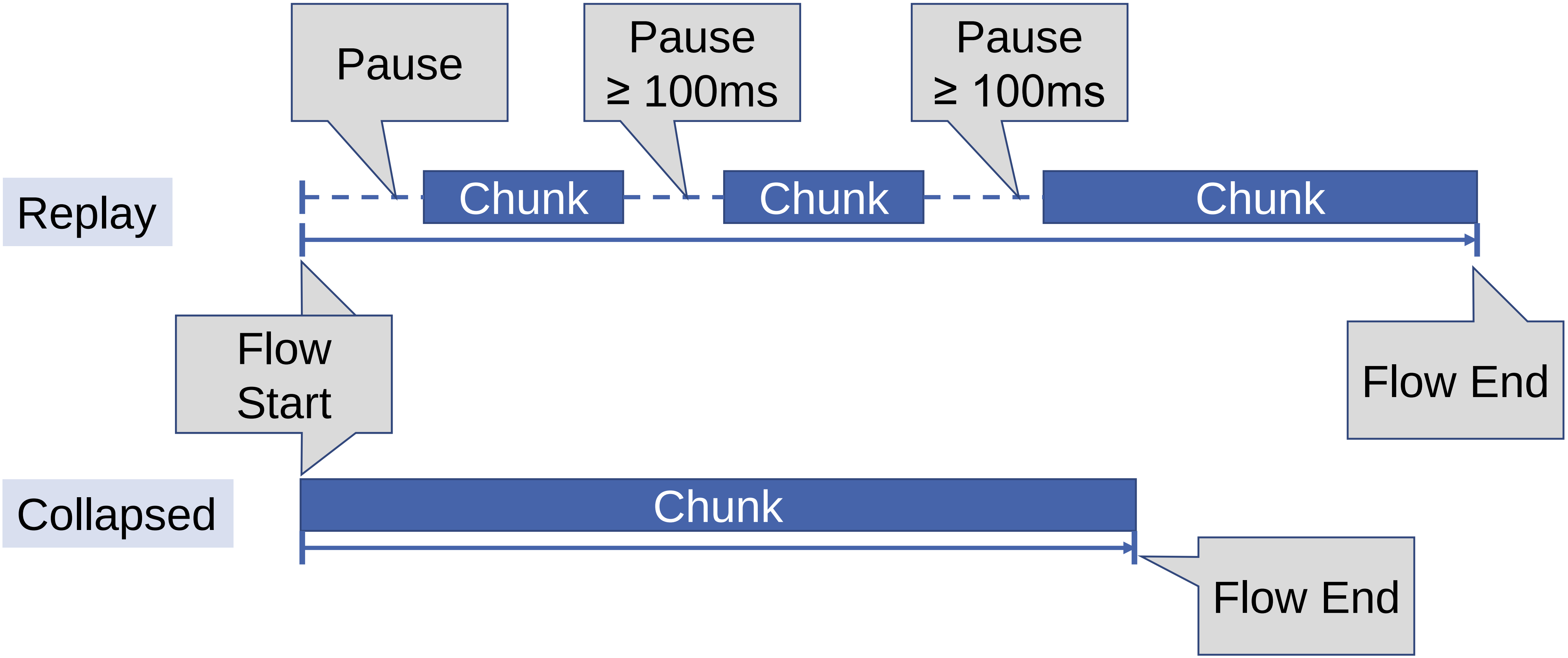


Latecomer Disadvantage

- Long convergence time for flows that start later

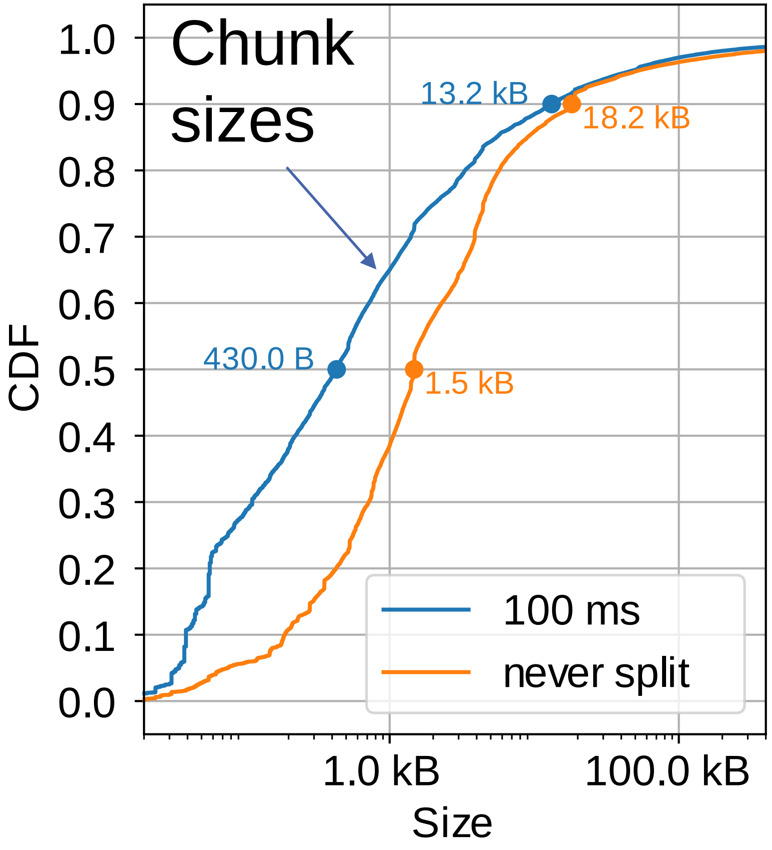


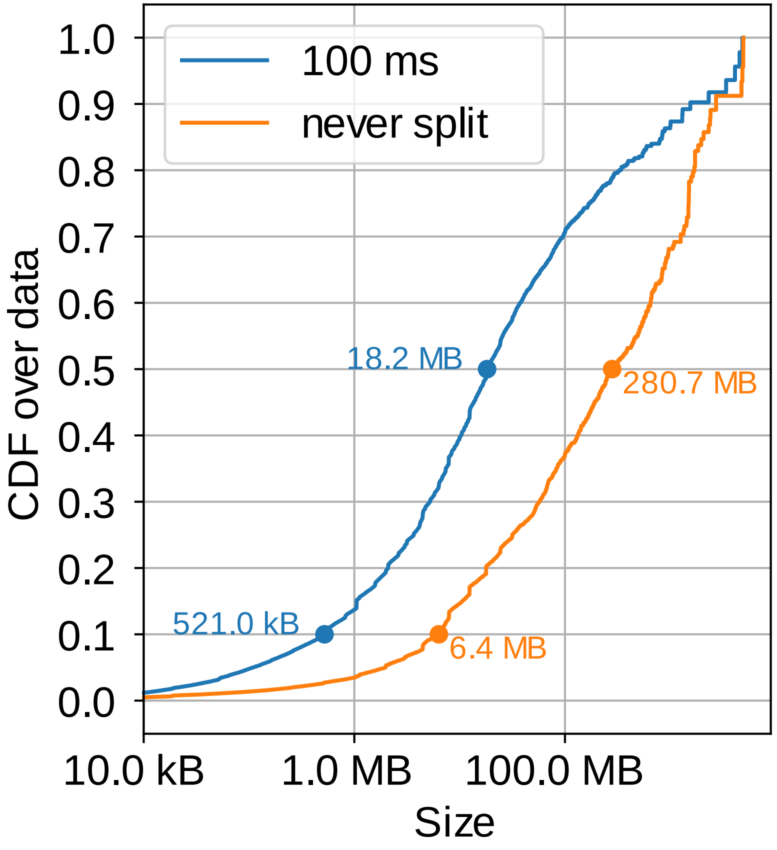
Flow-based Traffic Models



Flow Properties of MAWI Trace

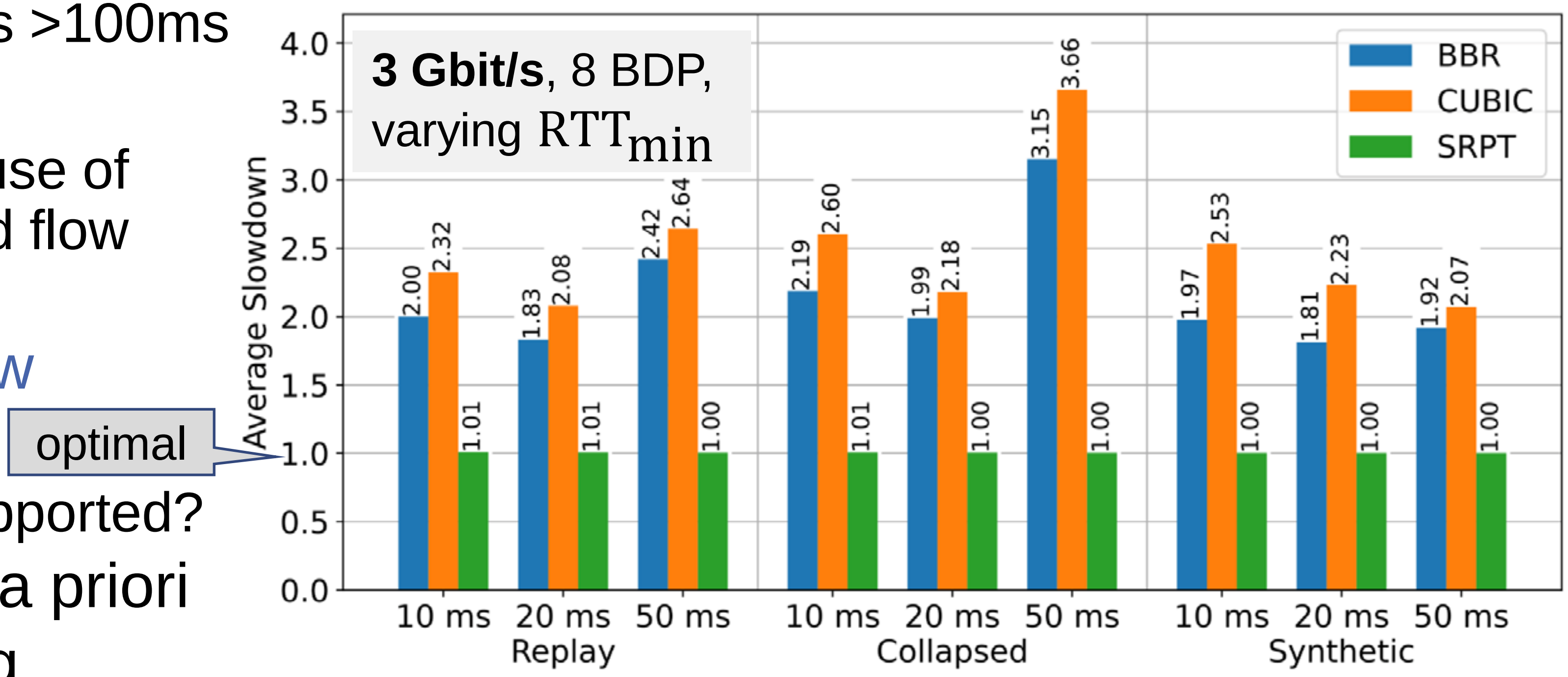
- 90% of flows smaller than 18.2kByte
- If you select a byte randomly, 90% probability that it belongs to a flow >6.4MByte





Short Flows – Real-World Trace

- Used 15min MAWI trace (2020/06/10, 10 Gbit/s link, 890 000 TCP flows)
  - with different replay models at 3 Gbit/s
    - Replay considers pauses >100ms
    - Collapsed (no pauses)
    - Synthetic (independent use of flow inter-arrival time and flow volume distributions)
- Slowdown: Normalized Flow Completion Time
  - how well are short flows supported?
- SRPT knows flow duration a priori
  - near optimum scheduling

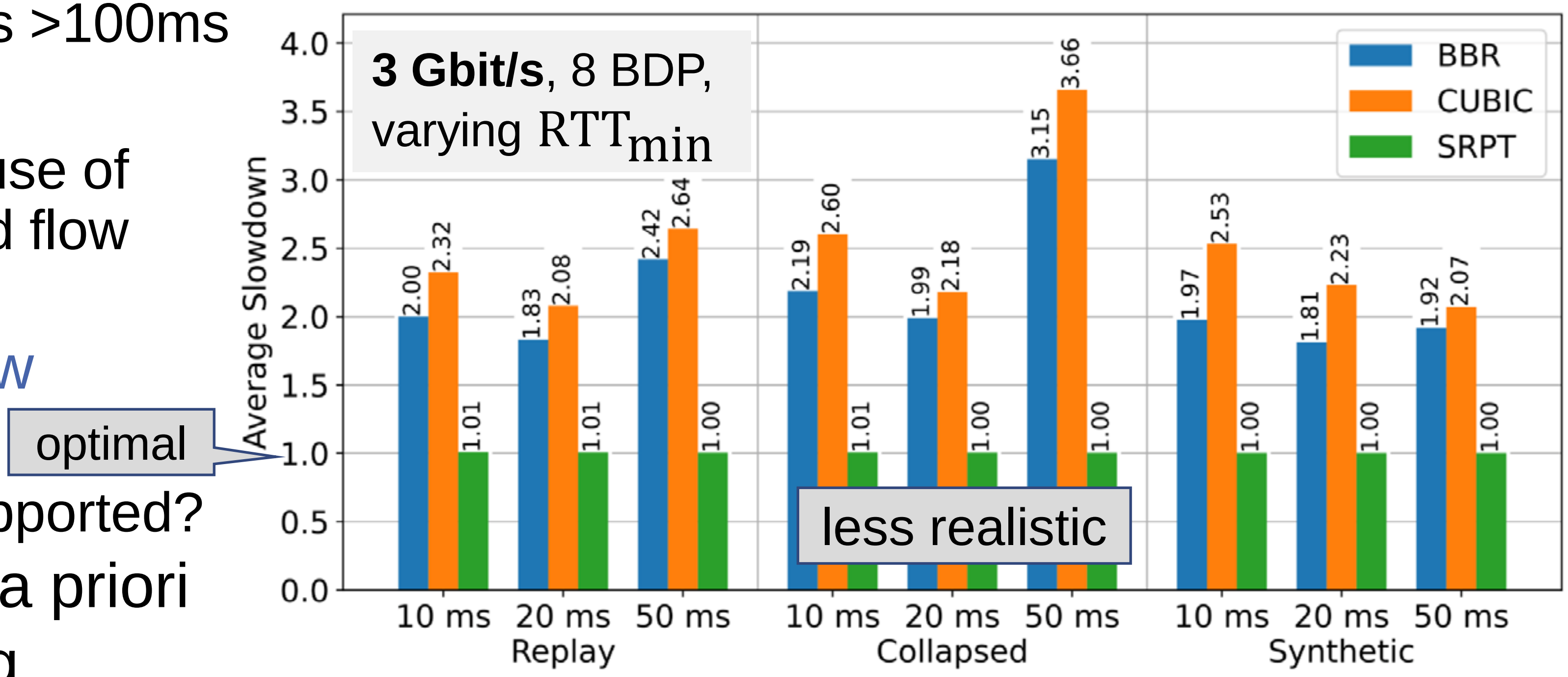


→ BBR performs slightly but consistently better than Cubic

SRPT: Shortest Remaining Processing Time

Short Flows – Real-World Trace

- Used 15min MAWI trace (2020/06/10, 10 Gbit/s link, 890 000 TCP flows)
  - with different replay models at 3 Gbit/s
    - Replay considers pauses >100ms
    - Collapsed (no pauses)
    - Synthetic (independent use of flow inter-arrival time and flow volume distributions)
- Slowdown: Normalized Flow Completion Time
  - how well are short flows supported?
- SRPT knows flow duration a priori
  - near optimum scheduling

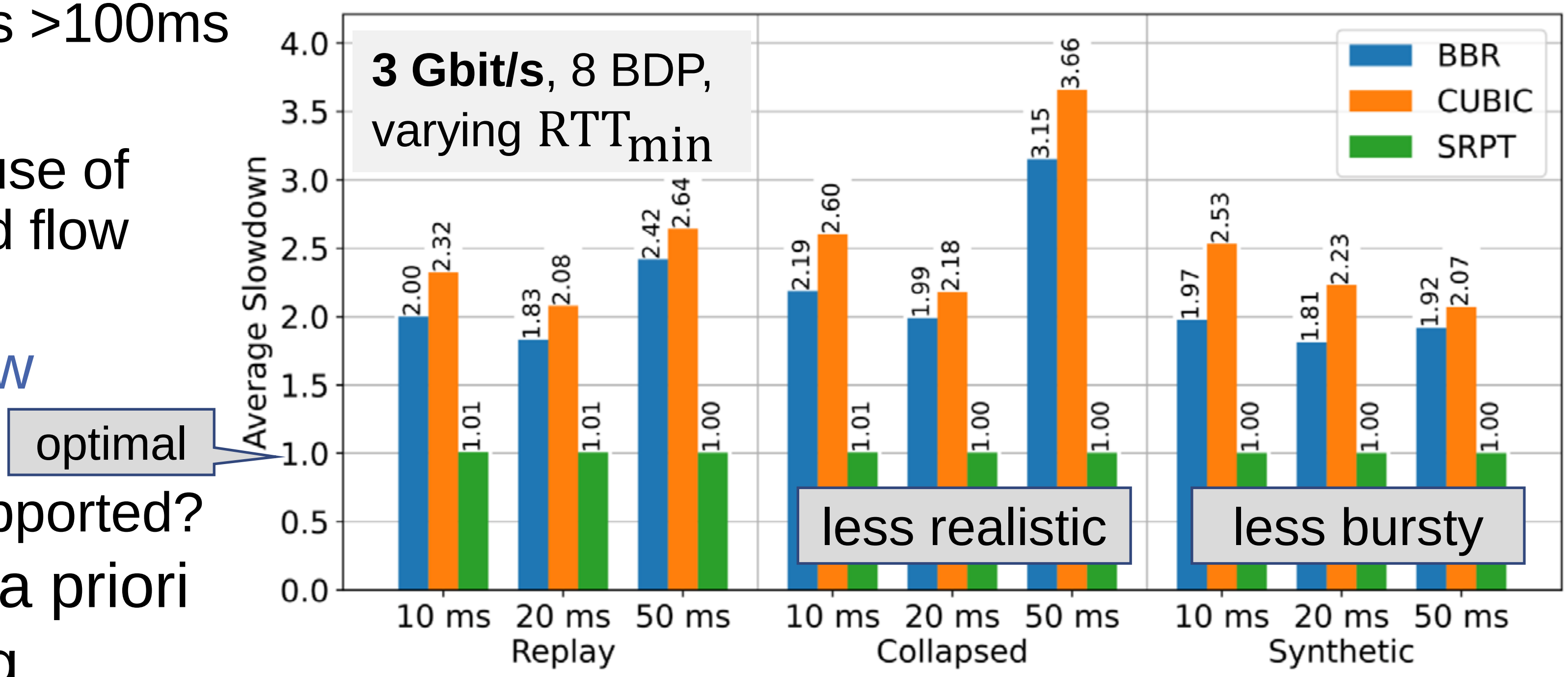


→ BBR performs slightly but consistently better than Cubic

SRPT: Shortest Remaining Processing Time

Short Flows – Real-World Trace

- Used 15min MAWI trace (2020/06/10, 10 Gbit/s link, 890 000 TCP flows)
  - with different replay models at 3 Gbit/s
    - Replay considers pauses >100ms
    - Collapsed (no pauses)
    - Synthetic (independent use of flow inter-arrival time and flow volume distributions)
- Slowdown: Normalized Flow Completion Time
  - how well are short flows supported?
- SRPT knows flow duration a priori
  - near optimum scheduling



→ BBR performs slightly but consistently better than Cubic

SRPT: Shortest Remaining Processing Time