Experimental Evaluation of BBR Congestion Control

Mario Hock, Roland Bless, Martina Zitterbart

Published at IEEE ICNP 2017, Oct 10–13, Toronto, Canada
Google’s Congestion Control BBR

- Overall objectives:
  - Replace loss-based congestion control
  - High throughput with a small queue
- Model-based approach
- Experimental evaluation based on [1] and Linux 4.9 implementation
- Key findings [2]
  - Model does not work for multiple flows at the bottleneck
  - Massive packet loss in small buffers
  - Unfairness
  - Suppression of loss-based congestion control
So?

What’s wrong with the model?

Network model ok for bottleneck

But used at the sender!

Model lacks dynamics of multiple senders!
Behavior in the Single Flow Case

- Simplified example with a single flow
  - Bottleneck 100Mbit/s, fully utilized

- Flow probes and cannot get higher delivery rate, since bottleneck fully utilized
- Excess data gets queued and removed afterwards
Experimental Evaluation – Single Flow

- 1 Gbit/s bottleneck

Works as expected since the model fits!
Behavior in the Multiple Flows Case

- Simplified example with two flows:
  - Bottleneck 100Mbit/s, fully utilized
  - Each flow sends with 50Mbit/s initially

- Flow probes and actually gets higher delivery rate, although bottleneck fully utilized!

- Windowed maximum filter keeps send rate too high
Behavior in the Multiple Flows Case

- Rate-based approach: amount of inflight data steadily increases
- Bottleneck becomes overloaded
Large buffer ($\geq bdp$)

- BBR operates at its inflight cap (1 to 1.5$bdp$ queued!)

- BBR's Inflight Cap $2 \cdot bdp$

- $bdp$ + Bottleneck Buffer Size

- Operating point BBR

- Operating point CUBIC TCP

- Delivery Rate

- Round-trip Time

- Amount of inflight data
Small buffer ($< bdp$)

- BBR ignores packet loss as congestion signal

![Diagram showing the relationship between operating points for BBR and CUBIC TCP, with annotations for Round-trip Time (RTT) and Delivery Rate vs. Amount of inflight data.](image)
Experimental Evaluation – Setup

- Several experiments with BBR (Linux v4.9) at 1 Gbit/s and 10 Gbit/s

RTT: 20ms

Bottleneck buffers
- Large: 160ms (= 8 bdp)
- Small: 16ms (= 0.8 bdp)

Sender is not application-limited (iperf3)

Repeated every experiment 5 times
Different RTTs, Two Flows – Large buffer

- 2 Flows, same $RTT_{min}$: RTT is doubled $\rightarrow$ BBR queues $1 bdp$

\[ RTT_{min} = 20\text{ms} \]

\[ RTT_{min} = 40\text{ms} \]

\[ RTT_{min} = 80\text{ms} \]
Multiple Flows and Small Buffer (0.8 BDP)

- 6 BBR flows (2 per interface)
- BBR causes massive packet loss
Comparison to CUBIC

Sender’s transmission rate

Throughput (Bit/s)

Time (s)

Three Orders of Magnitude

CUBIC

BBR

Retransmits (log scale)

Retransmits (log scale)

10^8
10^7
10^6
10^5
10^4
10^3
10^2
10^1
10^0

14,826
12,157
6,444
2,288

16,109,114
32,136
52,451
2,353,045

10G large
10G small
1G large
1G small

10G large
10G small
1G large
1G small

IETF 100 - ICCRG, Singapore
Inter-Protocol Fairness – BBR vs. CUBIC

- 1 Gbit/s, 1 BBR flow vs. 1 CUBIC flow
- Small buffers: BBR suppresses loss-based congestion control
- Single BBR flow works as intended

Small Buffer (0.8 BDP)

![Graph showing throughput over time](image-url)
2 BBR vs. 2 CUBIC Flows

- Model mismatch: multiple BBR flows behave more aggressively
- Loss-based congestion control flows get severely suppressed

![Graph showing throughput over time for BBR and CUBIC flows](image)

- Start 2nd BBR flow
- Start 2nd CUBIC flow

Operating point BBR
Intra-Protocol Fairness

- 6 flows (2 per interface), 20ms $RTT_{min}$
- No consistent fairness behavior

**1Gbit/s, large buffer**

**10Gbit/s, large buffer**

**1Gbit/s, small buffer**

**10Gbit/s, small buffer**
RTT Fairness

- 3 concurrent BBR flows with different $RTT_{min} = 20\text{ms}, 40\text{ms}, 80\text{ms}$
- Each BBR flow operates at inflight cap of $2 \, bdp$
- Larger $RTT_{min}$ means more data inflight
  $\rightarrow$ Higher throughput at the bottleneck

![Graphs showing throughput for 10Gbit/s and 1Gbit/s with different RTTs](image)
Summary

- BBR: model-based congestion control
  - Works well if no congestion present (e.g., single flow at the bottleneck)

- Multiple flows: BBR steadily increases the amount of inflight data
  - Large buffers: BBR operates at inflight cap, RTT unfairness
  - Small buffers: high amount of packet losses

- No consistent fairness behavior
- Unfairness to flows with loss-based congestion control, e.g., CUBIC
- BBR is already in use: but probably application-limited

- BBR is still under development
References


BACKUP SLIDES
Multiple Flows – Large buffer

- 1 Gbit/s, 20ms $RTT_{min}$
- RTT is increased to 40ms $\rightarrow$ BBR operates at inflight cap of $2bdp$
Inter-Protocol Fairness – BBR vs. CUBIC

- 1 Gbit/s, 1 BBR flow vs. 1 CUBIC flow
- Large buffers
  - BBR’s inflight cap is larger due to present queuing delay
  - BBR may lose against loss-based congestion control
- Small buffers: BBR suppresses loss-based congestion control

![Large Buffer](image1)

![Small Buffer](image2)
Interprotocol Fairness – BBR vs. CUBIC

1 BBR flow vs. 1 CUBIC flow

large buffer

small buffer
Packet Loss BBR – Outgoing data at sender

10Gbit/s, large buffer

10Gbit/s, small buffer

1Gbit/s, large buffer

1Gbit/s, small buffer
Packet Loss CUBIC – Outgoing data at sender

10Gbit/s, large buffer

1Gbit/s, large buffer

10Gbit/s, small buffer

1Gbit/s, small buffer