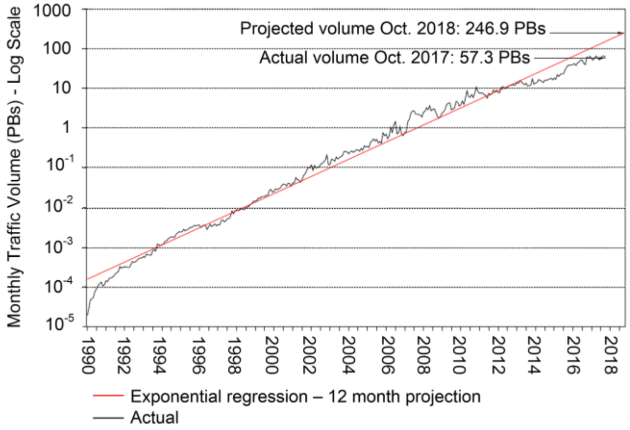
IMPACT OF SEGMENT SIZE AND PARALLEL STREAMS ON TCP BBR

Jorge Crichigno, Zoltan Csibi, Elias Bou-Harb, Nasir Ghani
University of South Carolina, USA

Motivation

- Science and engineering applications are generating data at an unprecedented rate
- From large facilities to portable devices, these instruments produce hundreds of terabytes in short periods of time
- Professionals rely on networks to move data between sensing locations, storage, and computing systems

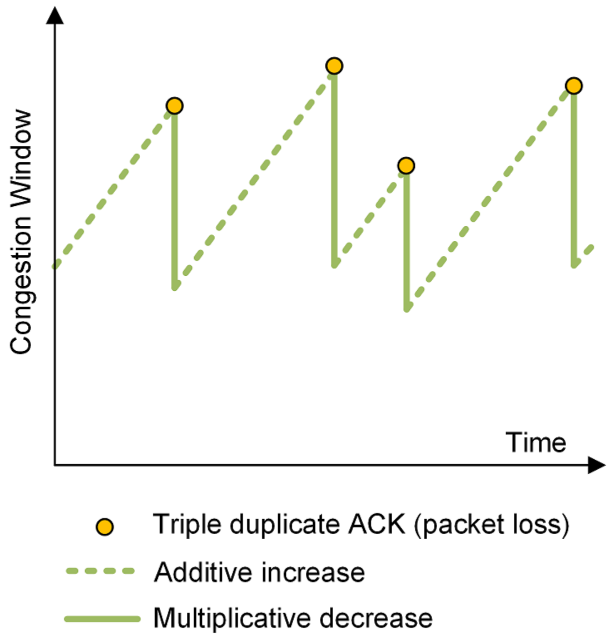
Monthly traffic volume over ESnet

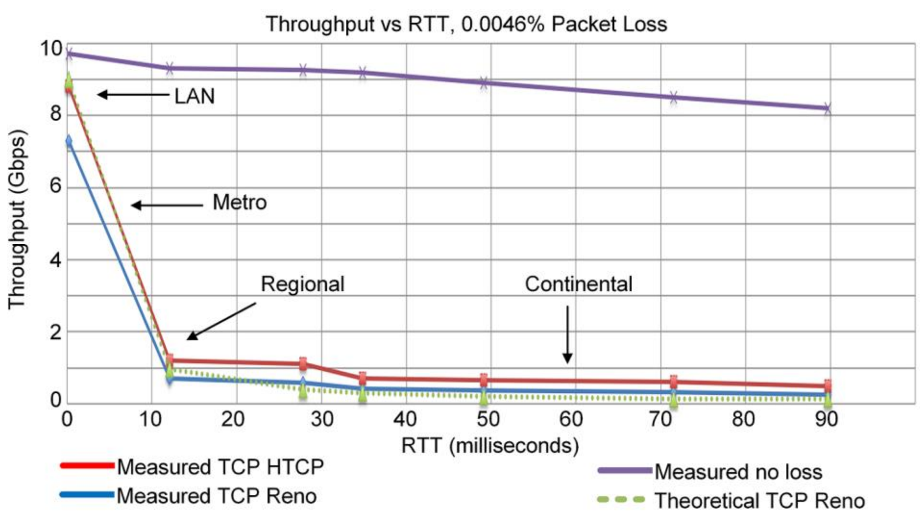


1. Very Large Array (VLA): http://www.vla.nrao.edu/
2. Portable DNA and RNA sequencing device: https://nanoporetech.com/
3. ESnet: https://www.es.net/

Motivation

- At the transport layer, TCP is the protocol used for reliable data transfers
- A major issue in high-throughput high-latency (WANs) networks is the packet loss
- Once a packet loss is detected, TCP's *congestion control algorithm* reacts by decreasing the sending rate





Data transfer between two devices connected by a 10 Gbps path

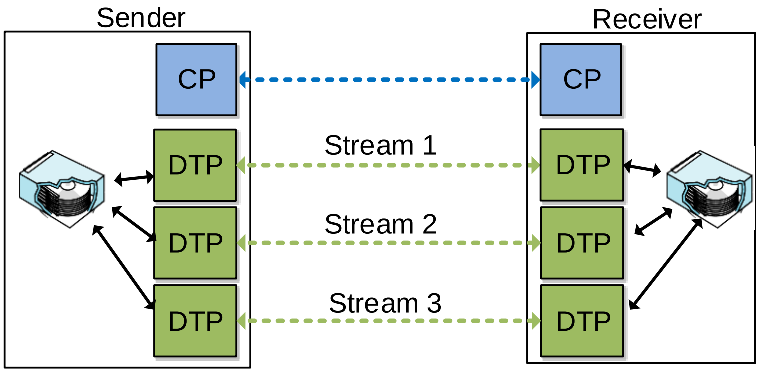
BBR Brief Overview

- TCP BBR has been recently proposed as a congestion control algorithm[1]
- BBR represents a disruption from the window-based loss-based congestion control used during the last decades[2]
- BBR uses ‘pacing’ to try to match the bottleneck rate

1. N. Cardwell, Y. Cheng, C. Gunn, S. Yeganeh, V. Jacobson, “BBR: congestion-based congestion control,” *Communications of the ACM*, vol 60, no. 2, pp. 58-66, Feb. 2017.
2. https://www.thequilt.net/wp-content/uploads/BBR-TCP-Opportunities.pdf

MSS and Parallel Streams

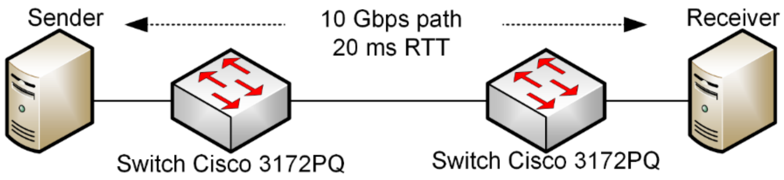
- Two of the main features impacting big flows
  - Maximum segment size (MSS)
  - Parallel streams
- A large MSS produces a faster recovery after a packet loss
- Opening parallel connections essentially creates a large virtual MSS on the aggregate connection



CP: Control process
DTP: Data transfer process

Scenario

- This work studies the impact of MSS and parallel streams on the performance of congestion control algorithms
- Sender/receiver connected by a 10 Gbps path, 20 ms RTT, running CentOS 7
- netem[1] used to adjust loss rate (Linux network emulator software)
- At 20 ms RTT, throughput already collapses when subject to a small loss rate

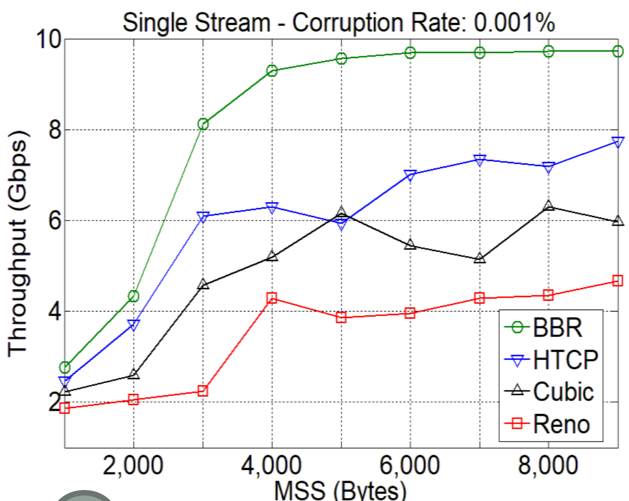


1. Network emulator (netem): https://wiki.linuxfoundation.org/networking/netem

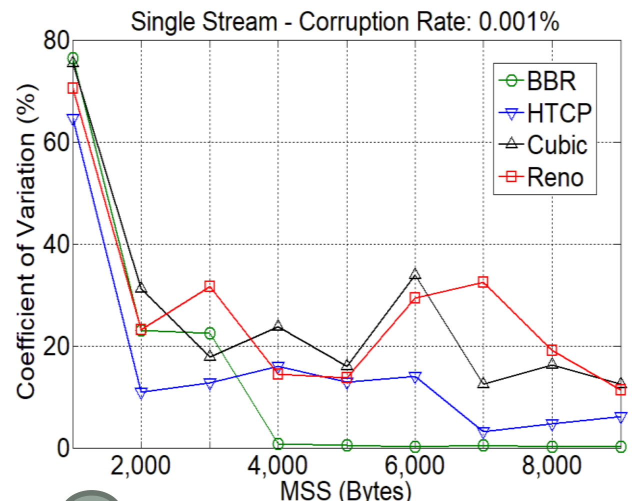
Scenario

- Each experiment lasted 70 seconds (first 10 seconds were not taken into account)
- For each test condition, ten experiments were conducted and the average throughput was computed
- The performance of TCP BBR was compared with that of TCP Reno, TCP HTCP, and TCP Cubic

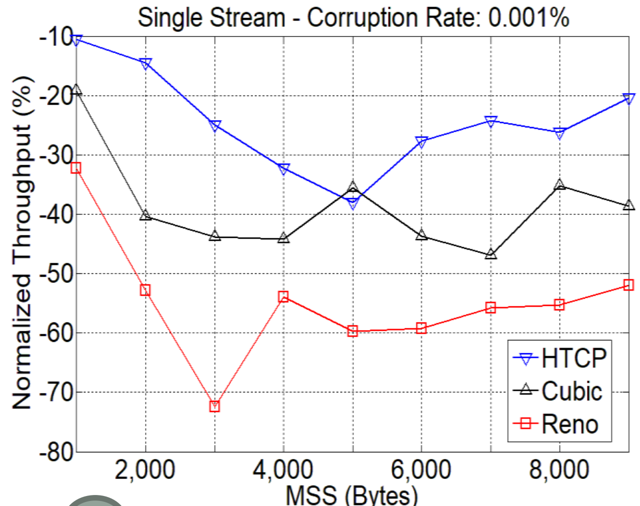
Results – Corruption Rate 0.001%



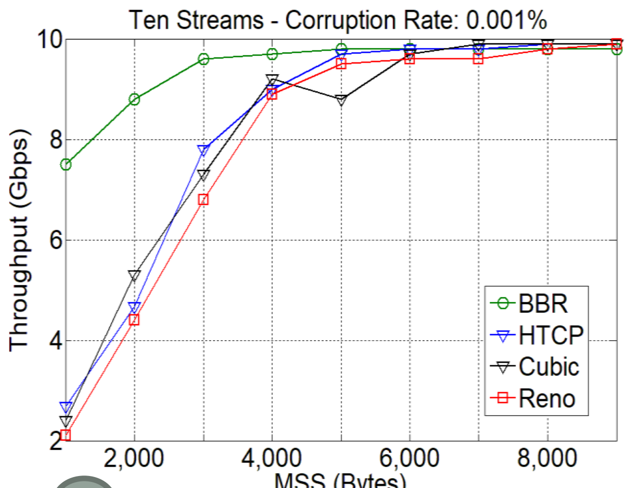
1



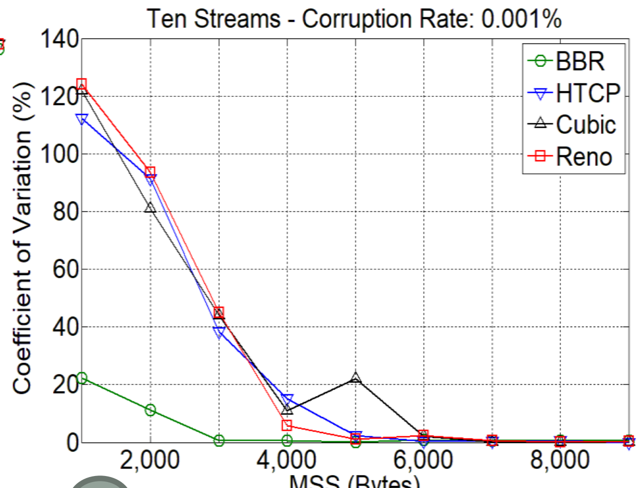
2



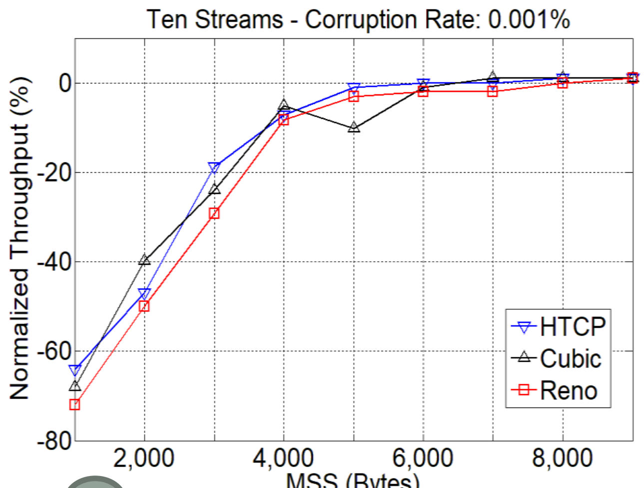
3



4

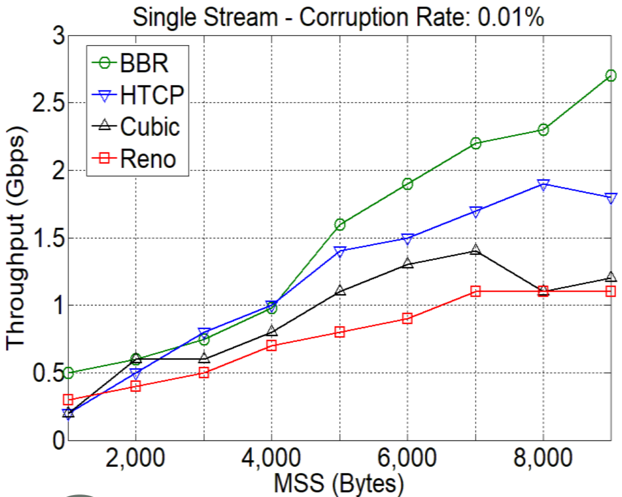


5

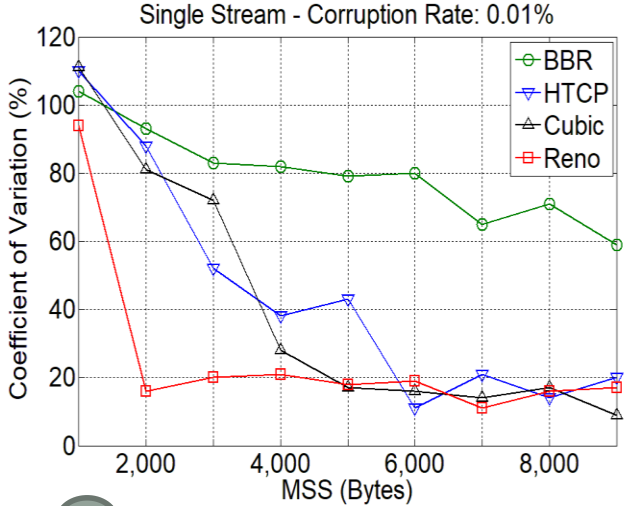


6

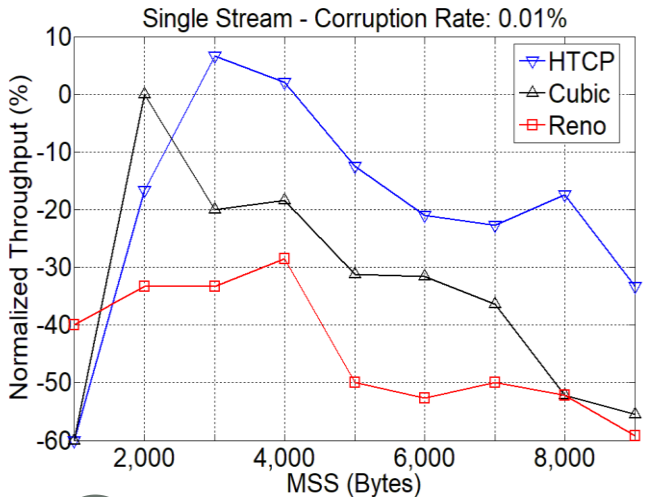
Results – Corruption Rate 0.01%



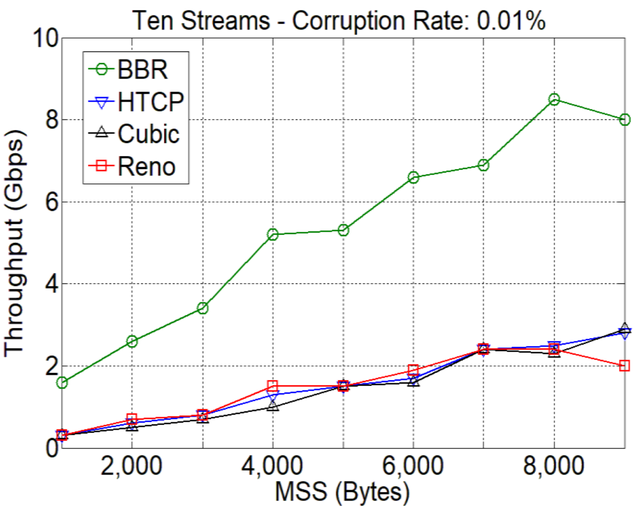
1



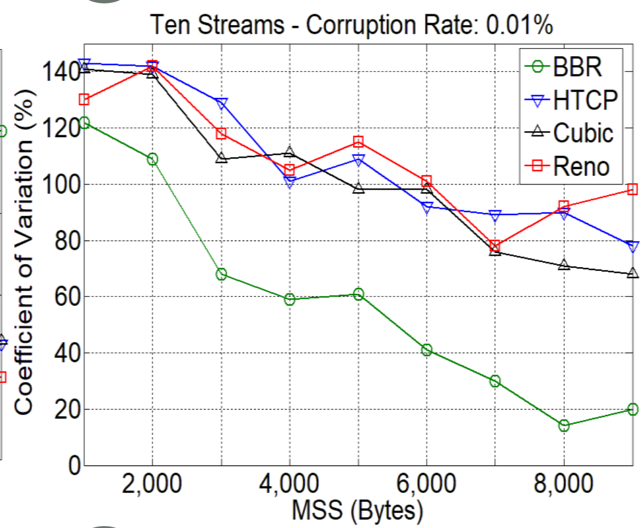
2



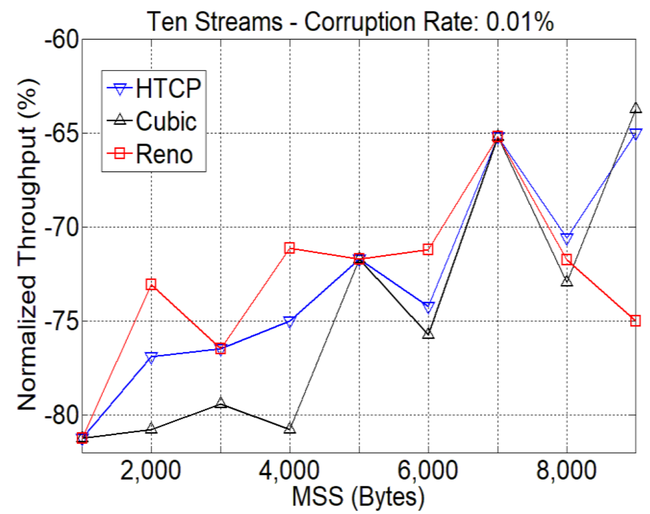
3



4



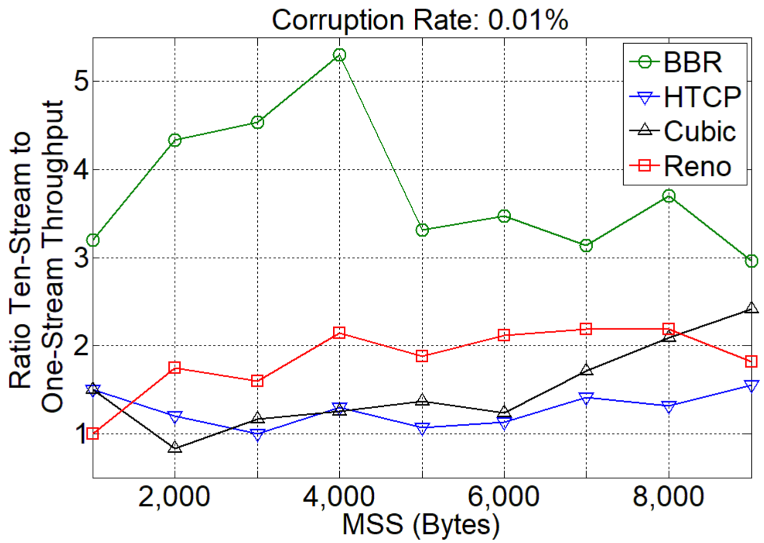
5



6

Results

- Parallel streams improved BBR's throughput by more than a factor of 3
- When parallel streams are used, the performance of HTCP, Cubic, and Reno are similar

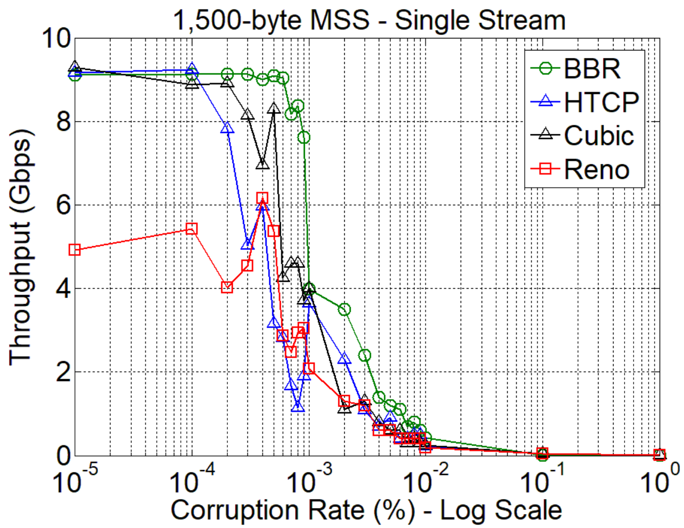


Improvement factor from using parallel streams

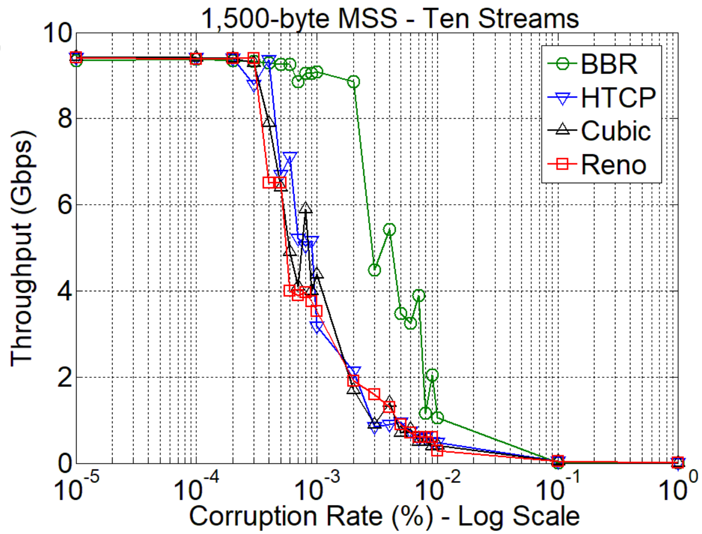
Algorithm	Average	Max	Min
BBR	3.77	5.31	2.96
HTCP	1.27	1.56	1
Cubic	1.48	0.83	2.21
Reno	1.85	2.81	1

Results

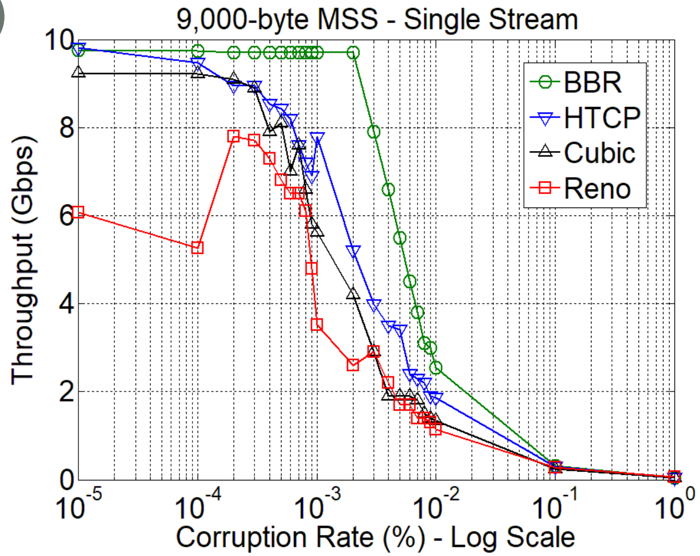
1



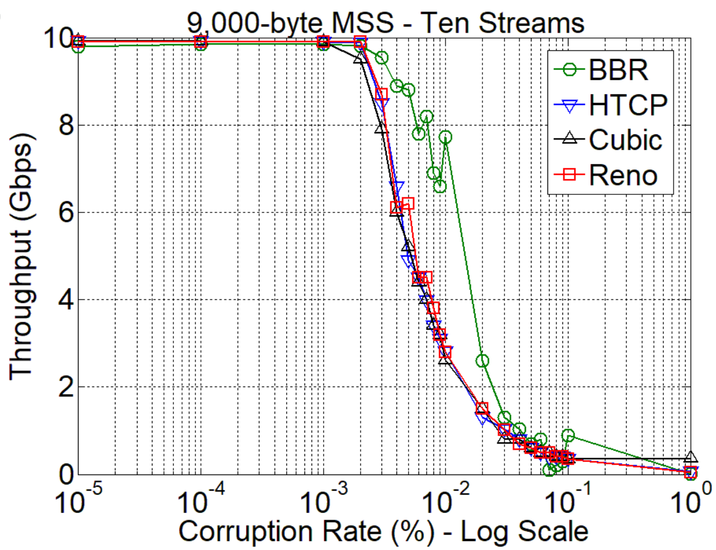
2



3



4



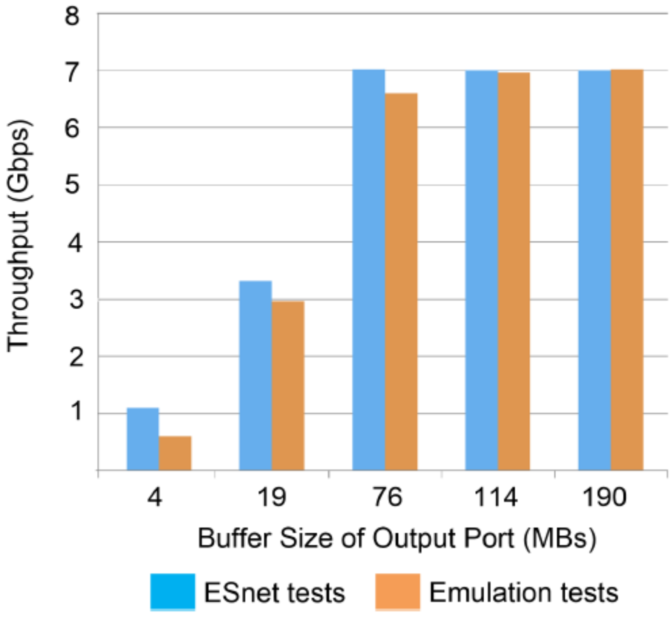
Conclusion

- This work focuses on the suitability of BBR for transporting big flows
- In particular, the two features studied here (overlooked in general-purpose networks) are parallel streams and MSS
- While the use of parallel streams leads to substantial throughput increase in all algorithms, BBR's gains are more pronounced than those of window-based loss-based algorithms
- The use of parallel streams allows BBR to tolerate almost an order of magnitude higher corruption rates than traditional algorithms, before the throughput collapses

Thank You

Emulation vs Real Networks

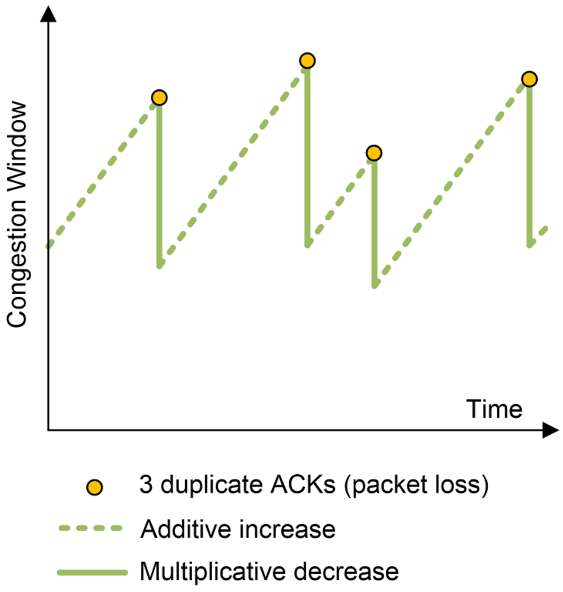
- Throughput of two TCP flows
- RTT: 70 milliseconds; 10 Gbps for all links; bandwidth-delay product: 83.4 MBs



M. Smitasin, B. Tierney, "Evaluating network buffer size requirements," in *2015 Technology Exchange Workshop*, Oct. 2015. [Online]. Available: https://meetings.internet2.edu/media/medialibrary/2015/10/05/20151005-smitasin-buffersize.pdf

MSS

- Large MSS produces a faster recovery after a packet loss



$$\text{TCP throughput} = \frac{c \cdot MSS}{RTT \cdot \sqrt{p}}$$

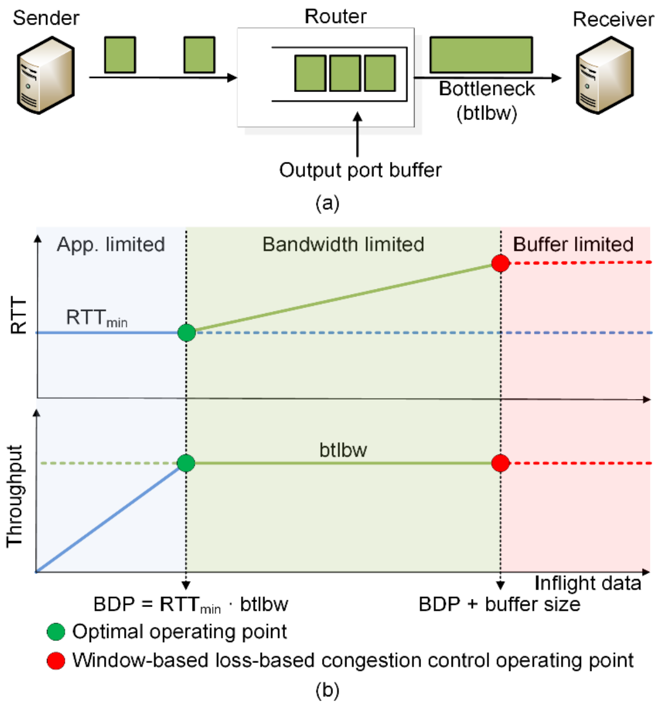
MSS: maximum segment size
RTT: round-trip time
p: loss rate
c: constant

Note: the above equation does not apply to BBR

M. Mathis, J. Semke, J. Mahdavi, T. Ott, "The macroscopic behavior of the tcp congestion avoidance algorithm," *ACM Computer Communication Review*, vol. 27, no 3, pp. 67-82, Jul. 1997.

BBR Brief Overview

- TCP BBR has been recently proposed as a congestion control algorithm[1]
- BBR represents a disruption from the window-based loss-based congestion control used during the last decades[2]
- BBR uses 'pacing' to try to match the bottleneck rate



(a) A viewpoint of a TCP connection. (b) Throughput and RTT, as a function of inflight data[1].

1. N. Cardwell, Y. Cheng, C. Gunn, S. Yeganeh, V. Jacobson, "BBR: congestion-based congestion control," *Communications of the ACM*, vol 60, no. 2, pp. 58-66, Feb. 2017.
2. https://www.thequilt.net/wp-content/uploads/BBR-TCP-Opportunities.pdf