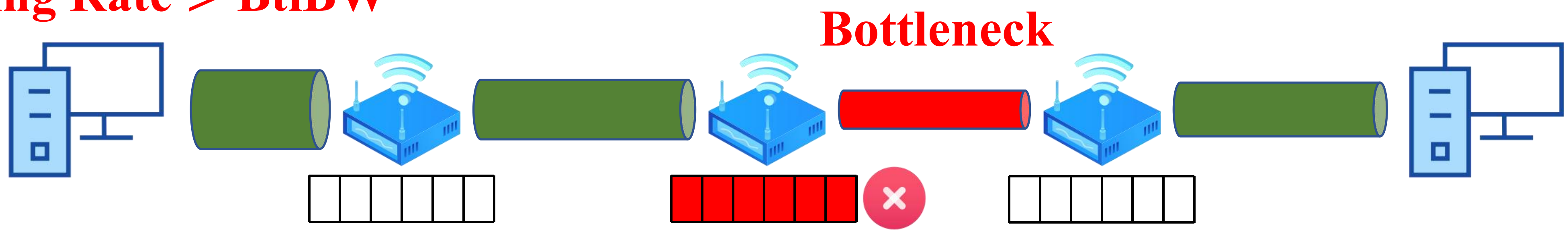
oBBR: Optimize Retransmissions of BBR Flows on the Internet

Pengqiang Bi, Mengbai Xiao, Dongxiao Yu, Guanghui Zhang
Jian Tong, Jingchao Liu, Yijun Li

Background

➢ Communication between two hosts

Sending Rate > BtlBW



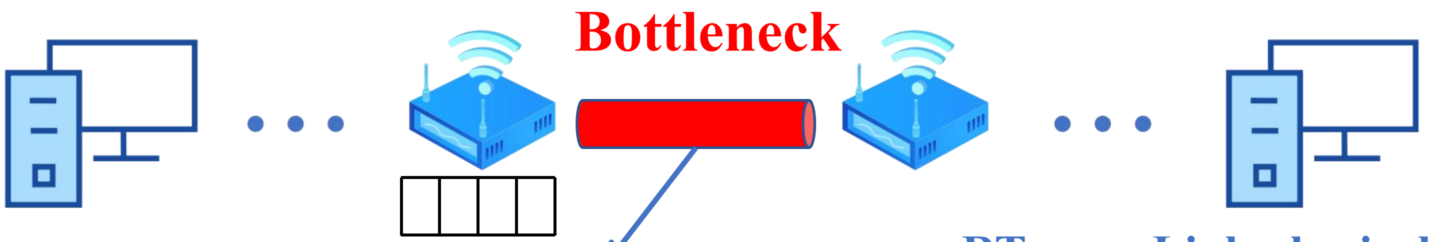
The data transmission is throttled at the bottleneck

➢ Congestion Control

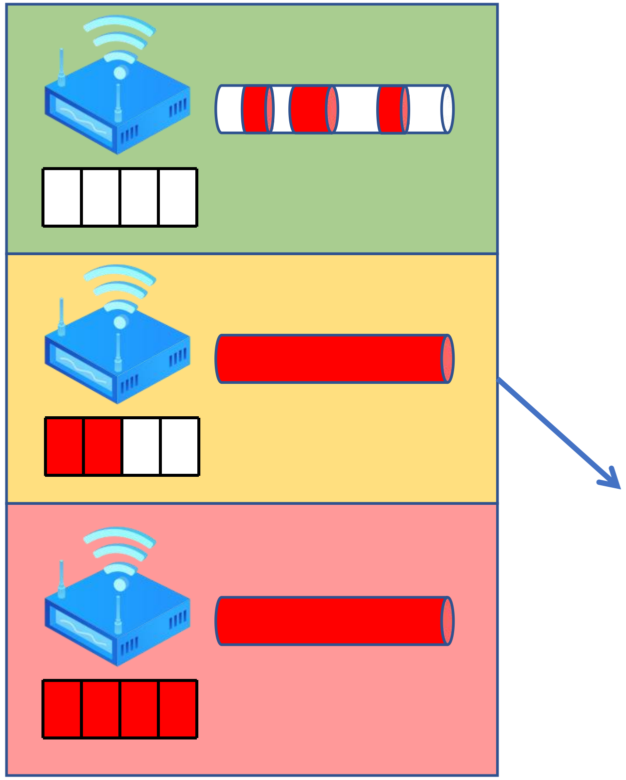
- Control the sending rate of packets
- Prevent excessive packet injection causing network breakdown

Background

- Communication between two hosts



RTprop: Link physical latency



App limited — Link not full → BtlBW is not fully utilized

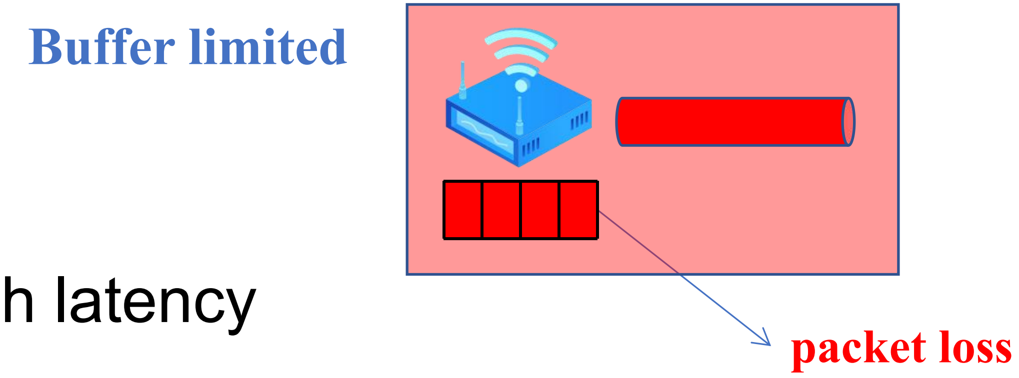
Bw. limited — Link is full, Buffer not full → Latency gradually increases

Data in the link is BtlBW × RTprop (BDP)

Buffer limited — Link is full, Buffer is full → Upcoming packet loss

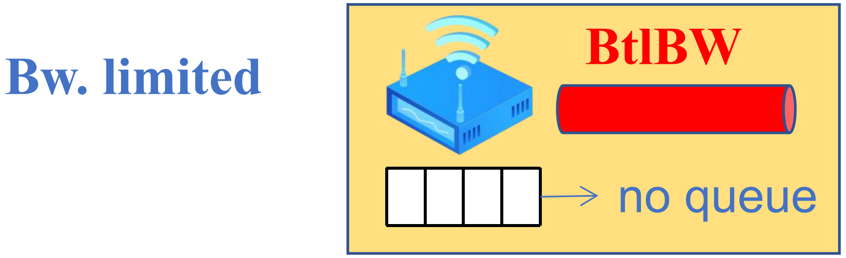
Congestion Control

- Cubic: loss-based congestion control
  - Tend to fill up the buffer
  - Back off when packet loss occurs



- Weakness: Throughput oscillation and high latency

- BBR: model-based congestion control
  - Probe BtlBW and RTprop
  - Send data at the rate of $\widehat{BtlBW}$



- Goal: Operate at the optimal point, with no data queued in the buffer

BBR State Machine

- Startup: Exponentially probe available bandwidth
- Drain: Drain the excessive data
- ProbeBW: Cyclically probe **BtlBW**
  - Divided into 8 RTTs per cycle

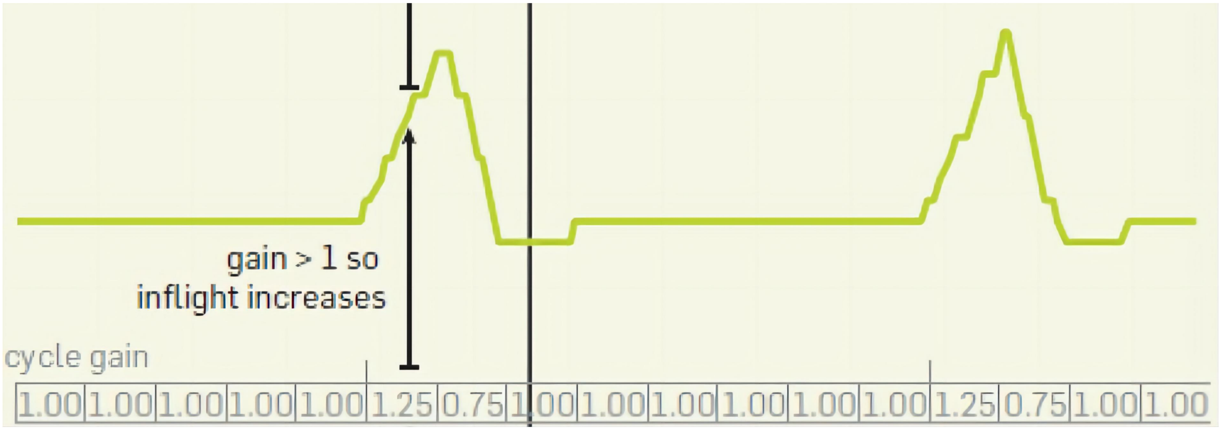
cycle

pacing_gain:

1.25	0.75	1.0	1.0	1.0	1.0	1.0	1.0

```
                 |
                 V
    +---> Startup  ----+
    |        |         |
    |        V         |
    |      Drain   ----+
    |        |         |
    |        V         |
    +---> ProbeBW -----+
    |      ^    |      |
    |      |    |      |
    |      +----+      |
    |                  |
    +---- ProbeRTT <---+
```

- ProbeRTT: Probe **RTprop**
  - Limit in-flight to 4 packets

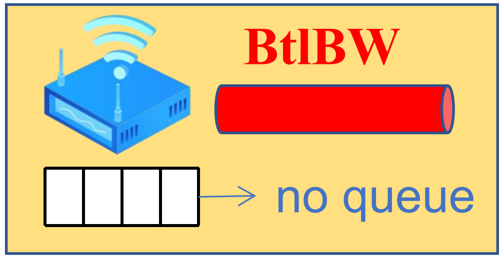


Neal Cardwell, et al., BBR: congestion-based congestion control

BBR Congestion Control

- Advantages of BBR
  - High throughput, low latency
  - High resistance to packet loss

Bw. limited



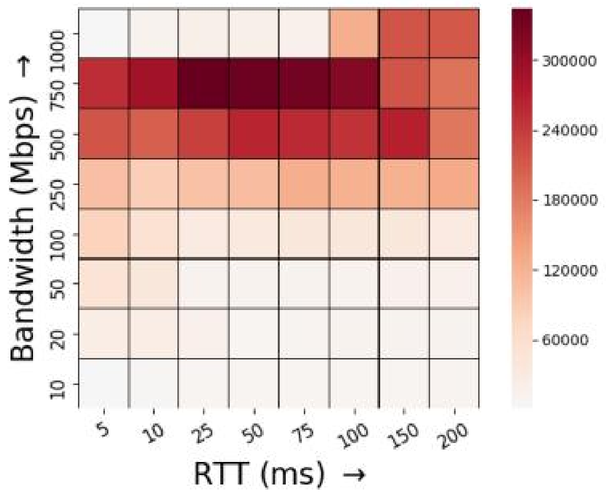
- BBR has high retransmissions

- Our goal: reduce retransmissions of BBR

Related Work

Discover and address high retransmissions in BBR

- Yi Cao, et al., When to use and when not to use BBR (IMC ’19)
  - BBR has high retransmissions in shallow buffer
- Neal Cardwell, et al., draft-cardwell-iccrg-bbr-congestion-control-02 (BBRv2)
  - Decrease sending rate when packet loss occurs
  - Cost: Throughput will suffer from random packet loss
- Santiago Vargas, et al., BBR Bufferbloat in DASH Video (WWW ’21)
  - BBR-S: Estimate bandwidth more accurately
- Our work:
  - Analyze the reasons of the high retransmission of BBR
  - Without weakening the packet loss resistance of BBR



(c) BBR’s # retransmits, *buffer*=100KB

Yi Cao, et al., When to use and when not to use BBR: An Empirical Analysis and Evaluation Study (IMC ’19)

High Retransmissions of BBR

Why BBR has high retransmissions ?

- The shallow buffer cannot hold the in-flight data
- The $\widehat{BtlBW}$ cannot timely react to bandwidth drops

↓

High Retransmissions

High Retransmissions of BBR

Why BBR has high retransmissions ?

- The shallow buffer cannot hold the in-flight data
- The $\widehat{BtlBW}$ cannot timely react to bandwidth drops

↓

High Retransmissions

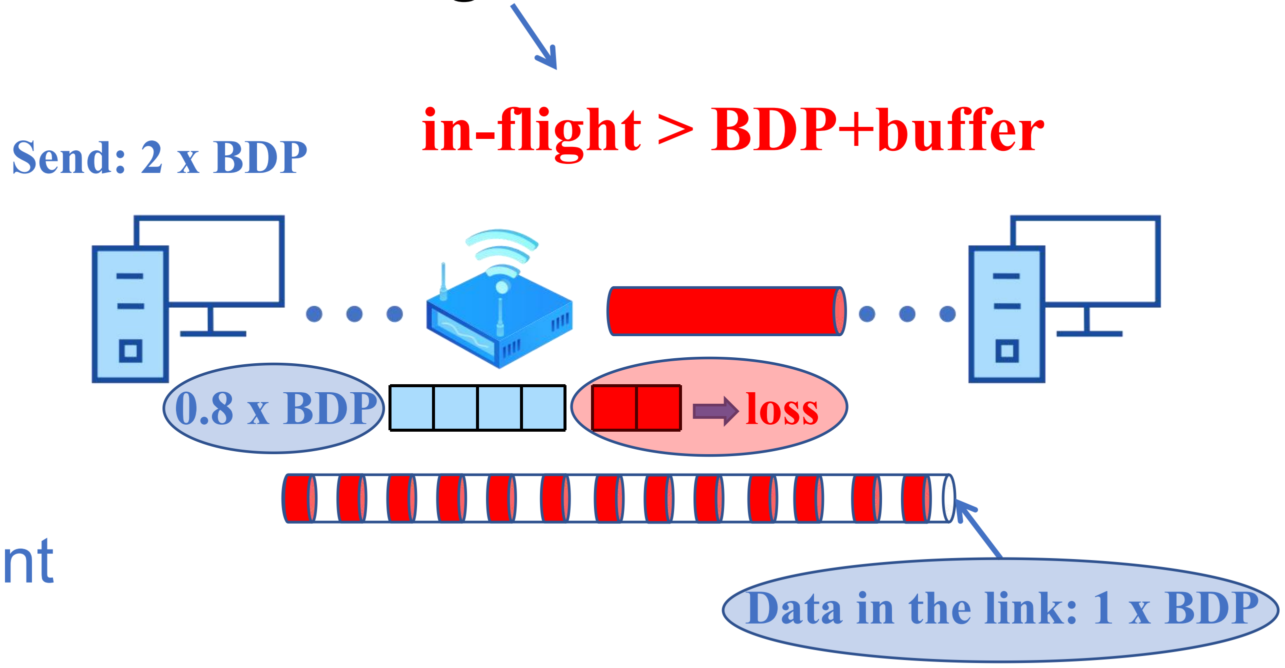
High Retransmissions of BBR

The shallow buffer cannot hold the in-flight data

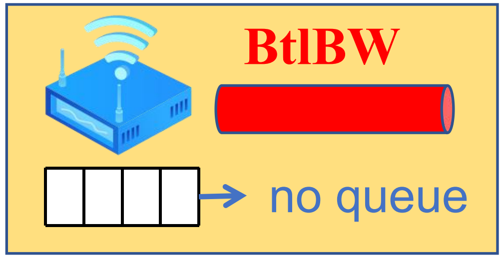
- BBR limit the in-flight data:

$$BBR_{inflight} = cwnd\_gain \times BDP$$

- $cwnd\_gain$ is set to 2 in practice



BBR goal: Operate at the optimal point



Why $cwnd\_gain$ is set to 2 instead of 1?

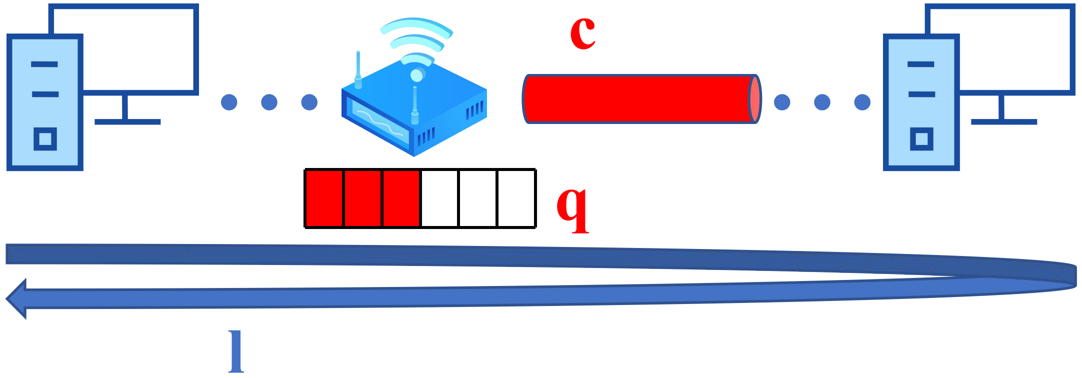
Modeling BBR

BBR behavior becomes complex when competing with loss-based flows (Cubic)

Assume:

- **c:** Bottleneck link capacity
- **l:** Link physical latency without queuing
- **q:** Capacity of the bottleneck buffer
- **g** Represents ***cwnd_gain***

- **p** **Buffer ratio occupied by Cubic**



So, the max bandwidth BBR can probe is $(1-p)\times c$

Then, the min RTT BBR can probe is $\frac{p\times q}{c} + l$ **(add the queuing latency of cubic)**

$$BBR_{inflight} = g \times \widehat{BtlBW} \times \widehat{RTprop} \Rightarrow BBR_{inflight} = g \times (1-p) \times c \times \left(\frac{p\times q}{c} + l\right)$$

Modeling BBR

Then, the queued data in the bottleneck of a BBR flow is:

$$queued = BBR_{inflight} - (1-p) \times c \times l$$ → Data in the link

Finally, we get the buffer occupied ratio by BBR:

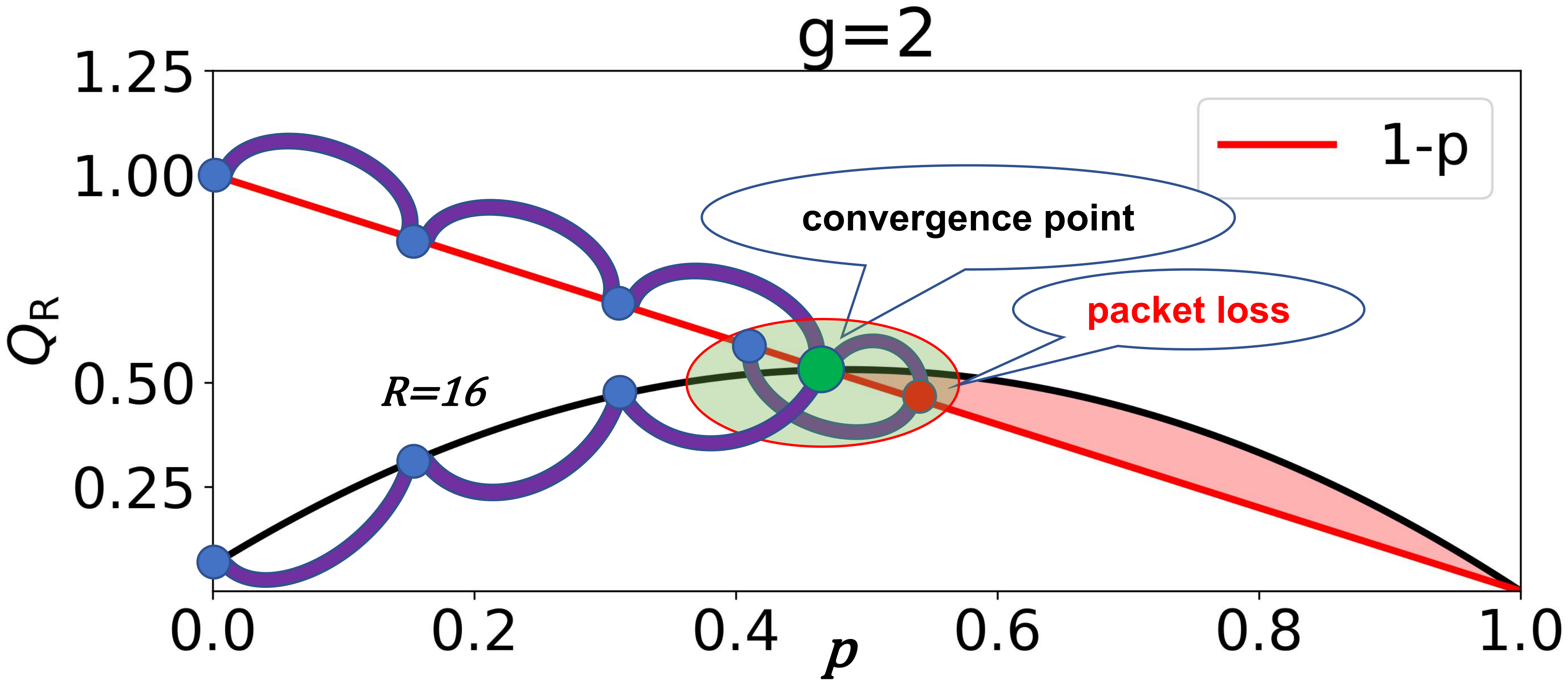
$$\boldsymbol{Q_R(p; g, R)}$$ → **Related to p, g, R**

- ***g*** represents ***cwnd_gain***
- **R** is the ratio of buffer size to BDP
- **p** is the buffer occupied ratio by BBR

A BBR flow is affected by coexisting Cubic flows

Modeling BBR

- In a deep buffer

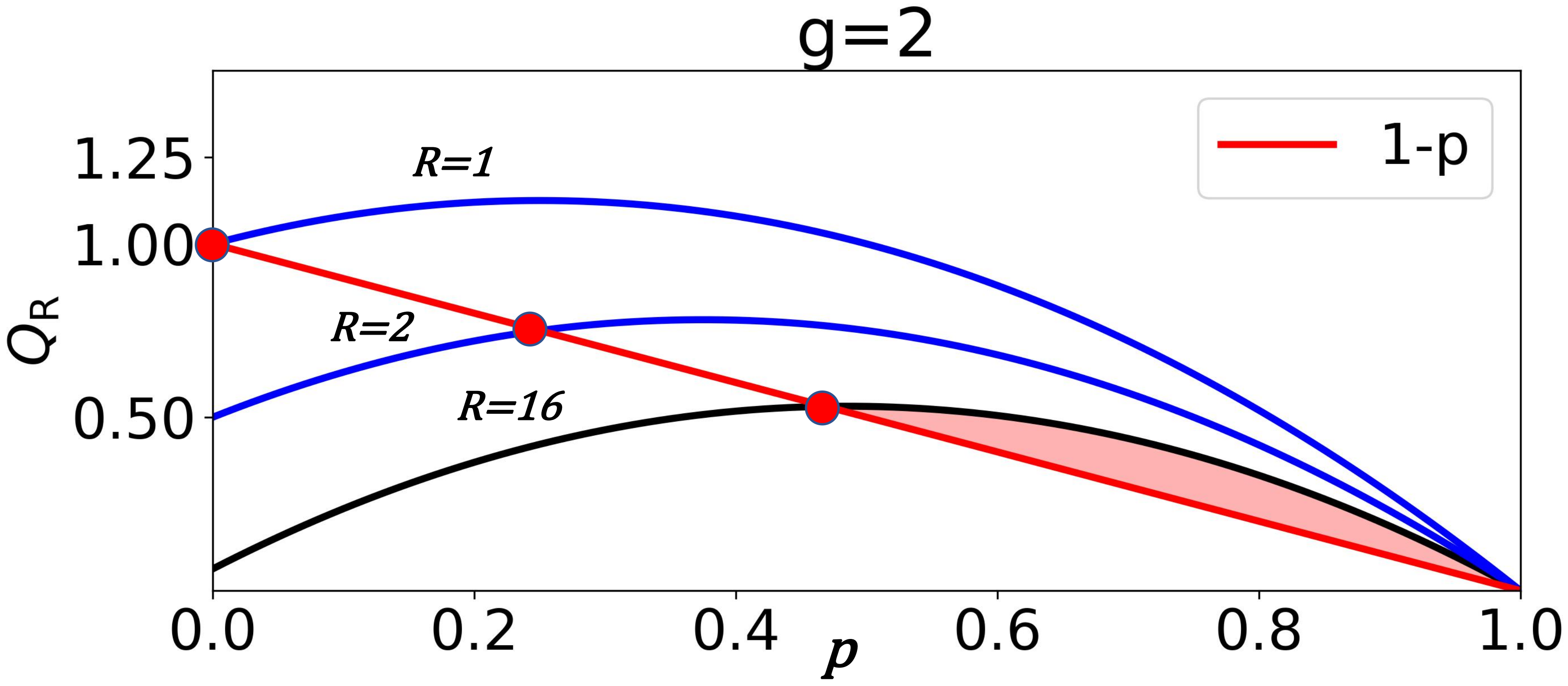


p: The buffer ratio occupied by Cubic flows

Q_R: The buffer ratio occupied by a BBR flow

Modeling BBR

➢ In shallow buffer

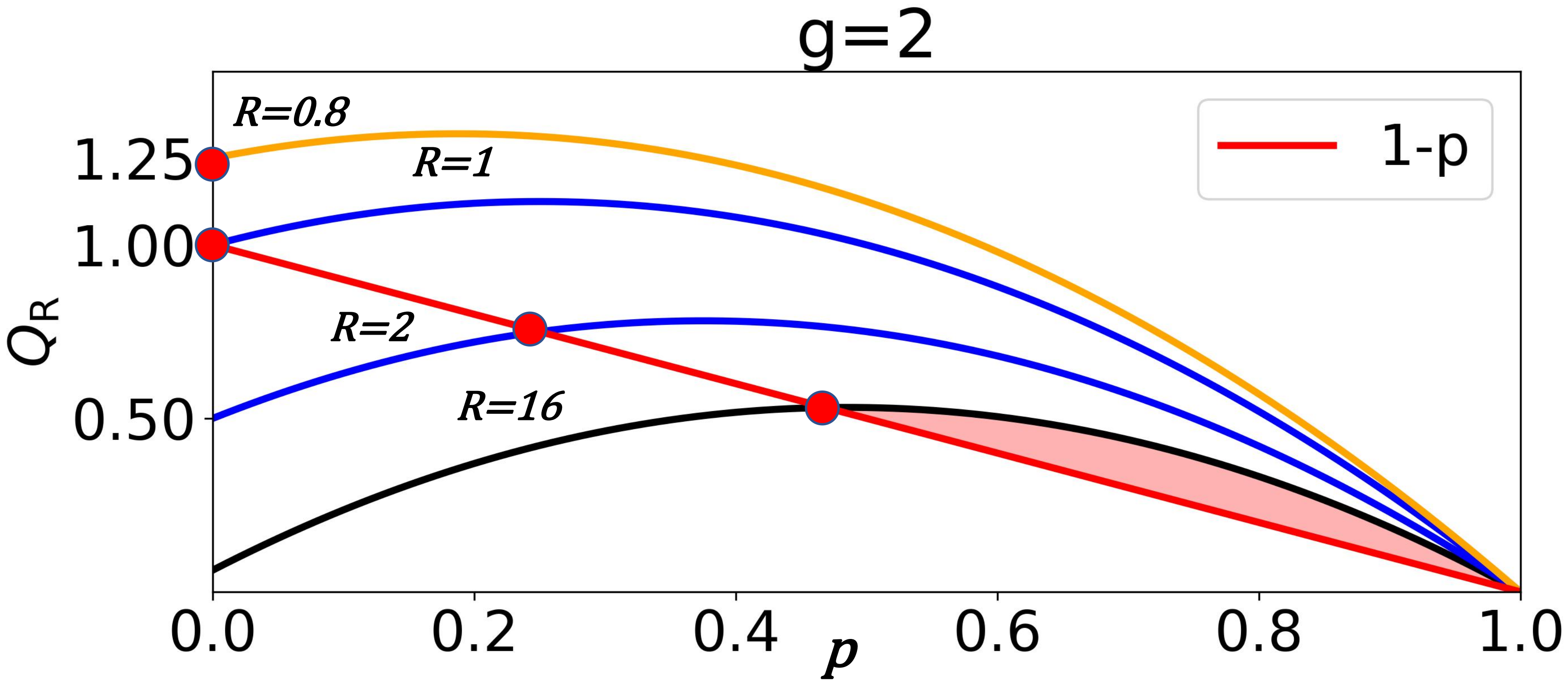


p: **The buffer ratio occupied by Cubic flows**

Q_R: **The buffer ratio occupied by a BBR flow**

Modeling BBR

➢ Buffer less than 1, packet loss occurs



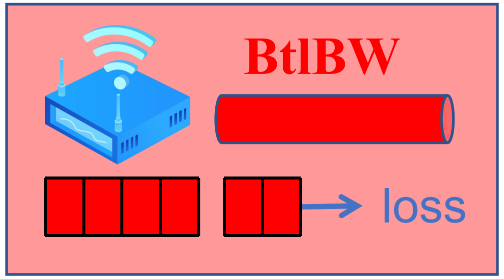
p: **The buffer ratio occupied by Cubic flows**

Q_R: **The buffer ratio occupied by a BBR flow**

High Retransmissions of BBR

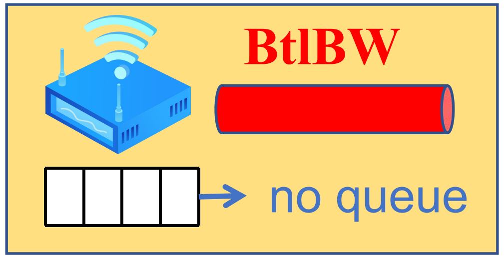
- In deep-buffered links, BBR and Cubic fairly share bandwidth

$$BBR_{inflight} = 2 \times BDP$$



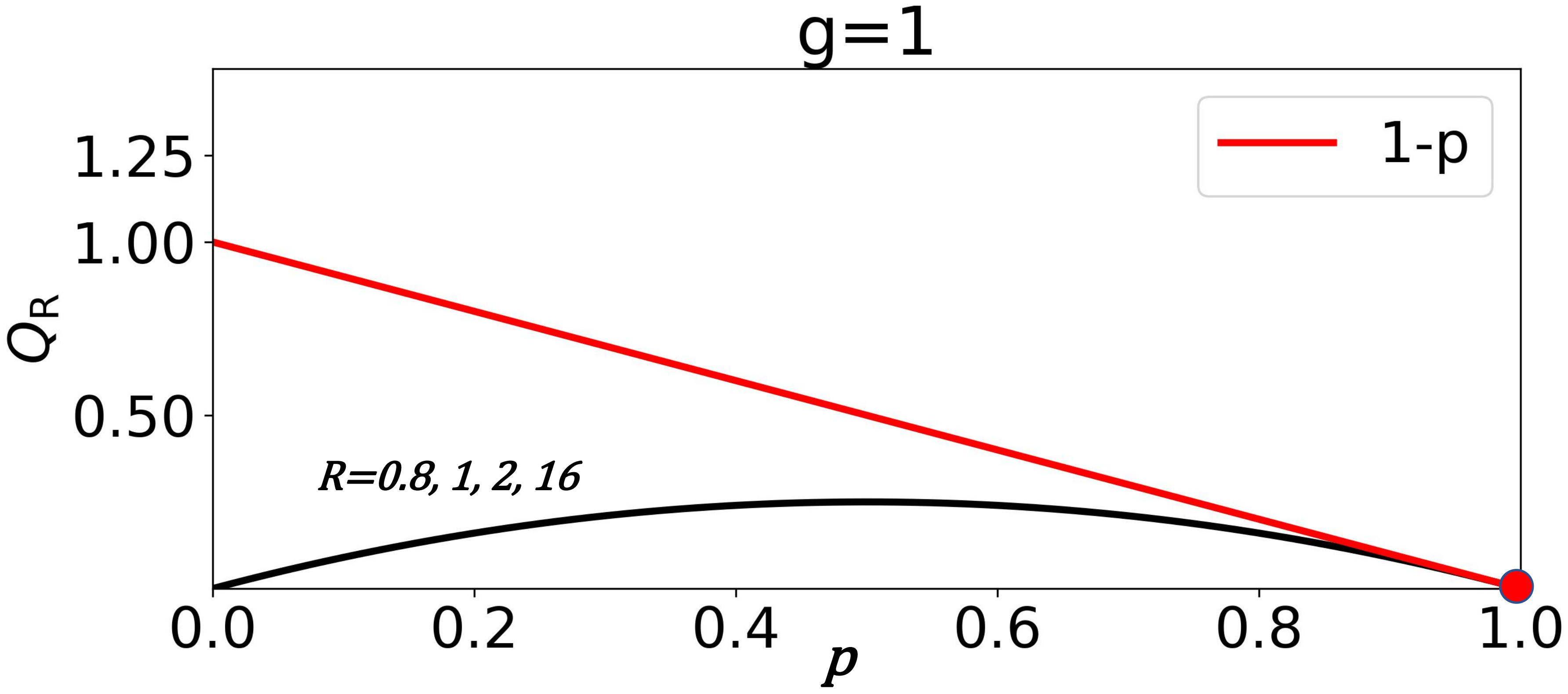
in-flight > BDP+buffer → **High retransmissions**

Can we reduce g to 1 ?



Modeling BBR

➤ Can we reduce g to 1 ?

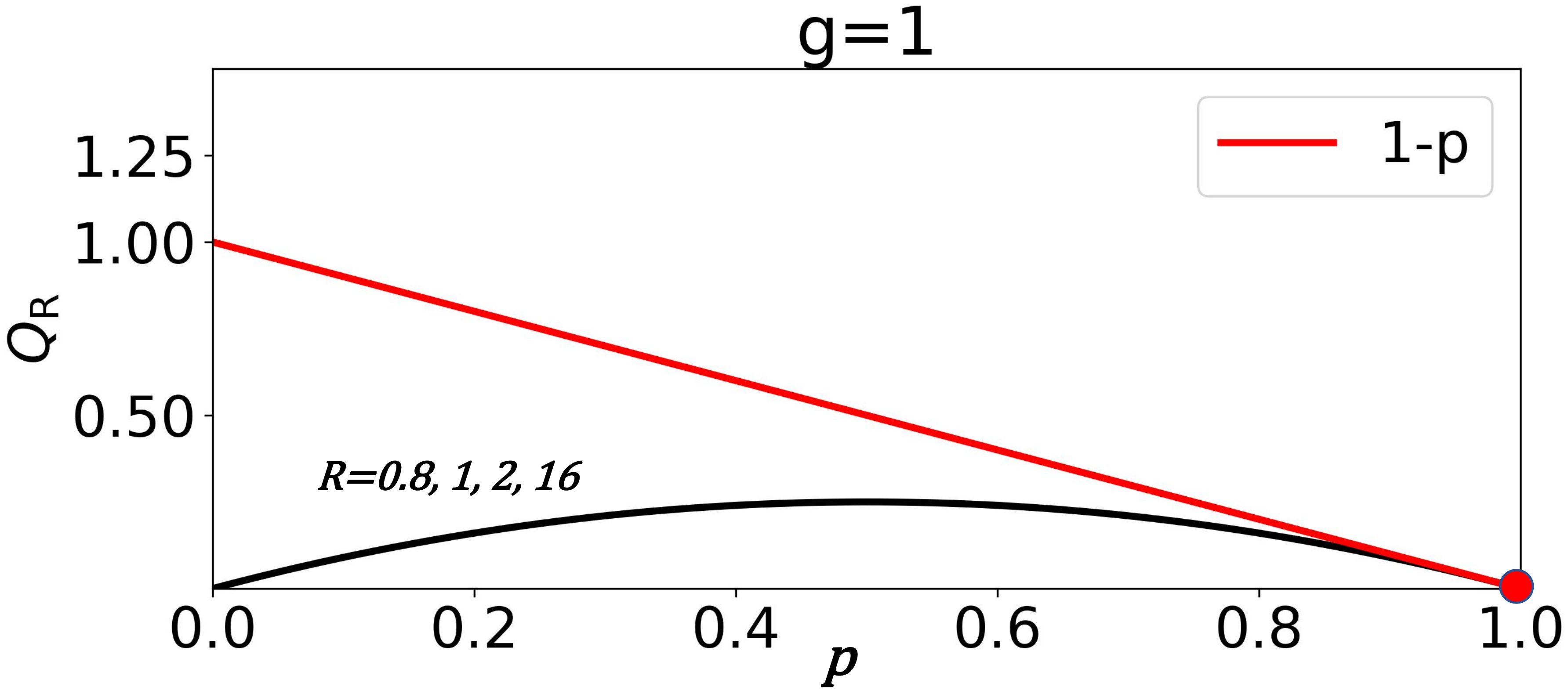


p: **The buffer ratio occupied by Cubic flows**

Q_R: **The buffer ratio occupied by a BBR flow**

Modeling BBR

- Can we reduce g to 1 ?



We cannot simply reduce the g

High Retransmissions of BBR

Why BBR has high retransmissions ?

- The shallow buffer cannot hold the in-flight data

- The $\widehat{BtlBW}$ cannot timely react to bandwidth drops

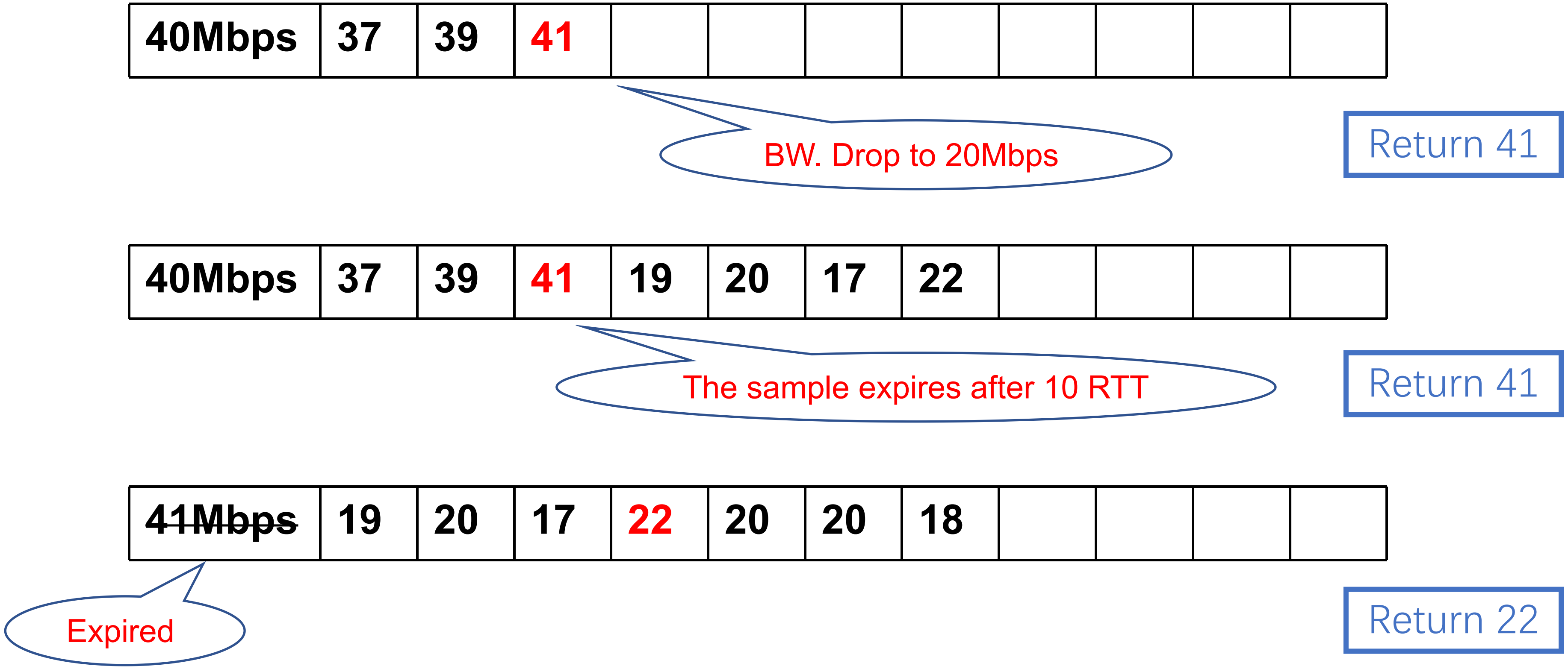
↓

High Retransmissions

Bandwidth Drops

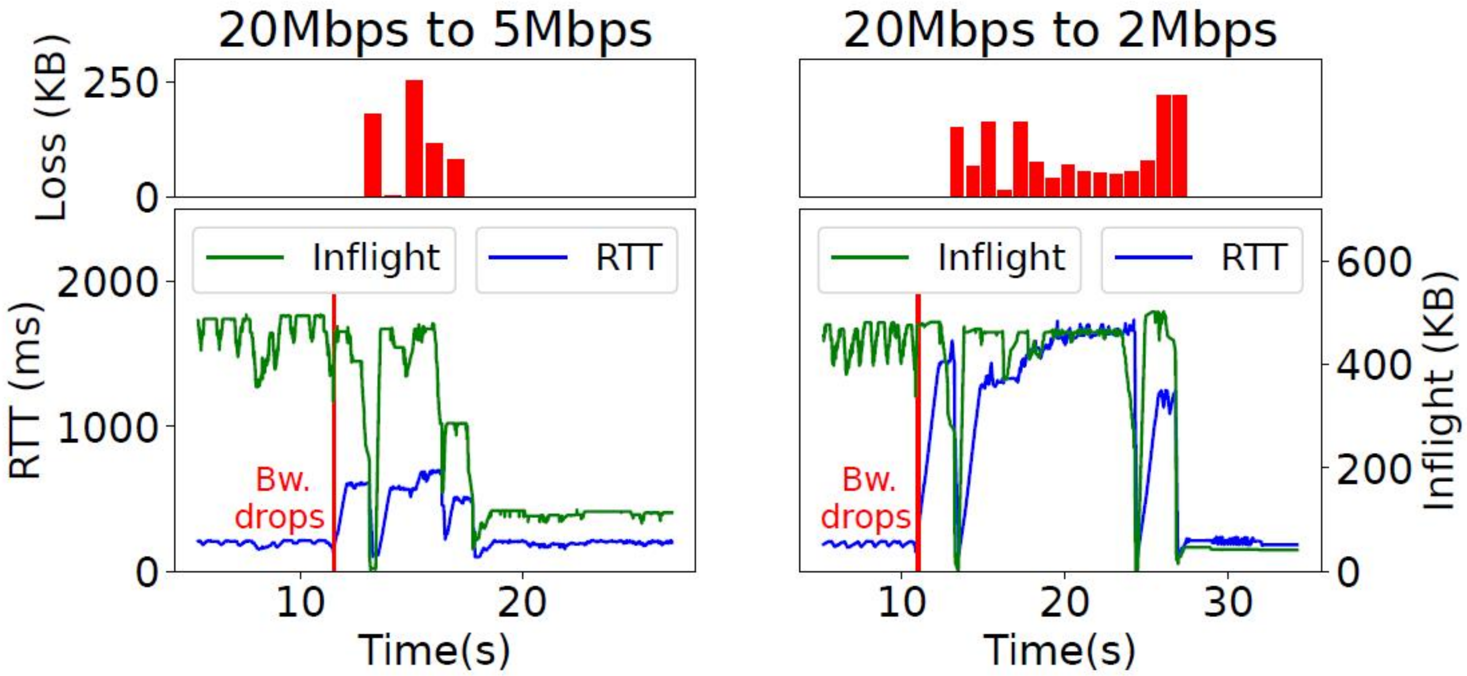
The bandwidth estimation of BBR

- Collects the delivery rate samples in a time window (typically 10 RTTs)
- Chooses the maximum as the bandwidth estimate
- BBR react to the bandwidth drops sluggishly



Preliminary Experiments

- Congestion leads to longer expiration time for overestimated samples



- ~7 seconds (RTT increases to 700ms) for 20Mbps to 5Mbps
- ~15 seconds (RTT increases to 1500ms) for 20Mbps to 2Mbps

oBBR Design

The main reasons for high retransmissions in BBR

- The shallow buffer cannot hold the in-flight data
- The $\widehat{BtlBW}$ cannot timely react to bandwidth drops

oBBR design:

- ➢ **Adaptive g**
  - Estimate the bottleneck buffer size
  - Adaptively adjust g
- ➢ **Timely bandwidth updates**
  - Sense bandwidth drops accurately
  - Update the bandwidth estimate in time and reversible

oBBR Design

The main reasons for high retransmissions in BBR

- The shallow buffer cannot hold the in-flight data
- The $\widehat{BtlBW}$ cannot timely react to bandwidth drops

oBBR design:

- ➢ **Adaptive g**
  - Estimate the bottleneck buffer size
  - Adaptively adjust g
- ➢ **Timely bandwidth updates**
  - Sense bandwidth drops accurately
  - Update the bandwidth estimate in time and reversible

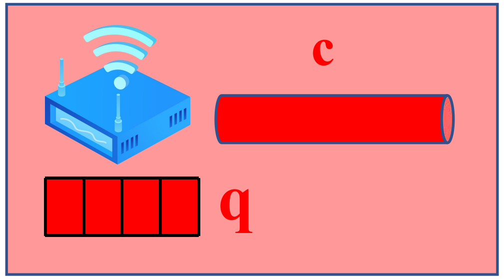
Adaptive g

Packet loss happens when buffer is full

- Estimate the bottleneck buffer size using the latest RTT sample

$$q = c \times (RTT_{latest} - \widehat{RTprop})$$

Queuing Latency



To enable the shallow buffer hold the in-flight data

Limit in-flight data to BDP+q

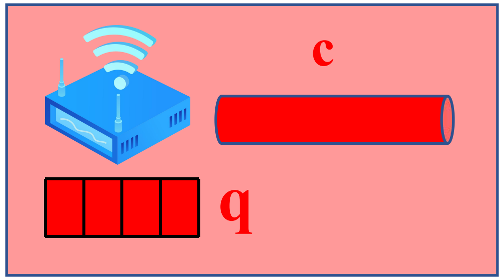
Adaptive g

- **Limit in-flight data to BDP+q**
  - Adjust g:

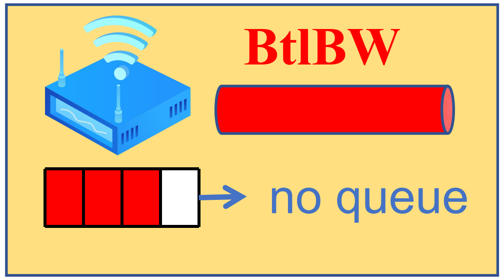
$$g = min\left(1 + \mu \times \frac{q}{BDP}, g_{max}\right) \Rightarrow BBR_{inflight} = BDP + \mu \times q$$

- $\boldsymbol{g_{max} = 2}$ **has the same behavior as BBR in the deep buffer**
- $\boldsymbol{\mu}$ $(0 < \boldsymbol{\mu} < 1)$**: leave a margin in the buffer**
  - Reduce retransmissions due to bandwidth overestimation

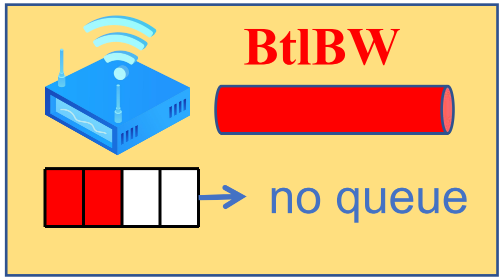
μ=1



μ=0.75



μ=0.5



Adaptive g

➢ Guarantee the competitiveness of BBR

- When ack is received, increase g:

$$g = min(g + \alpha \times \frac{Size_{ack}}{BDP}, g_{max}) \Rightarrow BBR_{inflight} += \alpha \times ack\ size$$

➢ $\boldsymbol{\alpha}$ controls the speed of growth

➢ Competitiveness is restored when migrating to a deep buffer

oBBR Design

The main reasons for high retransmissions in BBR

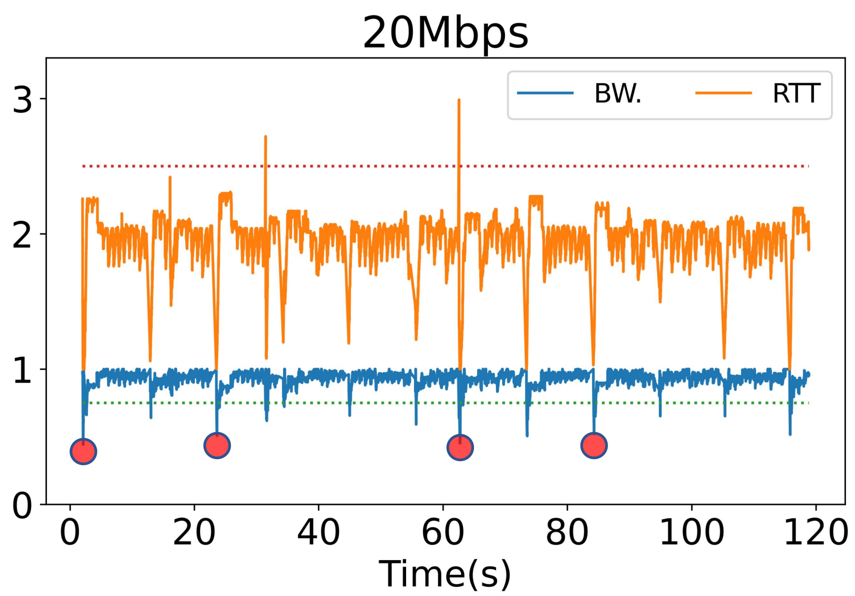
- The shallow buffer cannot hold the in-flight data
- The $\widehat{BtlBW}$ cannot timely react to bandwidth drops

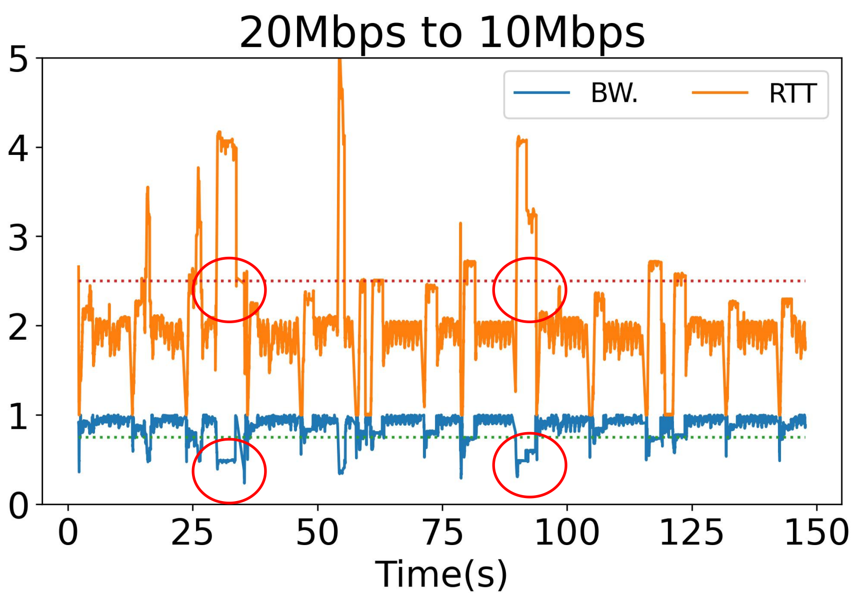
oBBR design:

- ➢ **Adaptive g**
  - Estimate the bottleneck buffer size
  - Adaptively adjust g
- ➢ **Timely bandwidth updates**
  - Sense bandwidth drops accurately
  - Update the bandwidth estimate in time and reversible

Detect the bandwidth drop

- **A single** low-rate sample cannot represent a bandwidth drop
- **Consecutive** low-rate samples more likely signal a bandwidth drop





Timely Bandwidth Updates

Detect the bandwidth drop

- Consecutive low bandwidth samples
- Consecutive high RTT samples

→ Update $\widehat{BtlBW}$

We may still have false positive cases

- Calculate a delivery score of a fixed time interval T

$$score = delivered - 10 \times unacked$$

→ The punishment of packet loss is more weighed than successful delivery

➢ $\widehat{BtlBW}$ is reverted if the score drops

Experimental Setup

- ➢ Evaluate based on QUIC protocol
  - server: ***nginx-quic***
  - client: ***chromium quic_client***
- ➢ In lab environment
  - Control network conditions using ***tc***
- ➢ In the Internet
  - Virginia, USA to Shandong, China
- ➢ Compare to other CCAs
  - CUBIC: loss-based one
  - BBR and its variants (BBRv2, BBR-S, B3R)

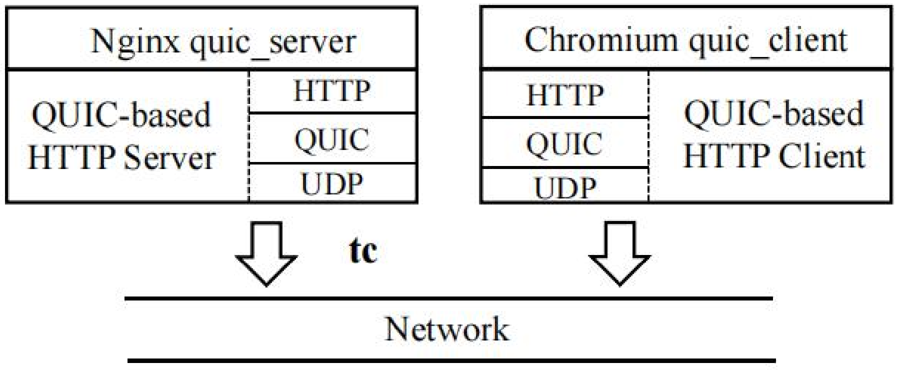
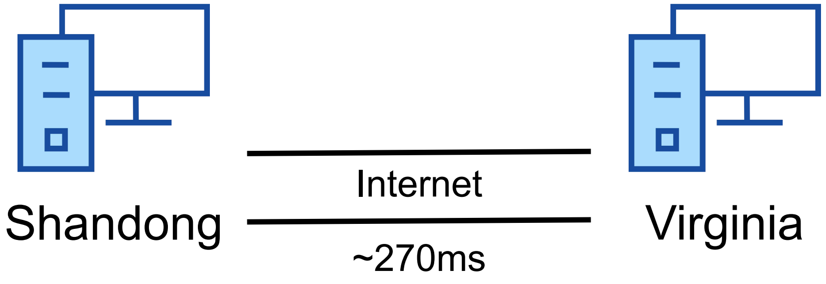


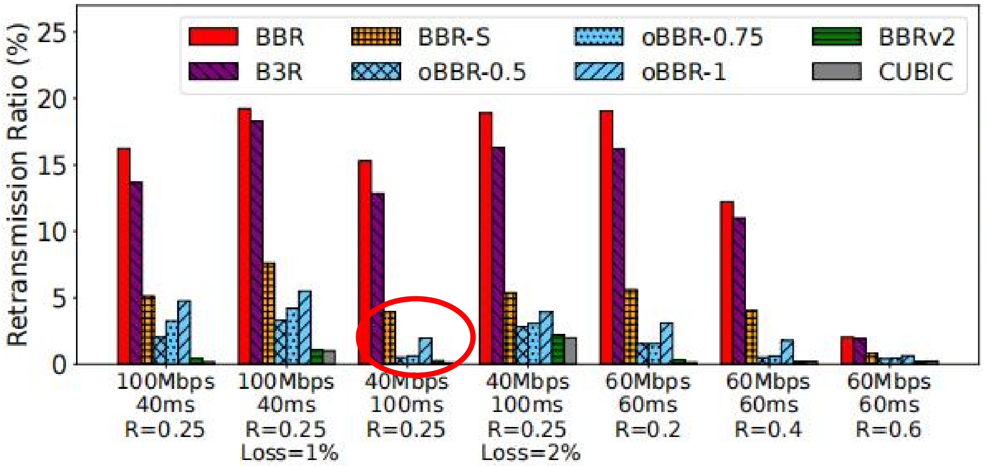
Figure 6: Overview of the testbed

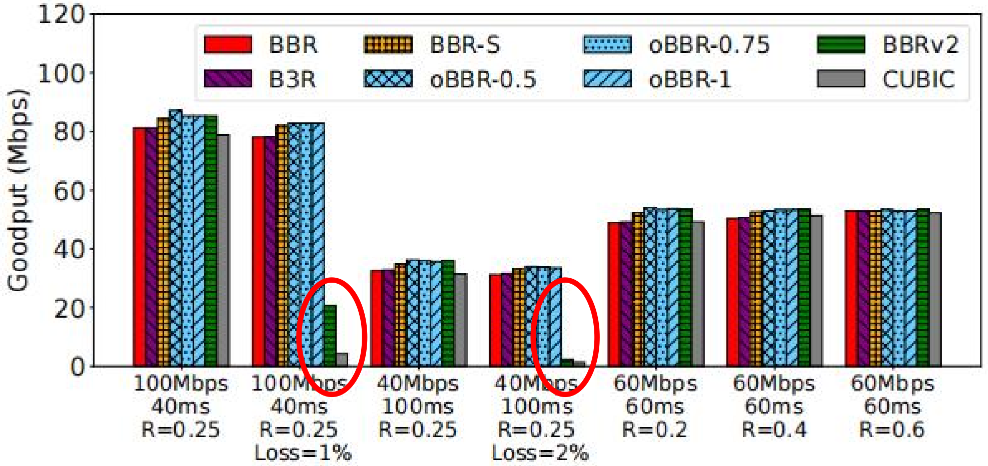


Stable Network Environment

➢ Single flow

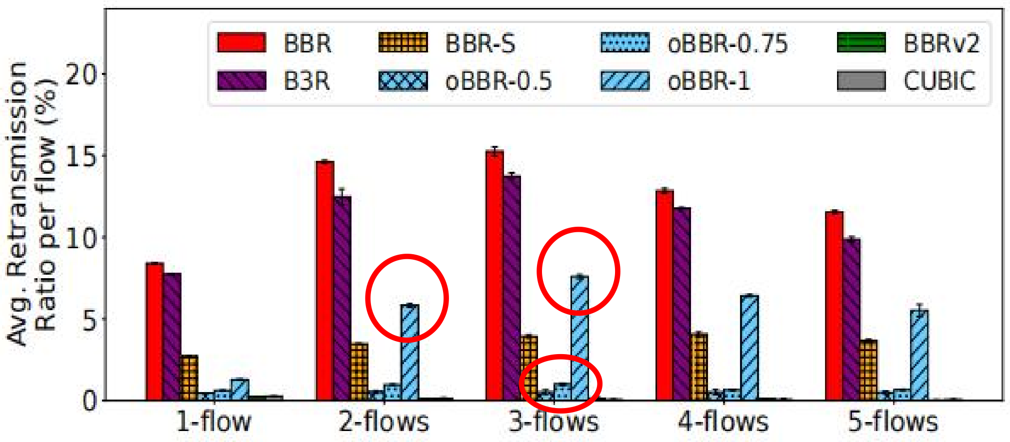
- oBBR reduces the retransmission ratio by up to 30× when compared to BBR
- With 2% packet losses, oBBR achieves 14.65× higher goodput than BBRv2

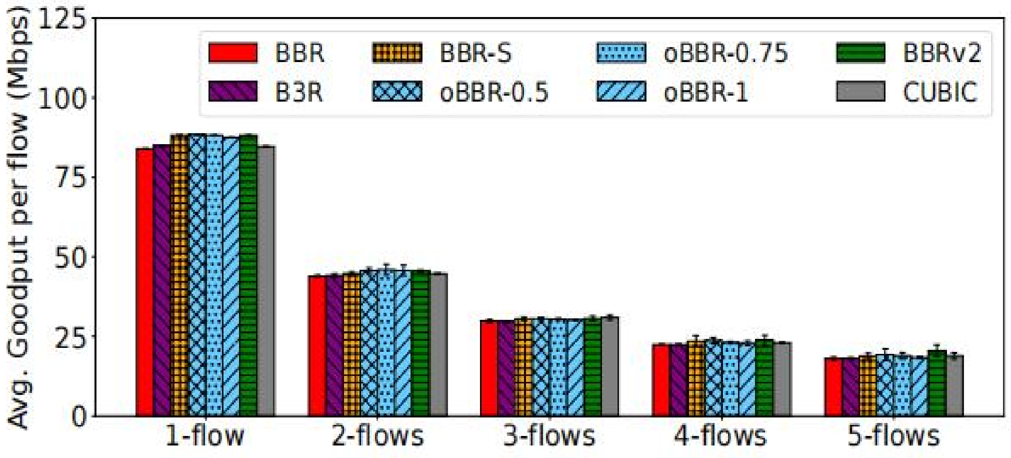




Stable Network Environment

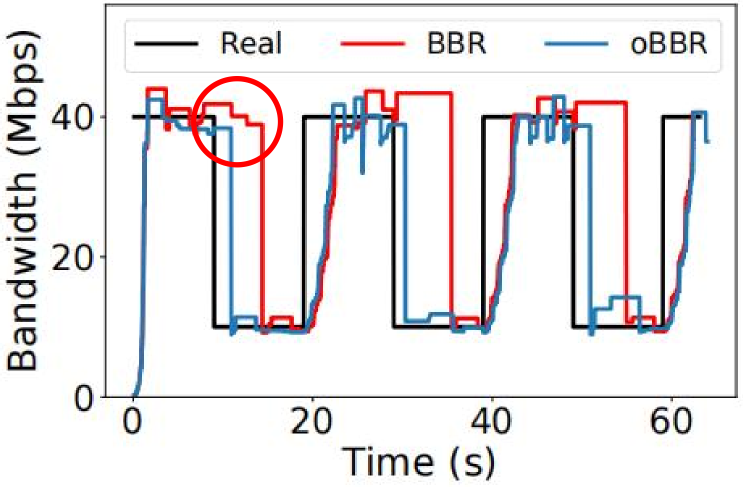
- Multiple flows with the same CCA
  - BBR overestimates bandwidth more severely in multiple flows
  - $\boldsymbol{\mu} = \mathbf{1}$ shows higher retransmissions than the single-flow scenarios
  - $\boldsymbol{\mu} < \mathbf{1}$ gains the lower retransmissions since it leaves a margin in the buffer

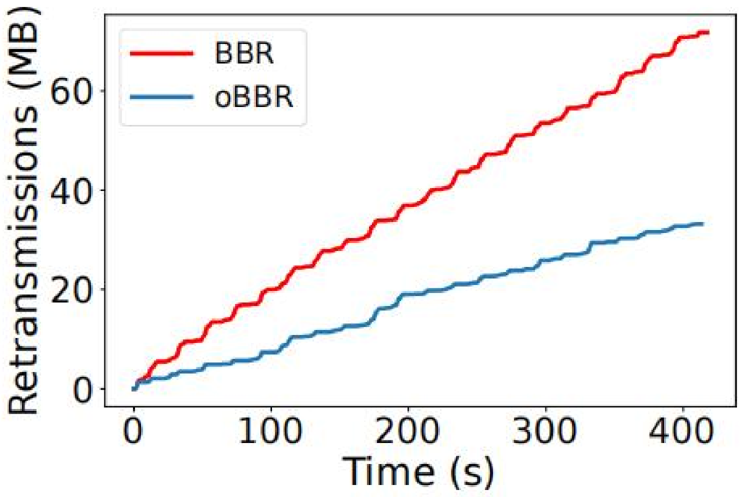




Variable Bandwidth

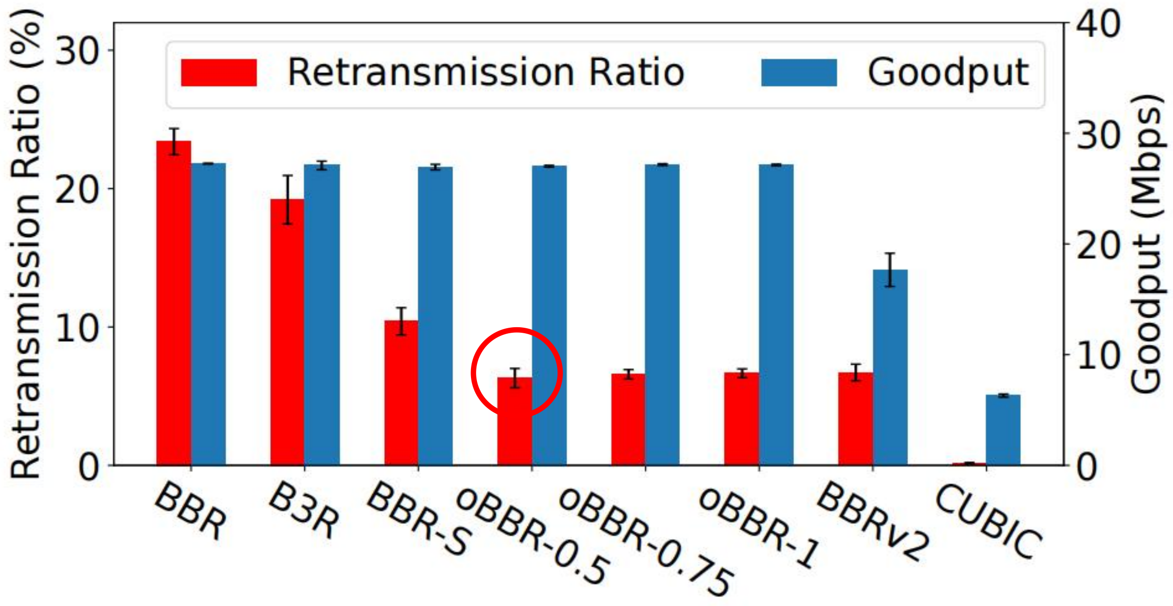
- 40Mbps and 10Mbps: bandwidth switches per 10s
  - oBBR follows the real bandwidth change in ~2s compared to ~6s of BBR
  - BBR spends 2.09× more bandwidth than oBBR on retransmitting lost data





Internet

- Virginia, USA to Shandong, China
- Max egress bandwidth: 80Mbps, Latency: ~270ms
  - The retransmissions of Cubic and BBRv2 are fewer but their throughput suffers
  - oBBR achieves 1.54× higher goodput than BBRv2 and 39.48% fewer retransmissions than BBR-S



Conclusion

- Loss-based CCAs
  - Throughput oscillation and high latency
  - Poor packet loss resistance and low throughput in lossy networks

- BBR Congestion Control
  - High throughput, low latency
  - High resistance to packet losses
  - High retransmissions in shallow-buffered links

- oBBR is designed to help a transmission session reach high throughput and low retransmissions simultaneously, while other CCAs only achieve one of them

- Our prototype is available at https://github.com/bpq233/oBBR

Thanks

Pengqiang Bi
pq.bi@mail.sdu.edu.cn