A Performance Evaluation of TCP BBRv2 Alpha

Jose Gomez*, Elie Kfoury*, Jorge Crichigno*, Elias Bou-Harb† and Guatam Srivastava ‡
*University of South Carolina, U.S.A
† University of Texas San Antonio, U.S.A
‡Brandon University, Canada

43rd International Conference on Telecommunications and Signal Processing
July 7 – June 9, 2020

Abstract

- The alpha version of Bottleneck Bandwidth and Round-trip Time version 2 (BBRv2) has been recently presented
- BBRv2 aims to mitigate the shortcomings of its predecessor, BBRv1
- Previous studies show that BBRv1 provides a high link utilization and low queuing delay by estimating the available bottleneck bandwidth
- However, its aggressiveness induces unfairness when flows use different congestion control algorithms, such as CUBIC

Abstract

- This paper presents an experimental evaluation of BBRv2 using Mininet
- Results show that the coexistence between BBRv2 and CUBIC improves compared to that of BBRv1-CUBIC
- BBRv2 mitigates the RTT unfairness problem observed in BBRv1
- BBRv2 achieves a better fair share of the bandwidth than its predecessor when network conditions such as bandwidth and latency dynamically change
- Results also indicate that the average flow completion time of concurrent flows is reduced when BBRv2 is used

Introduction

- An essential feature of TCP is the congestion control
- The congestion control algorithm probe for the available capacity of the network to determine how many packets the sender can transmit safely
- Traditional congestion control algorithm such as Reno and CUBIC linearly increases its congestion window size until a loss event occurs
- As a result of a packet loss the sender decreases its congestion window size by a factor of two

Introduction

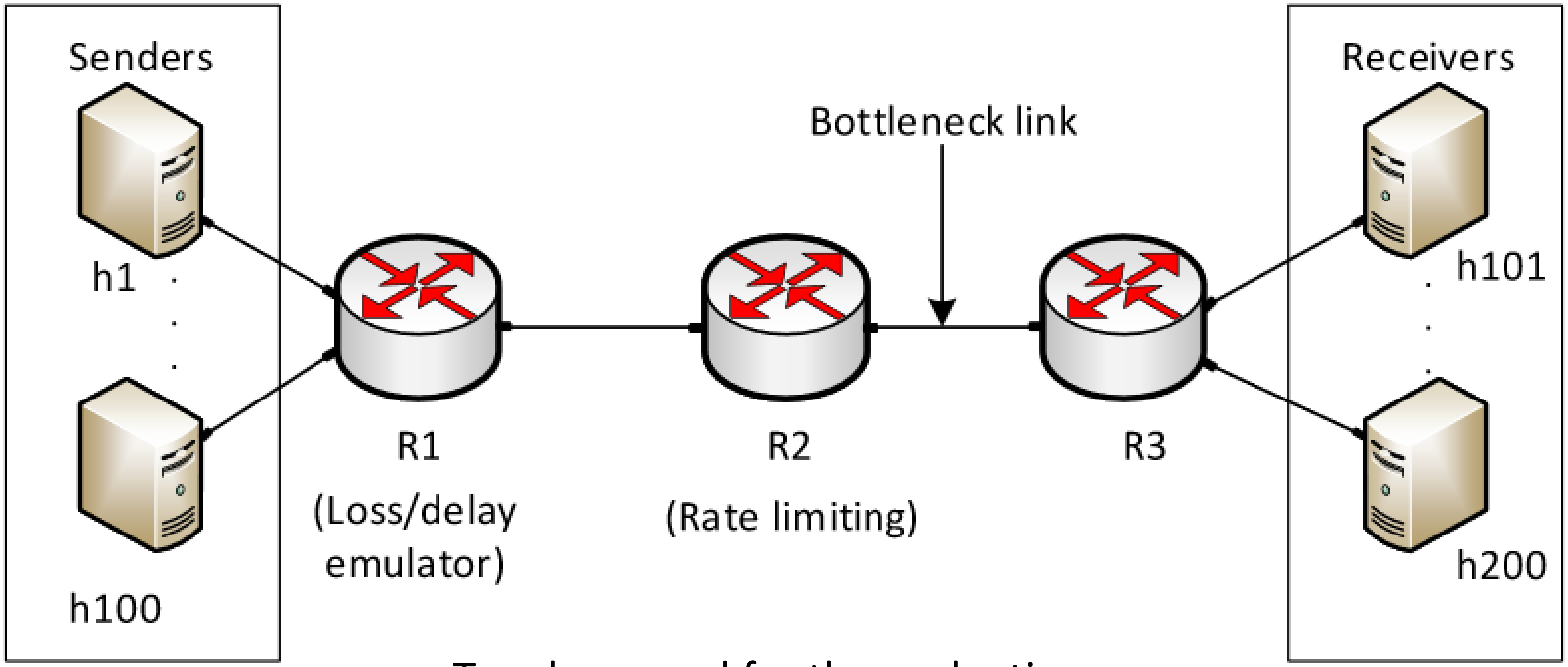
- BBRv1 is loss agnostic and does not follow the AIMD rule.
- Instead, it actively estimates the bottleneck bandwidth and the RTT, which are then used to establish the sending rate
- Although BBRv1 produces higher throughput than traditional loss-based congestion control, it suffers from the RTT unfairness problem

Introduction

- BBRv2 aims to mitigate the limitations of BBRv1
- BBRv2 is a hybrid congestion control algorithm that combines rate-based and model-based approaches
- This means that the algorithm actively measures the bottleneck bandwidth, the RTT, and the packet loss rate to build a model of the end-to-end path
- BBRv1 does not consider packet losses and explicit congestion notification (ECN) as inputs
- BBRv2 uses these variables to estimate the bandwidth-delay product (BDP) and sending rate

Experimental Setup

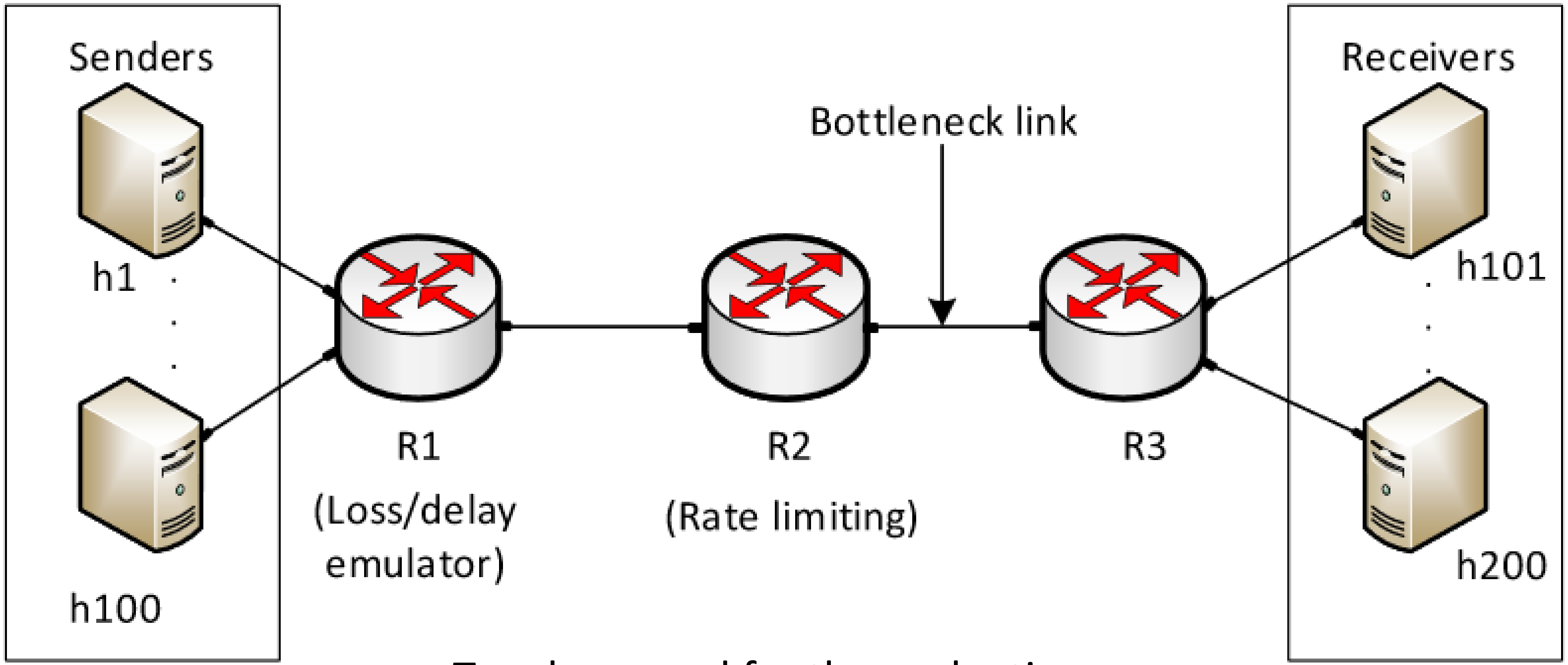
- The network used to conduct the experiments consists of 100 senders and 100 receivers
- Experiments were conducted using Mininet
- The CPU usage was kept below prudent levels



Topology used for the evaluations

Experimental Setup

- **Loss/delay emulation**: The router R1 is used to inject delay and packet losses using Network Emulator (NetEm)
- **Rate limitation and buffer size**: Router R2 uses the Token Bucket Filter (TBF) to emulate a bottleneck by limiting the link rate



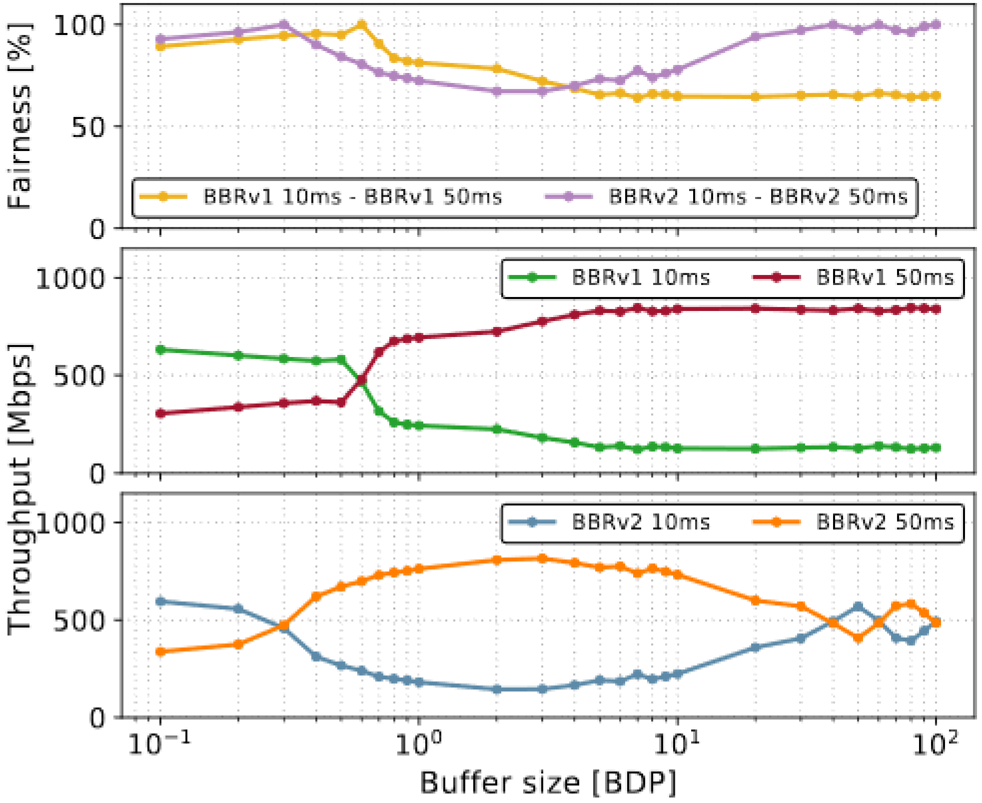
Topology used for the evaluations

Results and Evaluations

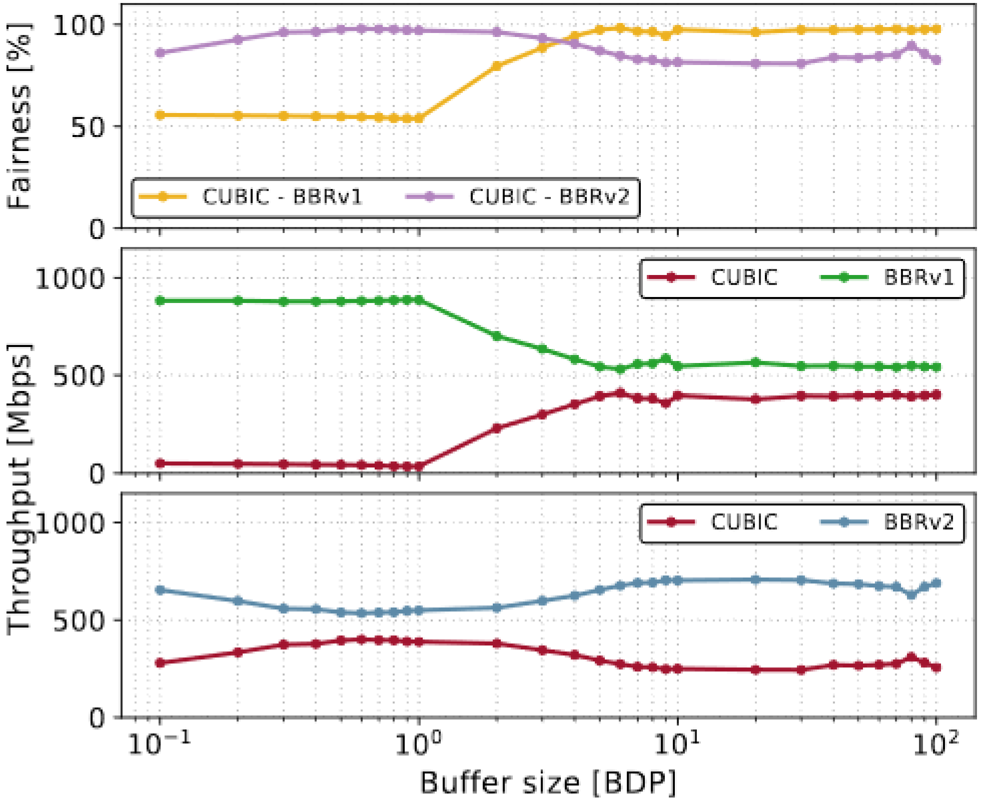
- Given a network condition with specific parameters such as the buffer size and RTT, the experiments were repeated 10 times and the results averaged
- The experiments consisted in measuring the:
  - Round-trip time unfairness
  - Accumulating Effects
  - Fairness under changing network conditions
  - Flow completion time

Results and Evaluations

- Round-trip time unfairness and coexistence between CUBIC



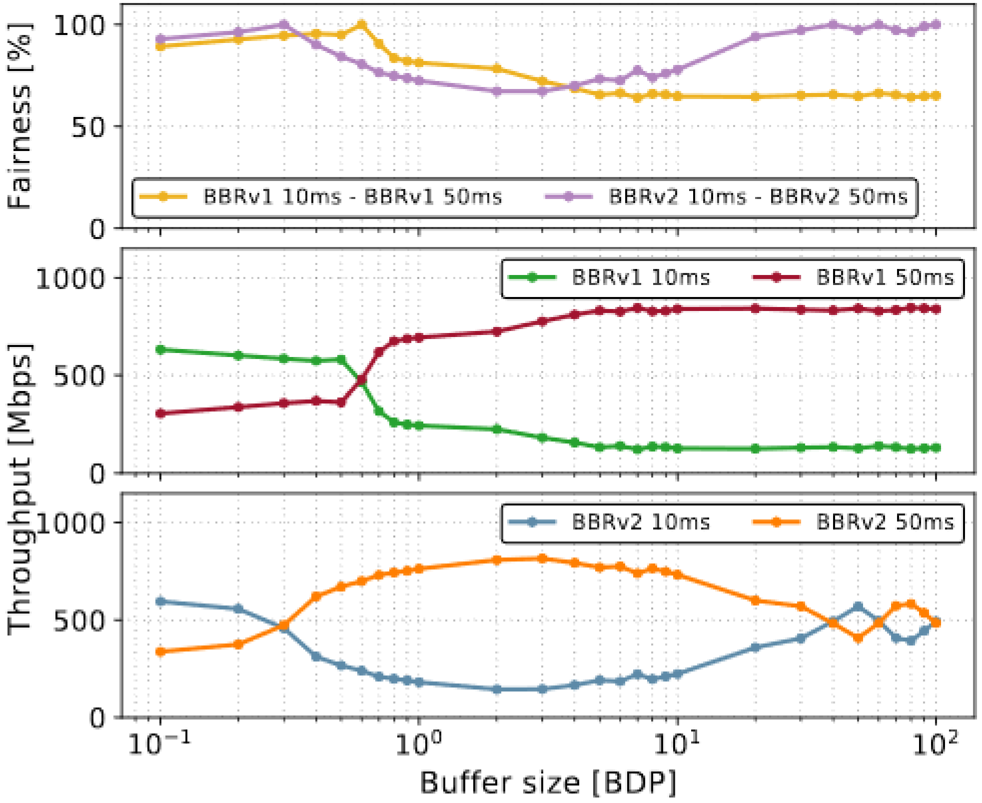
(a) RTT Unfairness



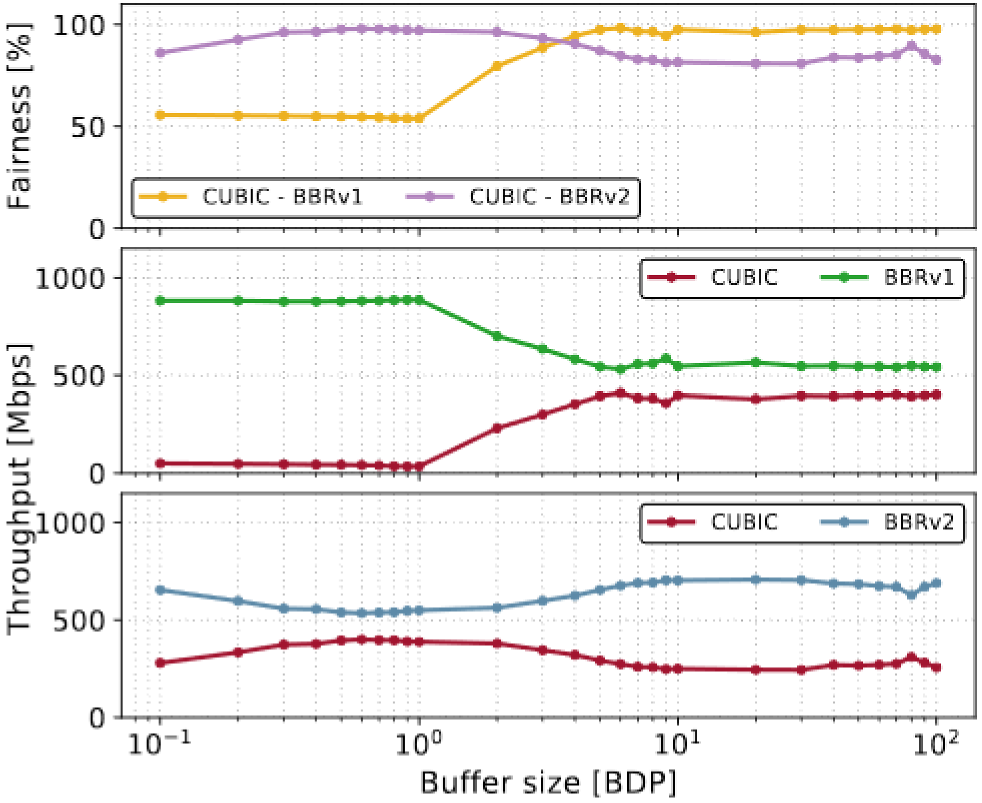
(b) CUBIC vs BBRv1/BBRv2

Results and Evaluations

- Round-trip time unfairness and coexistence between CUBIC



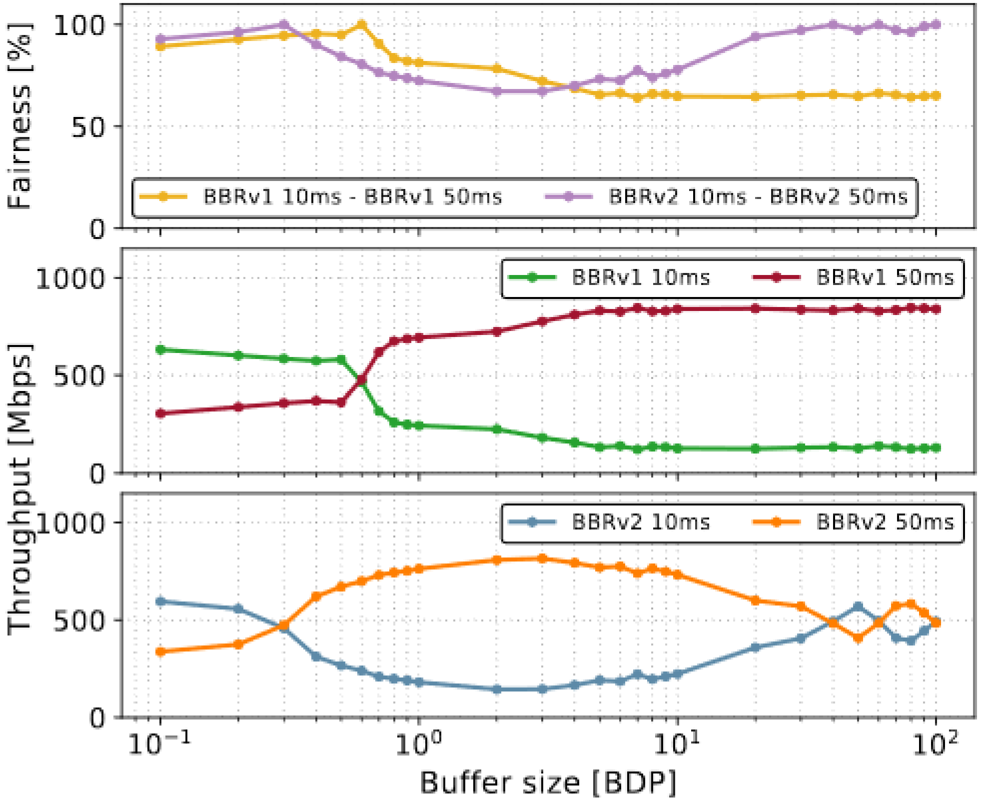
(a) RTT Unfairness



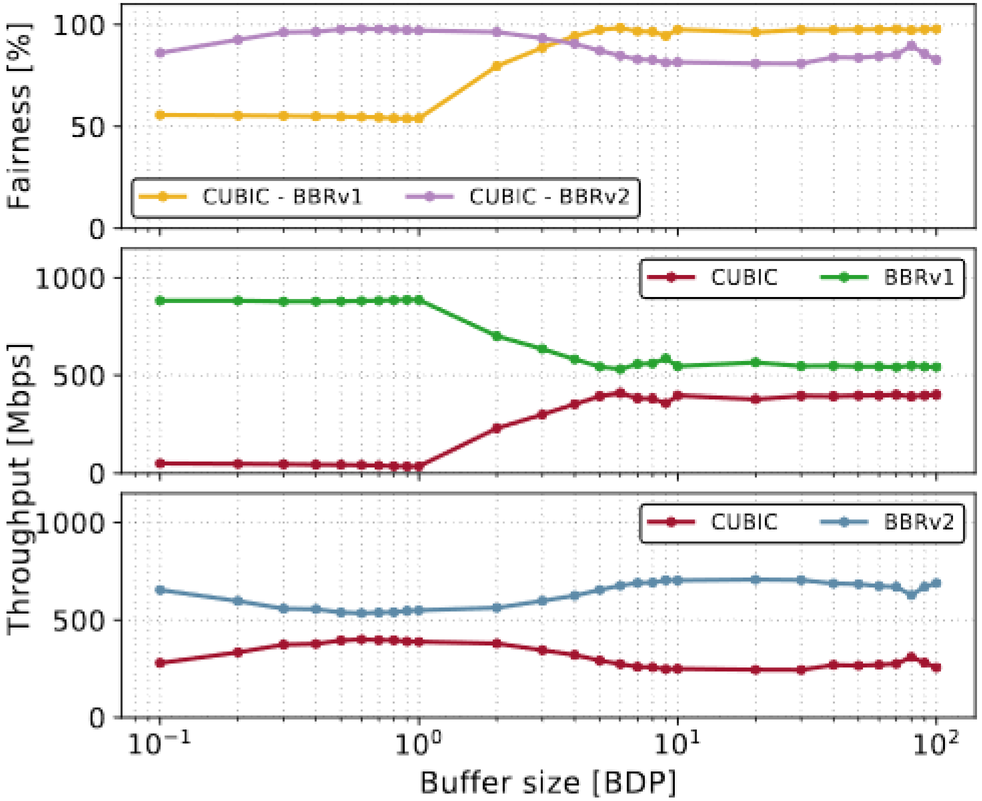
(b) CUBIC vs BBRv1/BBRv2

Results and Evaluations

- Round-trip time unfairness and coexistence between CUBIC



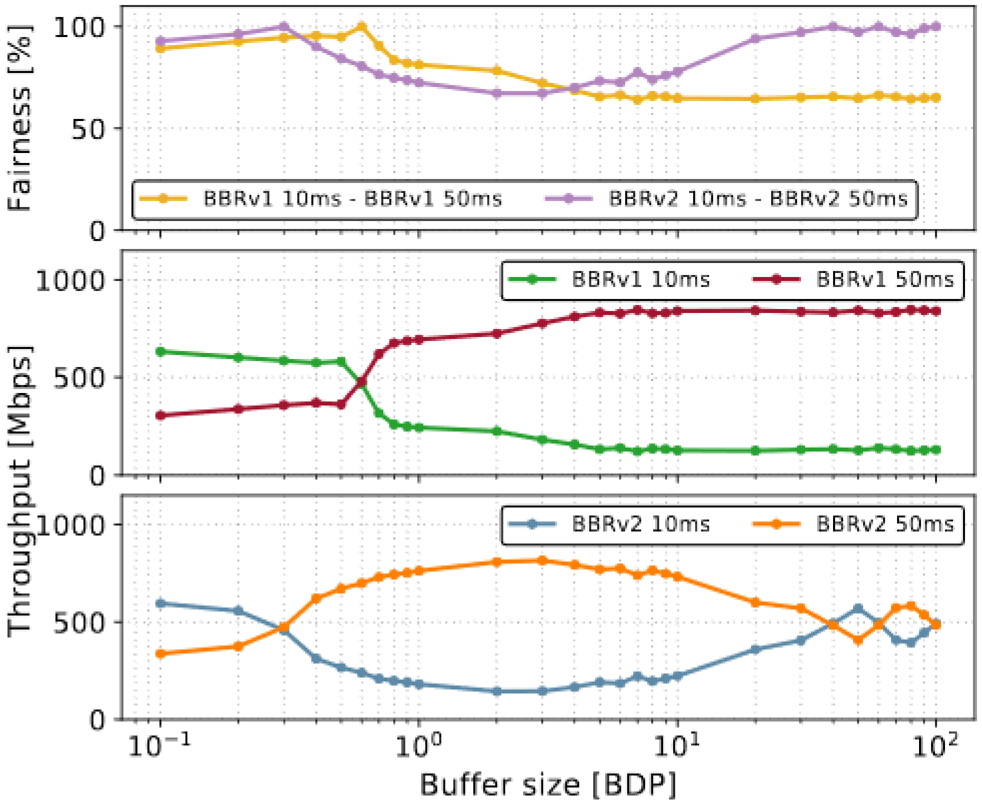
(a) RTT Unfairness



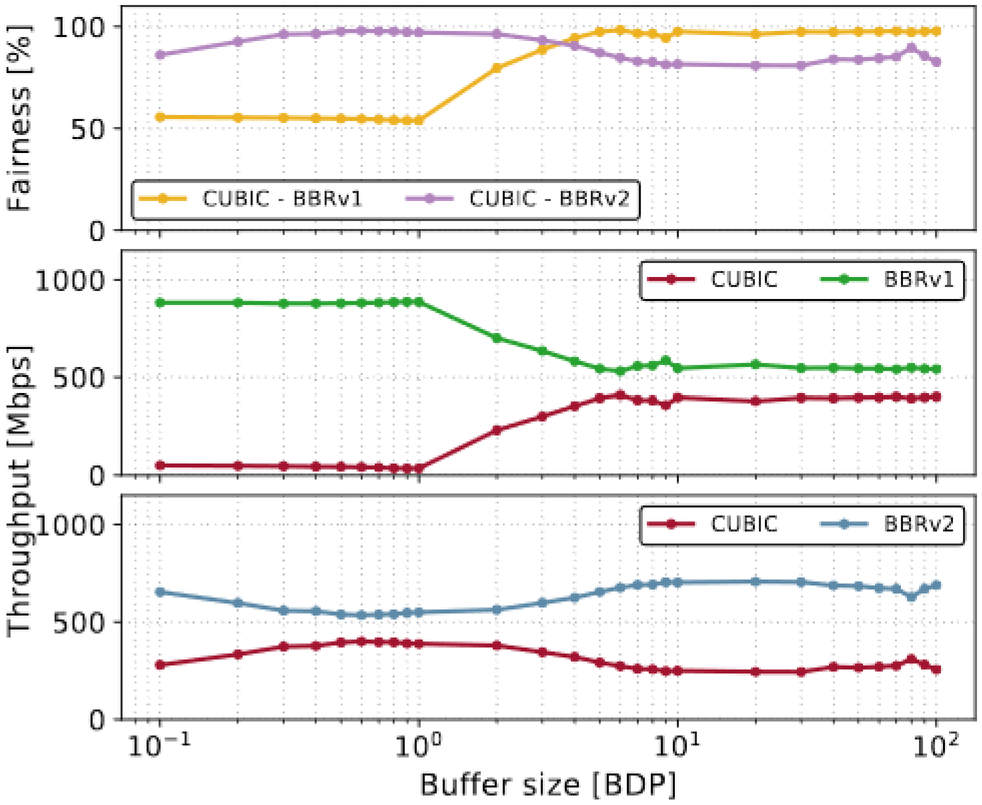
(b) CUBIC vs BBRv1/BBRv2

Results and Evaluations

- Round-trip time unfairness and coexistence between CUBIC



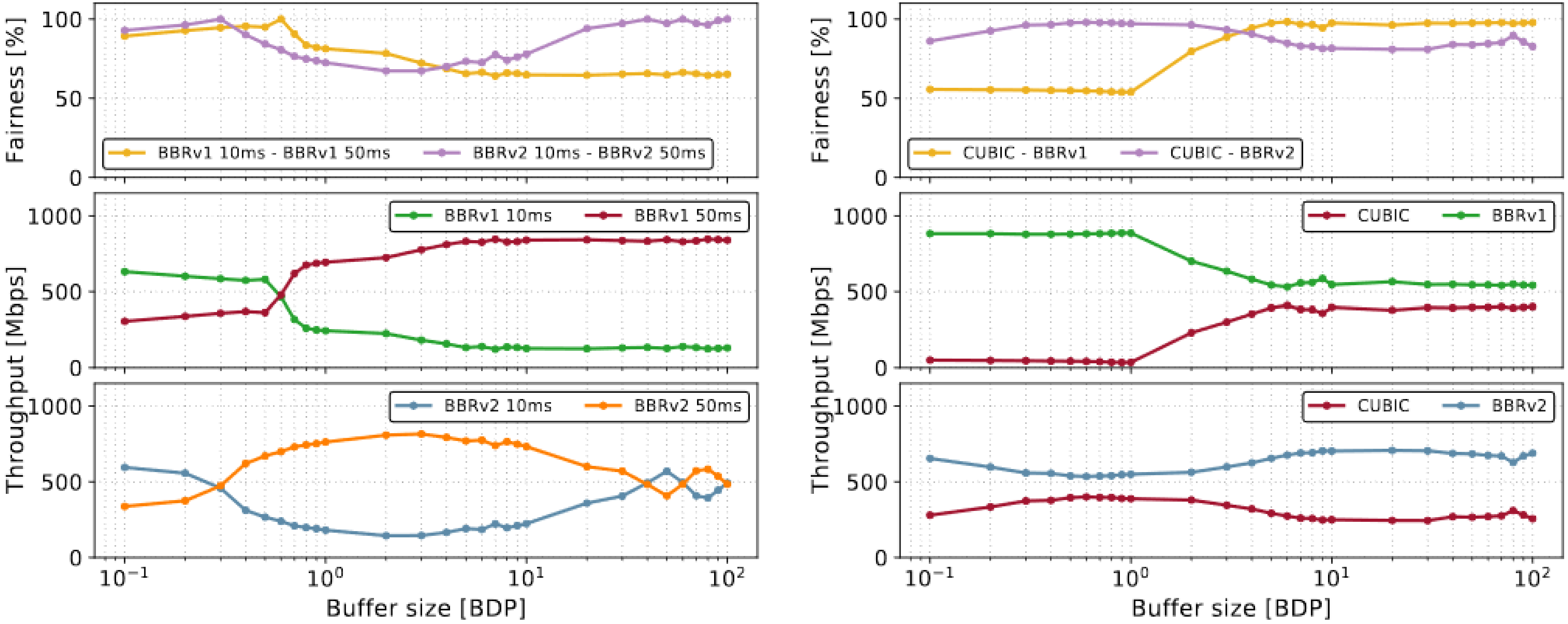
(a) RTT Unfairness



(b) CUBIC vs BBRv1/BBRv2

Results and Evaluations

- Round-trip time unfairness and coexistence between CUBIC

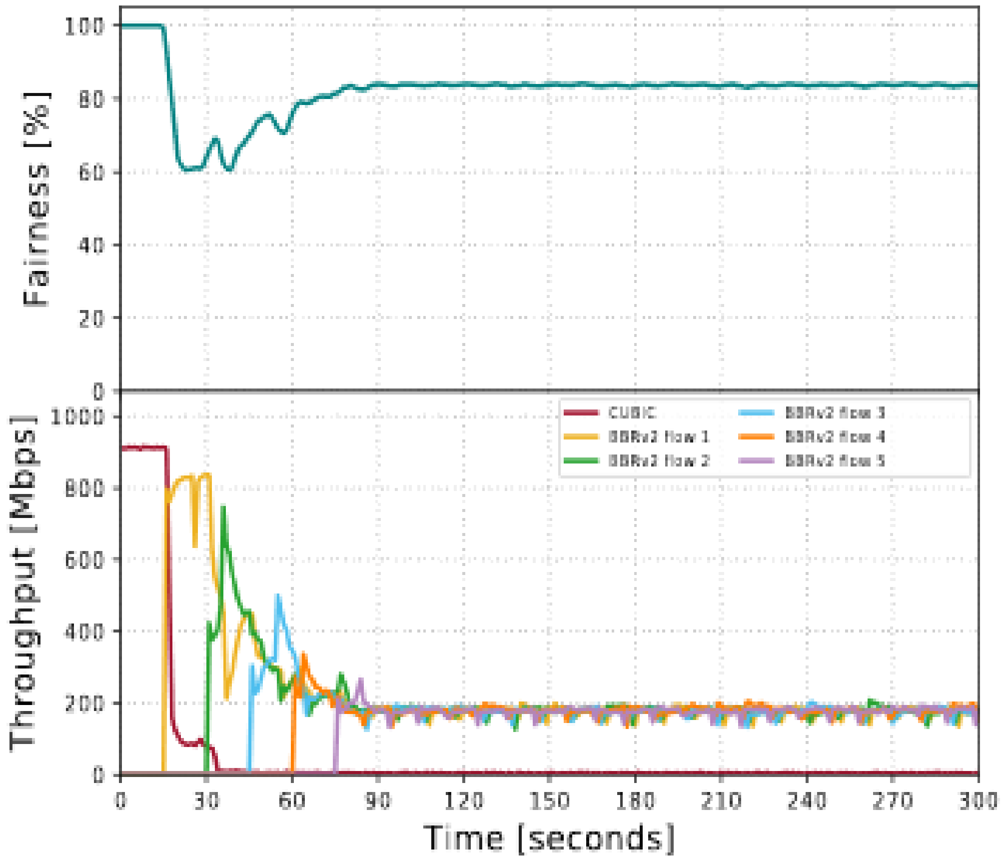


(a) RTT Unfairness

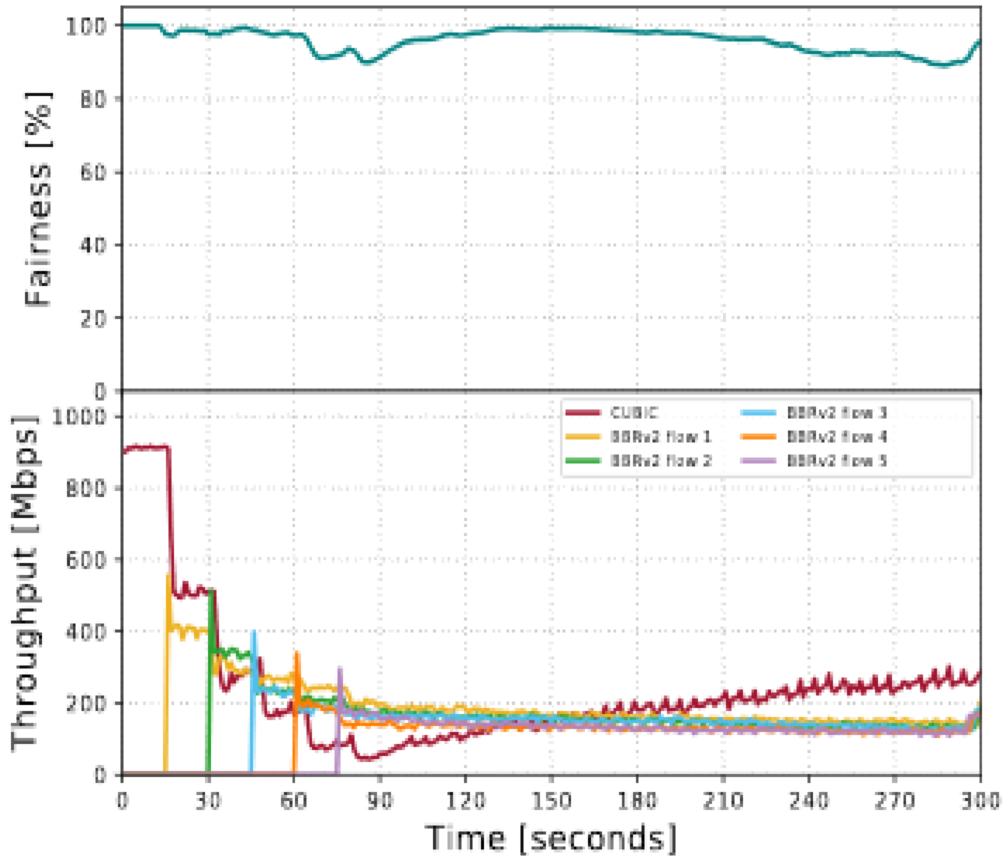
(b) CUBIC vs BBRv1/BBRv2

Results and Evaluations

- Accumulating Effects



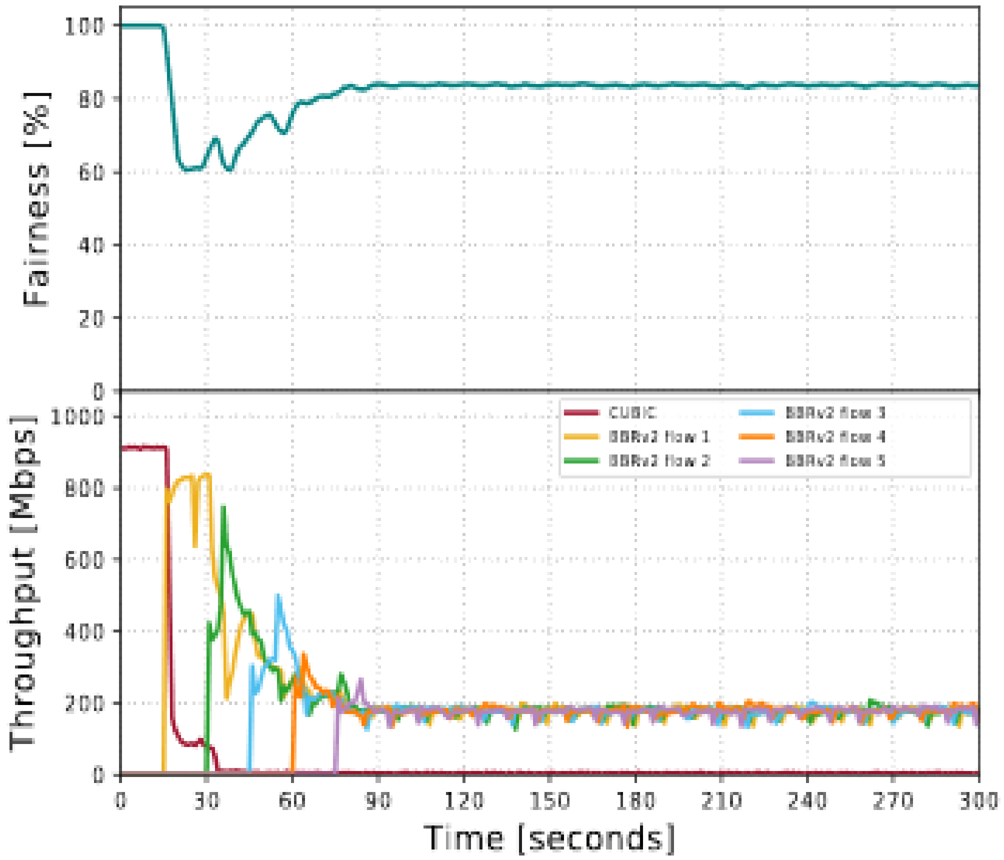
(a) A CUBIC flow with five subsequent BBRv1 flows



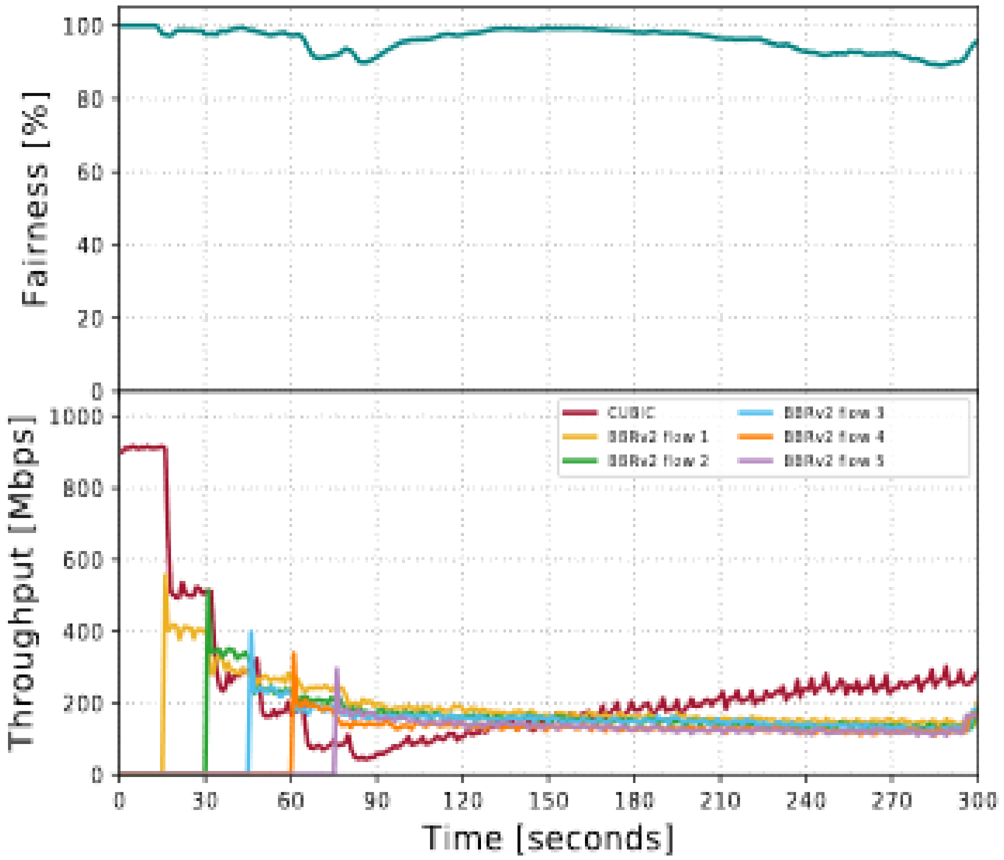
(b) A CUBIC flow with five subsequent BBRv2 flows

Results and Evaluations

- Accumulating Effects



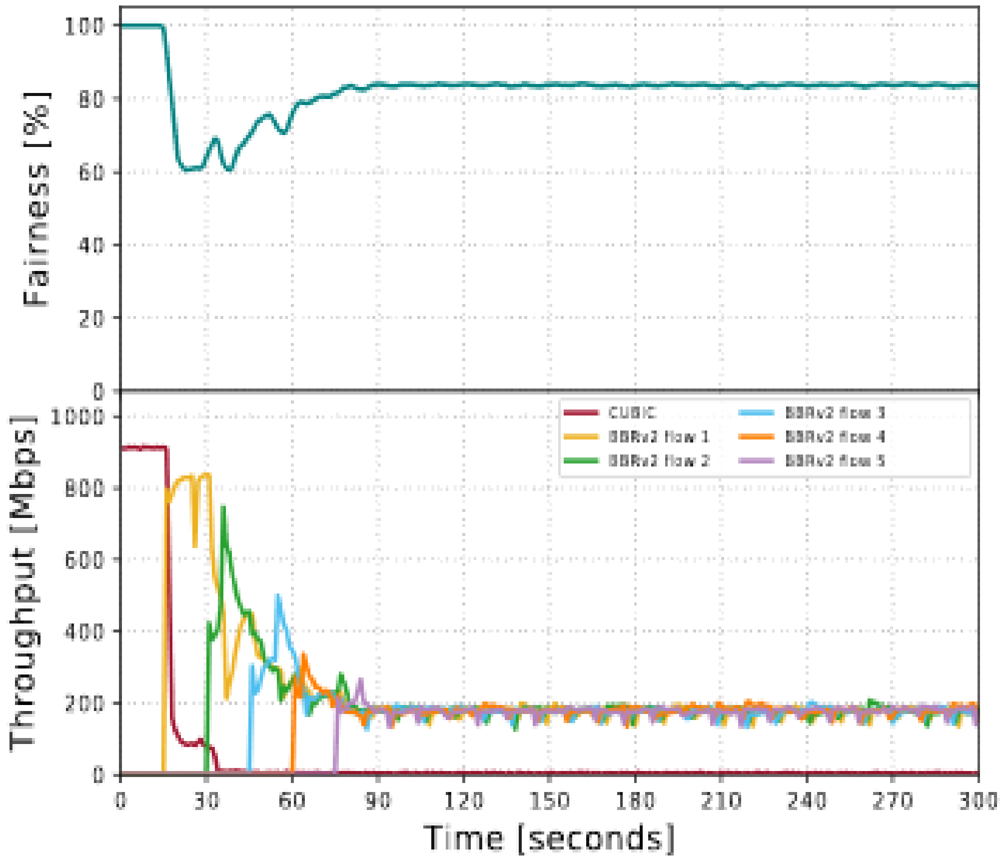
(a) A CUBIC flow with five subsequent BBRv1 flows



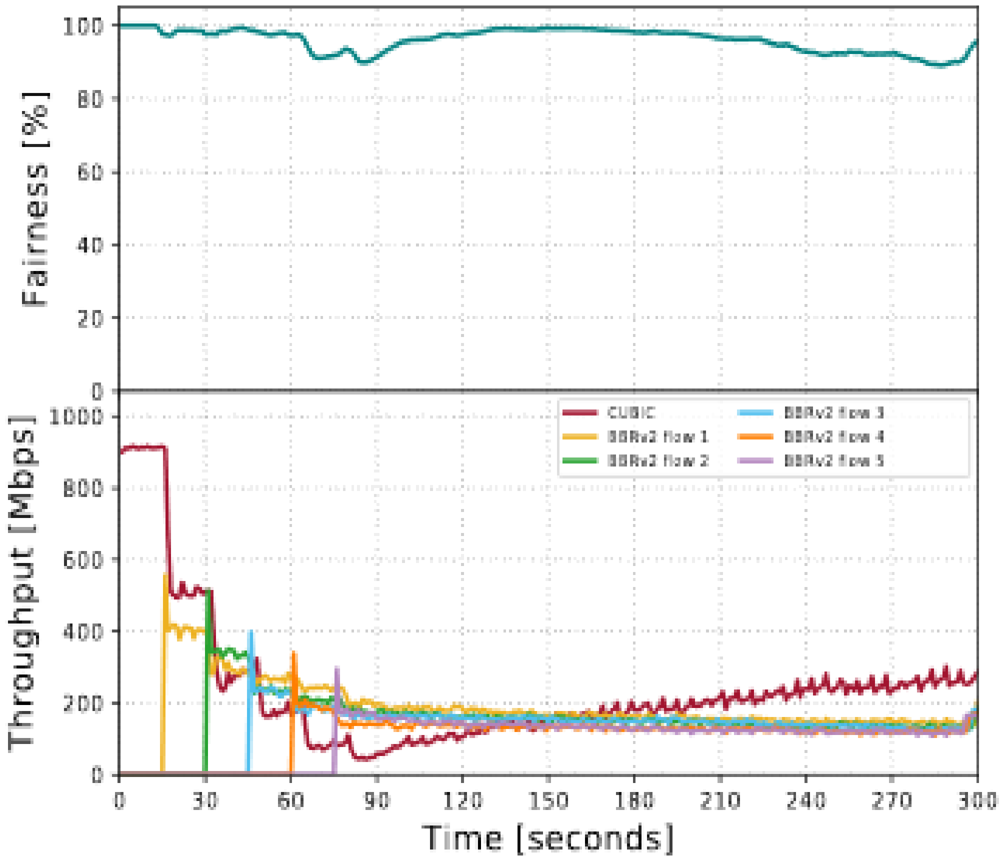
(b) A CUBIC flow with five subsequent BBRv2 flows

Results and Evaluations

- Accumulating Effects



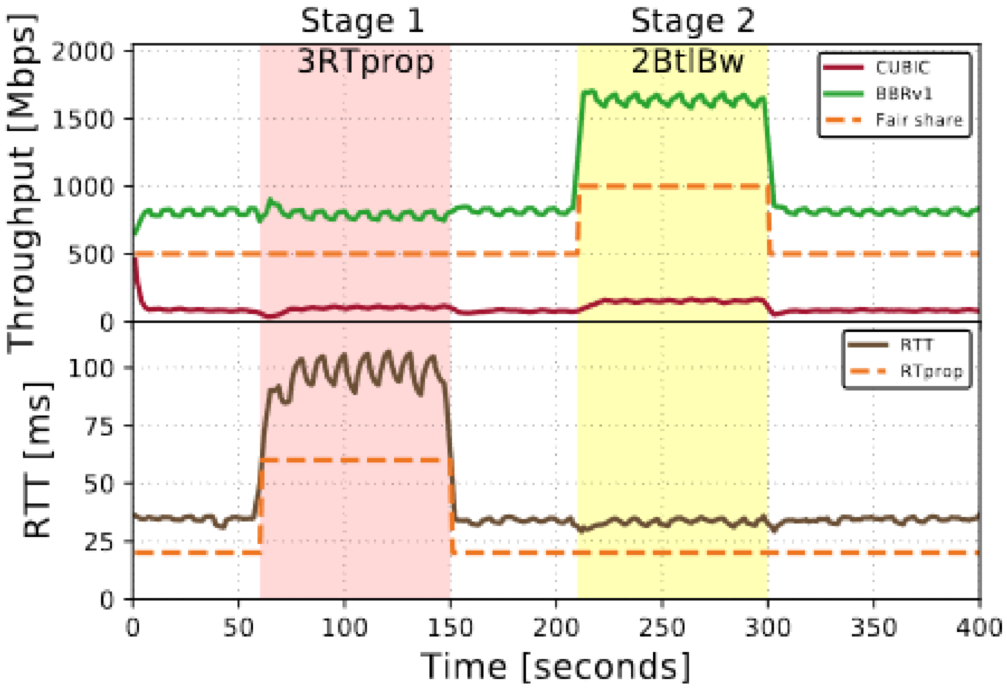
(a) A CUBIC flow with five subsequent BBRv1 flows



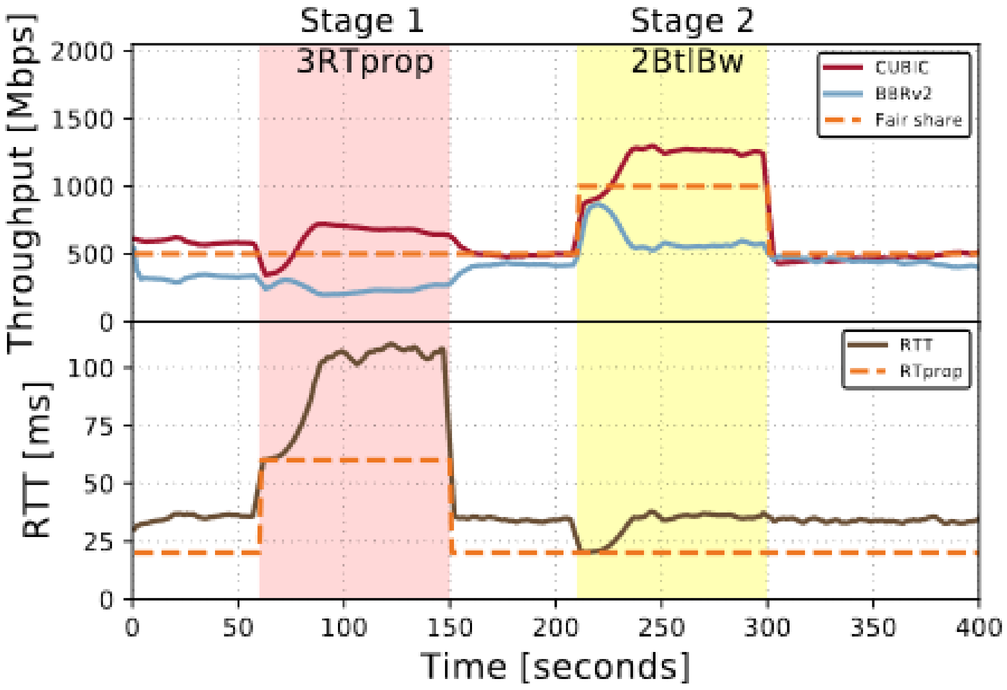
(b) A CUBIC flow with five subsequent BBRv2 flows

Results and Evaluations

- Changing network conditions



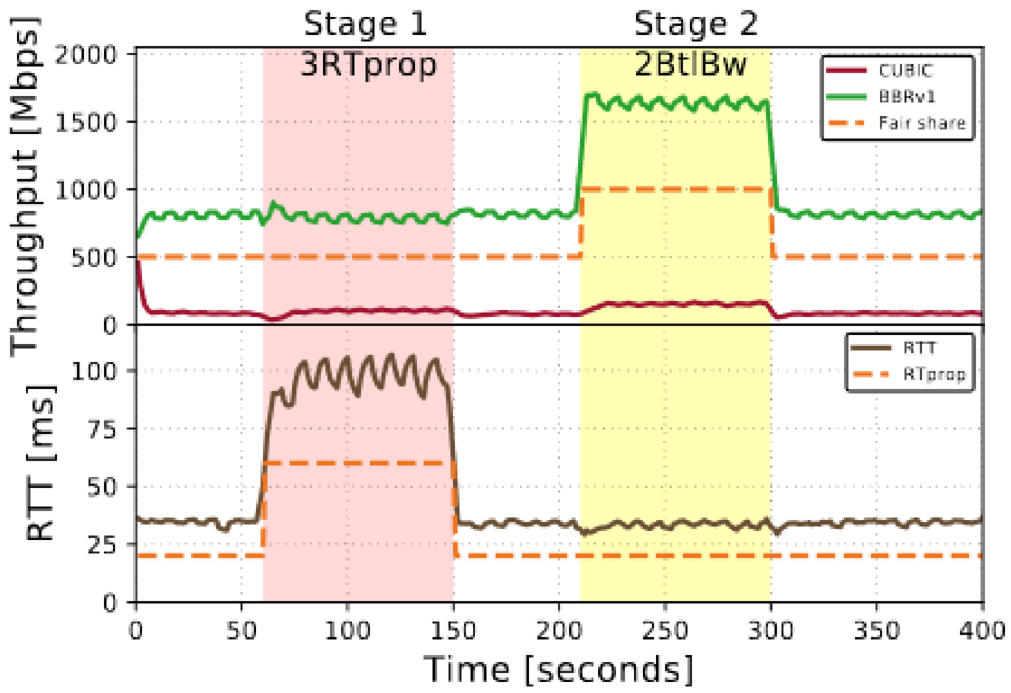
(a) CUBIC and BBRv1 flows



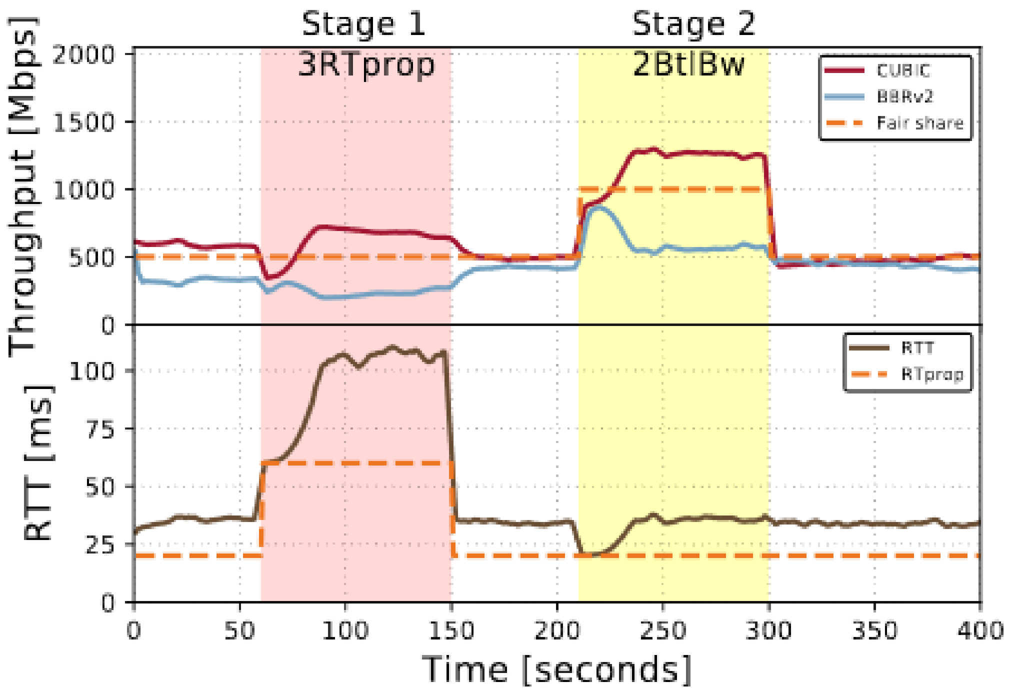
(b) CUBIC and BBRv2 flows

Results and Evaluations

- Changing network conditions



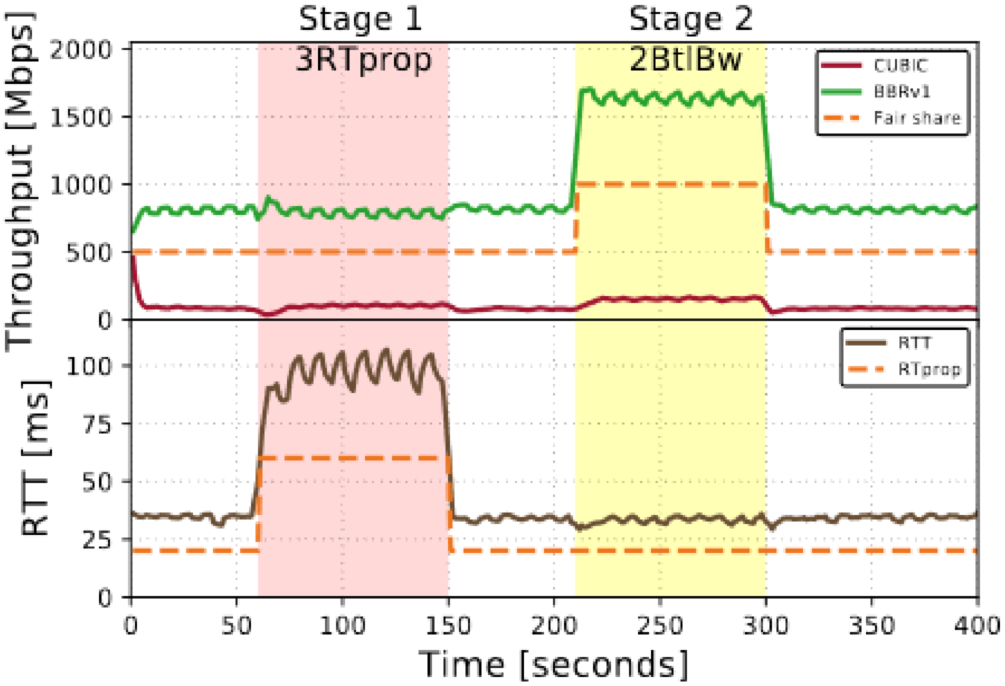
(a) CUBIC and BBRv1 flows



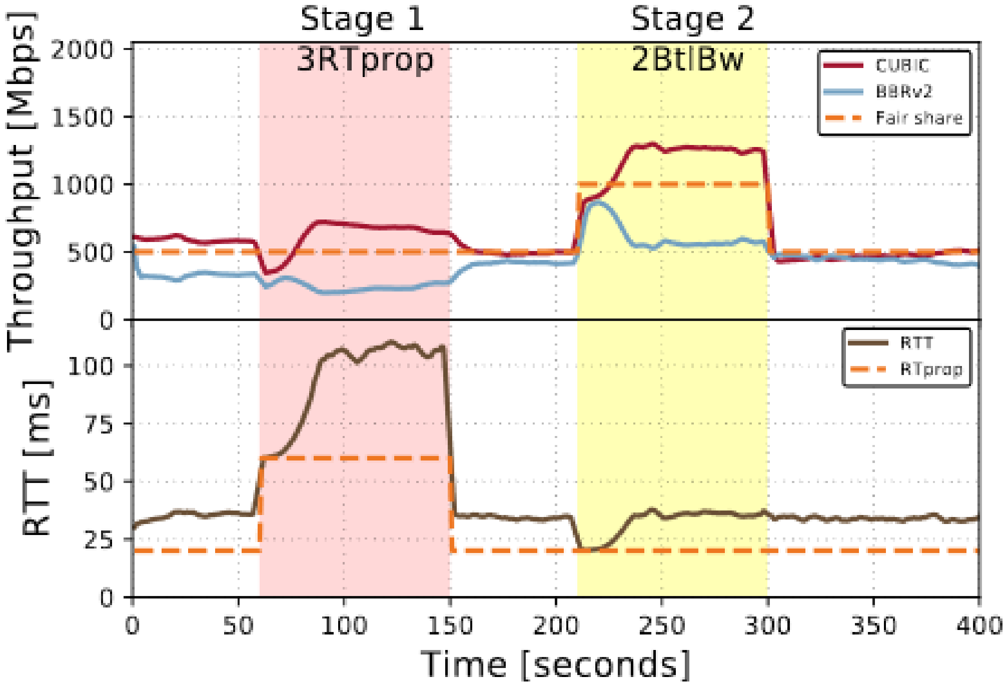
(b) CUBIC and BBRv2 flows

Results and Evaluations

- Changing network conditions



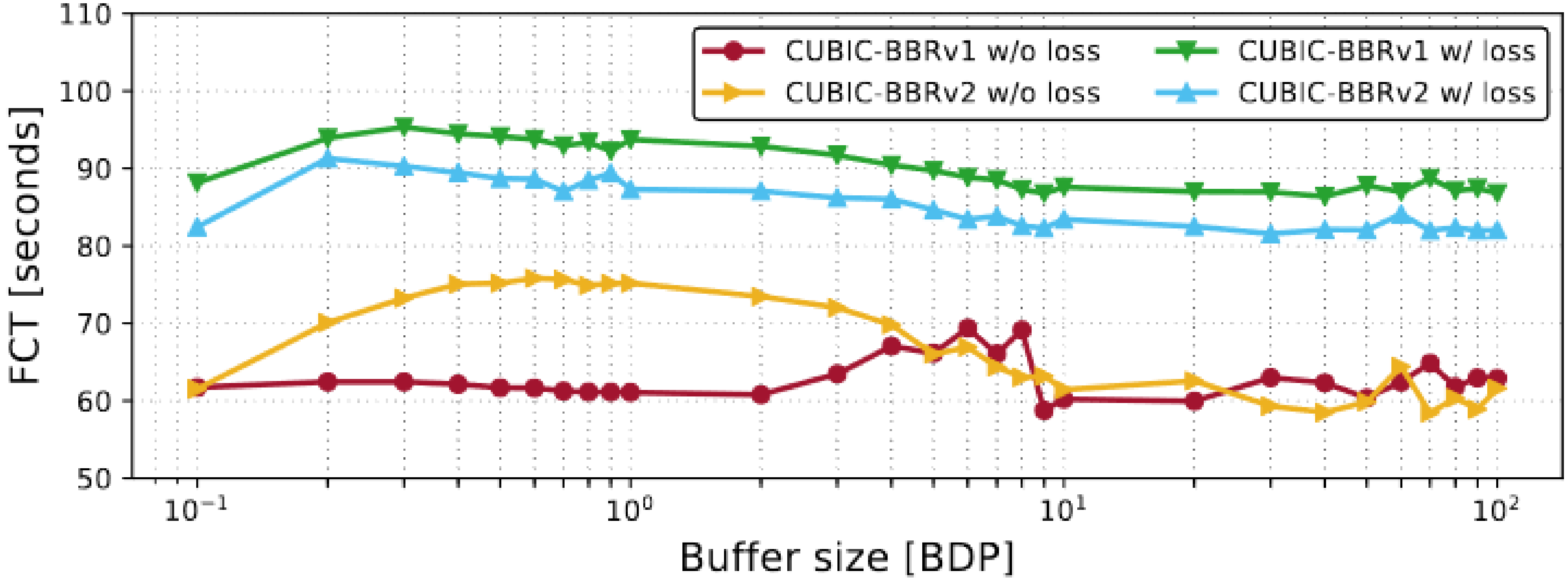
(a) CUBIC and BBRv1 flows



(b) CUBIC and BBRv2 flows

Results and Evaluations

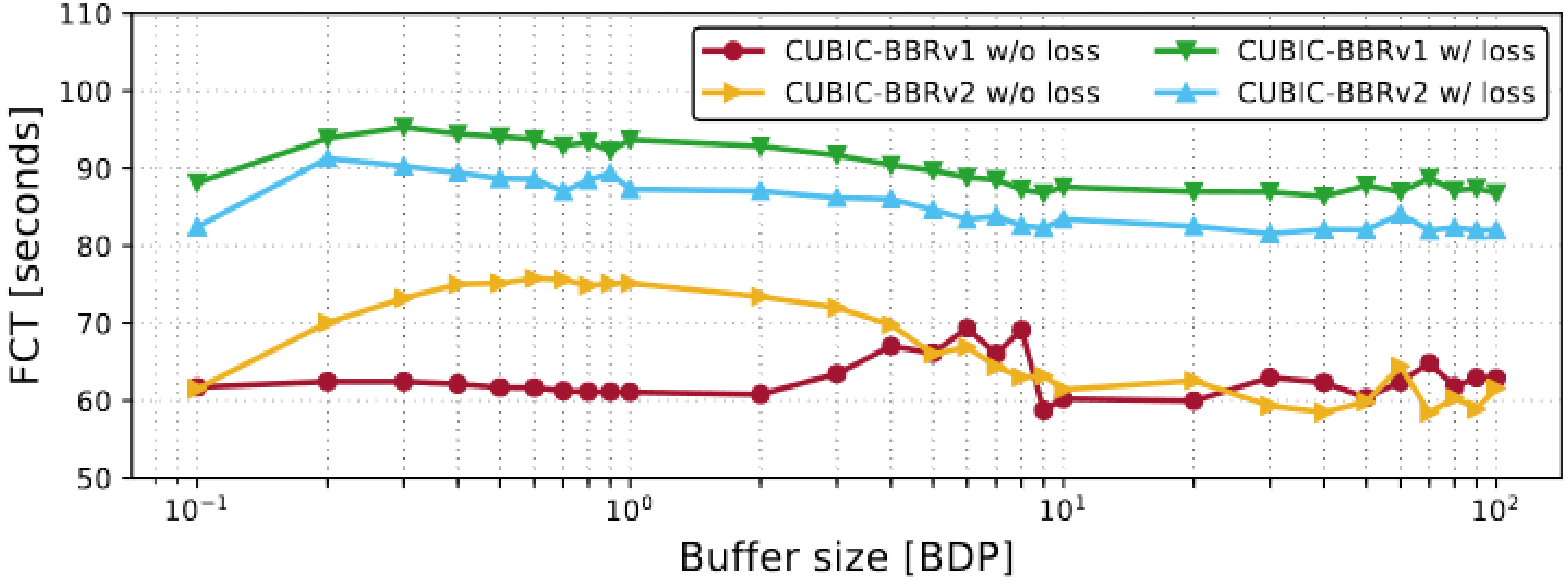
- Flow completion time



Average flow completion time of 100 flows as a function of the buffer size

Results and Evaluations

- Flow completion time



Average flow completion time of 100 flows as a function of the buffer size

Conclusions

- Despite its success in improving the throughput, BBRv1 presented some issues, including the poor coexistence with traditional congestion control algorithms such as CUBIC
- In this context, BBRv2 has been proposed to address such issues
- Results show that BBRv2 presents a better coexistence with CUBIC flows with respect to its predecessor,BBRv1
- It is also reported that BBRv2 mitigates the RTT unfairness problem observed in BBRv1
- Finally, results also indicate that the average flow completion time of concurrent flows is reduced when BBRv2 is used in the presence of packet loss

- Thank you!