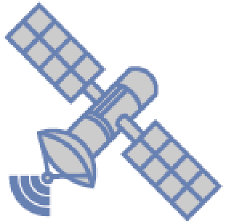
University of Stavanger

On the Suitability of BBR Congestion Control for QUIC over GEO SATCOM Networks



Aitor Martin

Naeem Khademi

Department of Electrical Engineering and Computer Science

University of Stavanger, Norway

1

Introduction

University of Stavanger

1 Introduction

University of Stavanger

- **Geosynchronous Satellite Communication** (GEO SATCOM) networks are becoming popular candidates for providing broadband Internet connectivity in novel 5G/6G use-cases
- In parallel, we are witnessing major breakthrough on the Internet transport layer;
  1) **standardization and deployment of QUIC,** a general-purpose transport protocol that is
     a) fully encrypted
     b) deployed on the user space over UDP
  2) **development of modern congestion control (CC), i.e. BBR**, aiming to optimize bandwidth utilization while minimizing network latency
- To this date, around 8,3% of websites are already using QUIC [1]. Therefore, we expect an **increase of QUIC traffic over SATCOM**

[1] Usage Statistics of QUIC for Websites, July 2022. Retrieved 07/20/2022 from https://w3techs.com/technologies/details/ce-quic.

1 Problem statement

University of Stavanger

CHALLENGES FOR THE TRANSPORT LAYER INTRODUCED BY SATCOM LINKS

mainly caused by

1. long propagation delay
2. propagation errors
3. bandwidth asymmetry

HOW TO MITIGATE THE IMPACT OF THESE CHALLENGES?

TCP traffic is usually optimized with **Performance-Enhancing Proxies (PEPs)**

However, **QUIC**'s full encryption disables PEP optimizations!

Several studies have shown that **TCP-PEP outperforms QUIC greatly** [2], even with QUIC's fast handshake

Raised interest in boosting QUIC performance over SATCOM through protocol mechanisms
We investigate the use of BBR congestion control

[2] Nicolas Kuhn et al., 2020. QUIC: Opportunities and threats in SATCOM. In 2020 10th Advanced Satellite Multimedia Systems Conference and the 16th Signal Processing for Space Communications Workshop (ASMS/SPSC)

1 Our contribution

University of Stavanger

In this work, we investigate the following items:

1. the **general performance** of QUIC with BBRv2 over GEO SATCOM
2. other important aspects of **CC performance** (e.g., intra- and inter-protocol fairness, latecomer fairness, etc.) over long-haul satellite links
3. the impact of **packet loss** and **bandwidth asymmetry**
4. the impact of **QUIC implementation choice**

2

Background

2 Transport Layer over SATCOM

Challenges introduced by SATCOM links

1. Long Round-Trip Time (RTT) ~ 600 ms

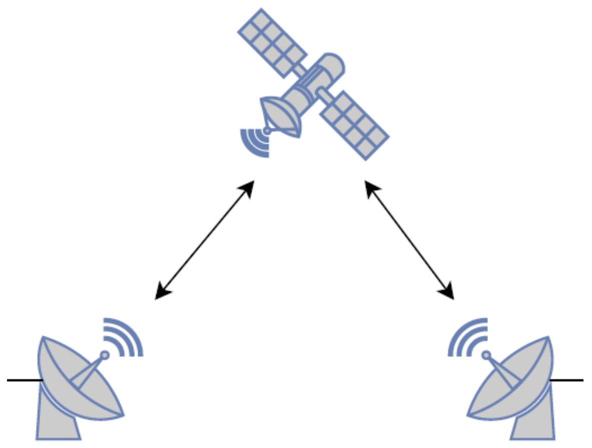
- Long protocol feedback
  - Slower CC convergence
  - Delayed loss detection and recovery
- High Bandwidth-Delay Product (BDP) → Larger buffers needed

2. **Propagation errors** due to e.g. rain fading

- Can be problematic for loss-based CC (e.g. NewReno or CUBIC)

3. Bandwidth asymmetry

- ACK congestion in the return path can limit forward throughput



2 Transport Layer over SATCOM

University of Stavanger

Challenges introduced by SATCOM links

Proposed Solutions

1. **Long Round-Trip Time (RTT) ~ 600 ms**
   - Long protocol feedback
     - Slower CC convergence
     - Delayed loss detection and recovery
   - High Bandwidth-Delay Product (BDP) → Larger buffers needed

   Solutions: Satellite-optimized CC (larger IW, faster slow start); BDP Frame extension [3]

2. **Propagation errors** due to e.g. rain fading
   - Can be problematic for loss-based CC (e.g. NewReno or CUBIC)

   Solutions: FEC for QUIC; Model-based CC (e.g. BBR)

3. **Bandwidth asymmetry**
   - ACK congestion in the return path can limit forward throughput

   Solution: ACK Frequency Extension [4]

[3] Nicolas Kuhn et al. BDP Frame Extension. Internet-Draft draft-kuhn-quic-bdpframe-extension-00, IETF, March 2022
[4] Jana Iyengar and Ian Swett. QUIC Acknowledgement Frequency. Internet-Draft draft-ietf-quic-ack-frequency-01, IETF, October 2021.

2 Transport Layer over SATCOM

University of Stavanger

Challenges introduced by SATCOM links

Proposed Solutions

1. **Long Round-Trip Time (RTT) ~ 600 ms**
   - Long protocol feedback
     - Slower CC convergence
     - Delayed loss detection and recovery
   - High Bandwidth-Delay Product (BDP) → Larger buffers needed

Satellite-optimized CC (larger IW, faster slow start)

BDP Frame extension [3]

2. **Propagation errors** due to e.g. rain fading
   - Can be problematic for loss-based CC (e.g. NewReno or CUBIC)

FEC for QUIC

Model-based CC (e.g. BBR)

3. **Bandwidth asymmetry**
   - ACK congestion in the return path can limit forward throughput

ACK Frequency Extension [4]

[3] Nicolas Kuhn et al. BDP Frame Extension. Internet-Draft draft-kuhn-quic-bdpframe-extension-00, IETF, March 2022
[4] Jana Iyengar and Ian Swett. QUIC Acknowledgement Frequency. Internet-Draft draft-ietf-quic-ack-frequency-01, IETF, October 2021.

2 BBR congestion control

University of Stavanger

- The <u>goal of BBR</u>: find **optimal pacing rate** to maximize link utilization while keeping path RTT as low as possible
  - How? Measuring path **Bottleneck Bandwidth** and **RTT.**
- Previous studies show that BBR tends to beat CUBIC over lossy paths [5]

However, three main issues were found with the first version of BBR [6]

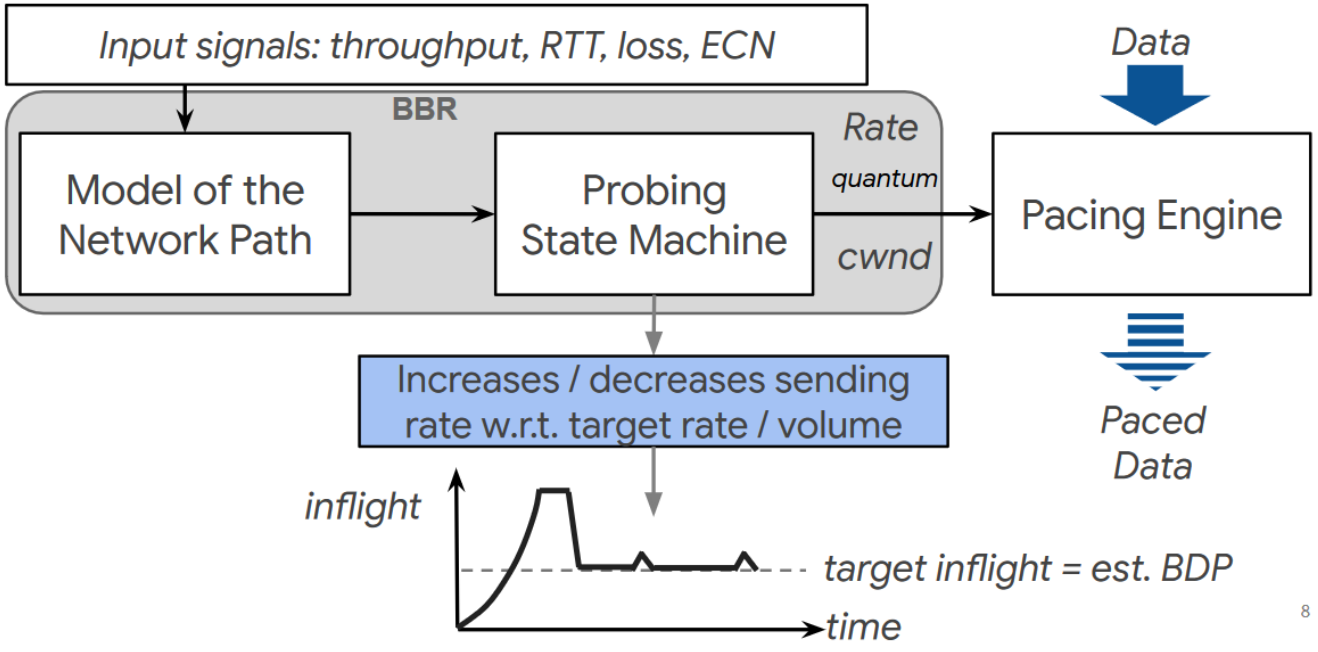
1. Unfairness between parallel BBR flows
2. Aggresiveness against parallel loss-based CC flows
3. RTT unfairness

[5] Cardwell et al. 2016. BBR: Congestion-Based Congestion Control: Measuring Bottleneck Bandwidth and Round-Trip Propagation Time. ACM Queue, 14, 5,
[6] Scholz et al. 2018. Towards a Deeper Understanding of TCP Congestion Control. In 2018 IFIP Networking Conference (IFIP Networking) and Workshops, 1–9.

2 Towards BBRv2

2019 - an update to BBR is proposed, named **BBRv2**, introducing a more complex bandwidth probing mechanism, aiming to solve the previously mentioned issues

- BBRv2 **also reacts to packet loss and ECN**



Source: BBR IETF112 (ICCRG)

2 Related studies

University of Stavanger

Year	Study	Transport	Scenario	Congestion Control
2016	Cardwell et al.	TCP	Terrestrial	BBRv1, CUBIC
2017	Hock et al.			
2018	Scholz et al.			
2019	Jaeger et al.			
2020	Gomez et al.	TCP	Terrestrial	BBRv2, BBRv1, CUBIC
2020	Song et al.			
2021	Song et al.			
2022	Yang et al.			
2021	Claypool et al.	TCP	SATCOM	BBRv1, CUBIC
2022	Zhao et al.			BBRv1, CUBIC, PCC, Hybla
2018	Wang et al.	QUIC	SATCOM	BBRv1, CUBIC
2022	**Our paper**	QUIC	SATCOM	**BBRv2**, BBRv1, CUBIC

2 QUIC implementations

University of Stavanger

A **high amount of QUIC implementations** out there, developed by different agents:

- Web and CDN service providers, e.g. LiteSpeed, Akamai, Cloudflare
- Big technological companies, e.g. Google, Facebook, Apple, Microsoft
- IETF/IRTF contributors, e.g., ngtcp2, picoquic

- *Robin Marx et al. (2020)* [7] , *Sebastian Endres et al. (2022)* [8] have reported high heterogeneity among implementations
- Even though RFC9002 specifies a CC mechanism similar to NewReno for QUIC, **many stacks implement CUBIC and BBR**

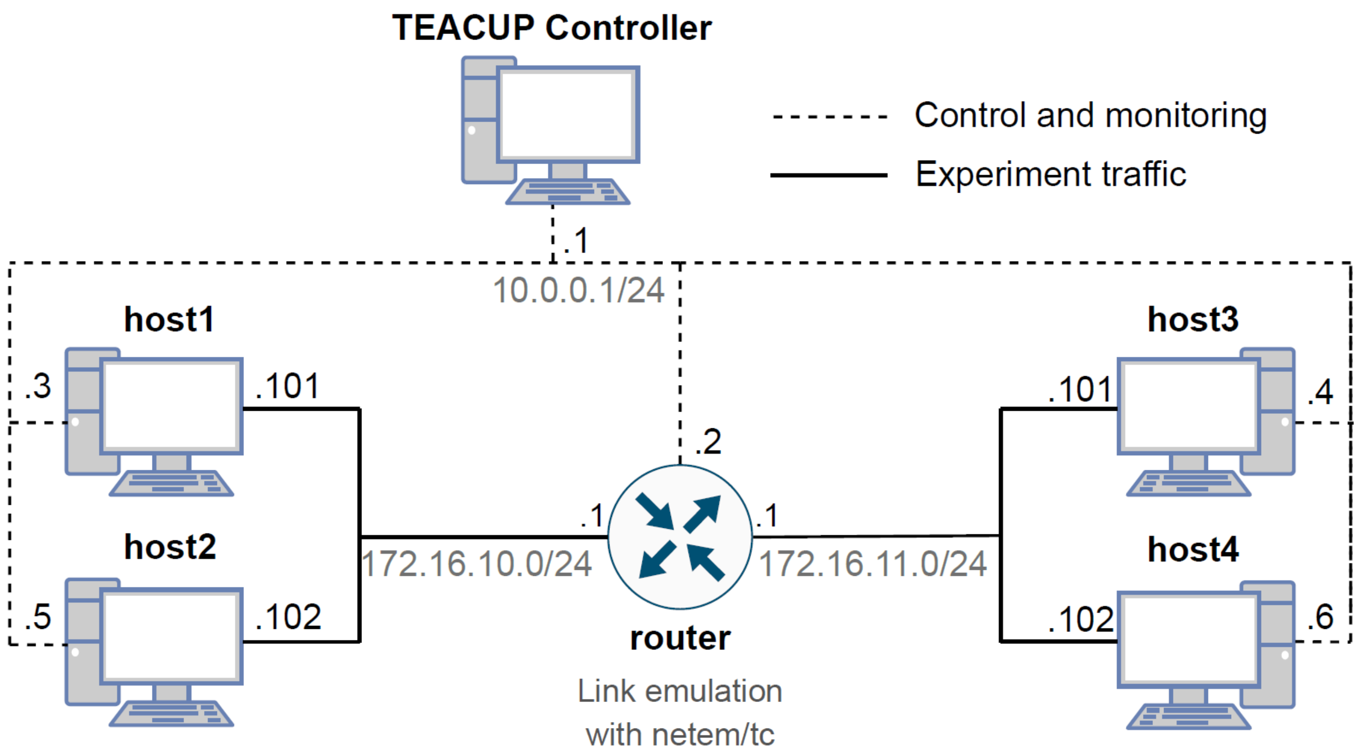
[7] Robin Marx et.al. 2020. Same Standards, Different Decisions: A Study of QUIC and HTTP/3 Implementation Diversity. en. In Proceedings of the Workshop on the Evolution, Performance, and Interoperability of QUIC. ACM, Virtual Event USA, (August 2020)
[8] Sebastian Endres et.al. 2022. Performance of QUIC Implementations Over Geostationary Satellite Links

University of Stavanger

3

Experimental Testbed Setup

3 Experimental Testbed Setup



QUIC clients

QUIC servers

UiS Testbed based on TEACUP

Flexible **experiment design** and **test automation**

For this work, TEACUP was **extended for QUIC and QLOG support**

TEACUP: http://caia.swin.edu.au/tools/teacup/

3 Experimental Testbed Setup

University of Stavanger

- Link emulation with **netem/tc**

	SAT	TERR
One Way Delay (OWD)	300 ms	50 ms
Downlink Bandwidth	20 Mbps	20 Mbps
Uplink Bandwidth	20/2 Mbps	20 Mbps
Bottleneck Buffer Size	0.25|0.5|1.0|2.0 x BDP	
Packet Loss Ratio (PLR)	0%, 0.1%, 1%	

- Two QUIC implementations:
    - **ngtcp2** – allows to experiment with BBRv2
    - **picoquic** – reported good performance over SATCOM [8]

[8] Sebastian Endres et.al. 2022. Performance of QUIC Implementations Over Geostationary Satellite Links

4

Results

4

Results

University of Stavanger

We set up four different experimental scenarios:

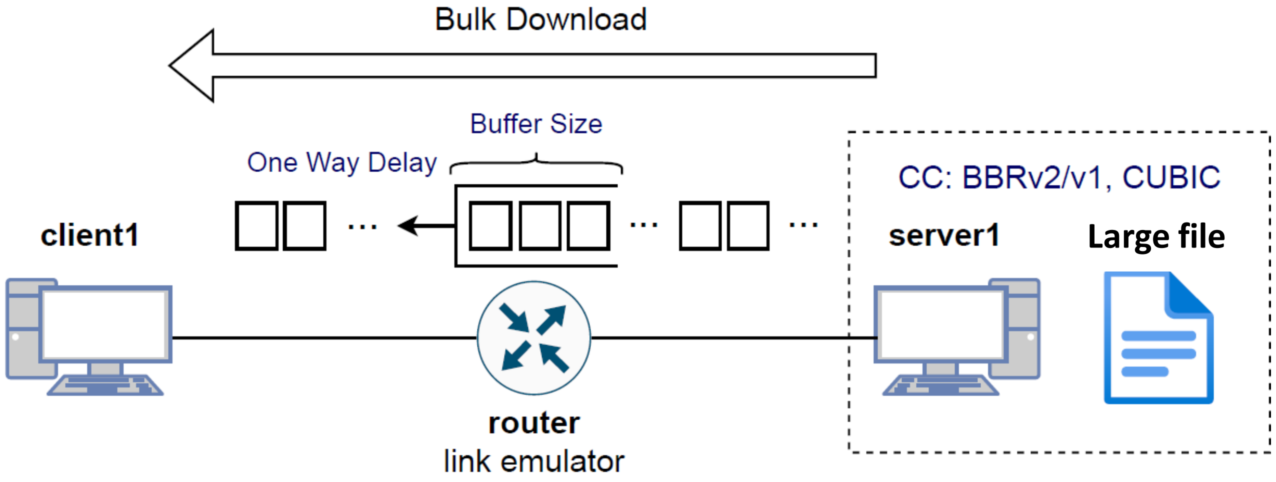
1. **Single-Flow Bulk Download**
2. **Multi-Flow Fairness**
3. **Latecomer Issue**
4. **Mice versus Elephant flows**

All the experiments are run <u>10 times</u>

University of Stavanger

We set up four different experimental scenarios:

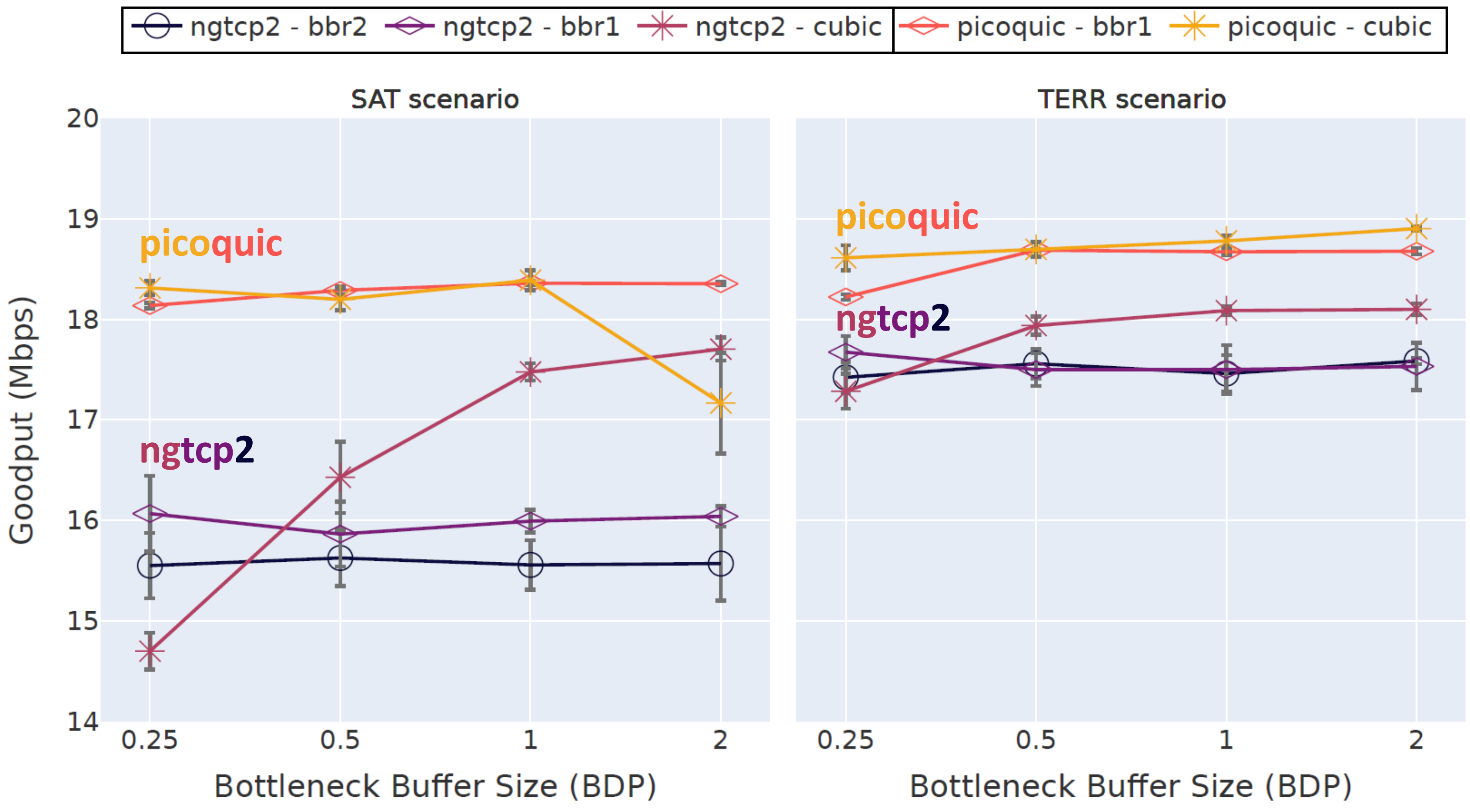
1. **Single-Flow Bulk Download**
2. Multi-Flow Fairness
3. Latecomer Issue
4. Mice versus Elephant flows



Run duration: 120 seconds
Implementations: ngtcp2 and picoquic

4 Results: Bulk Download

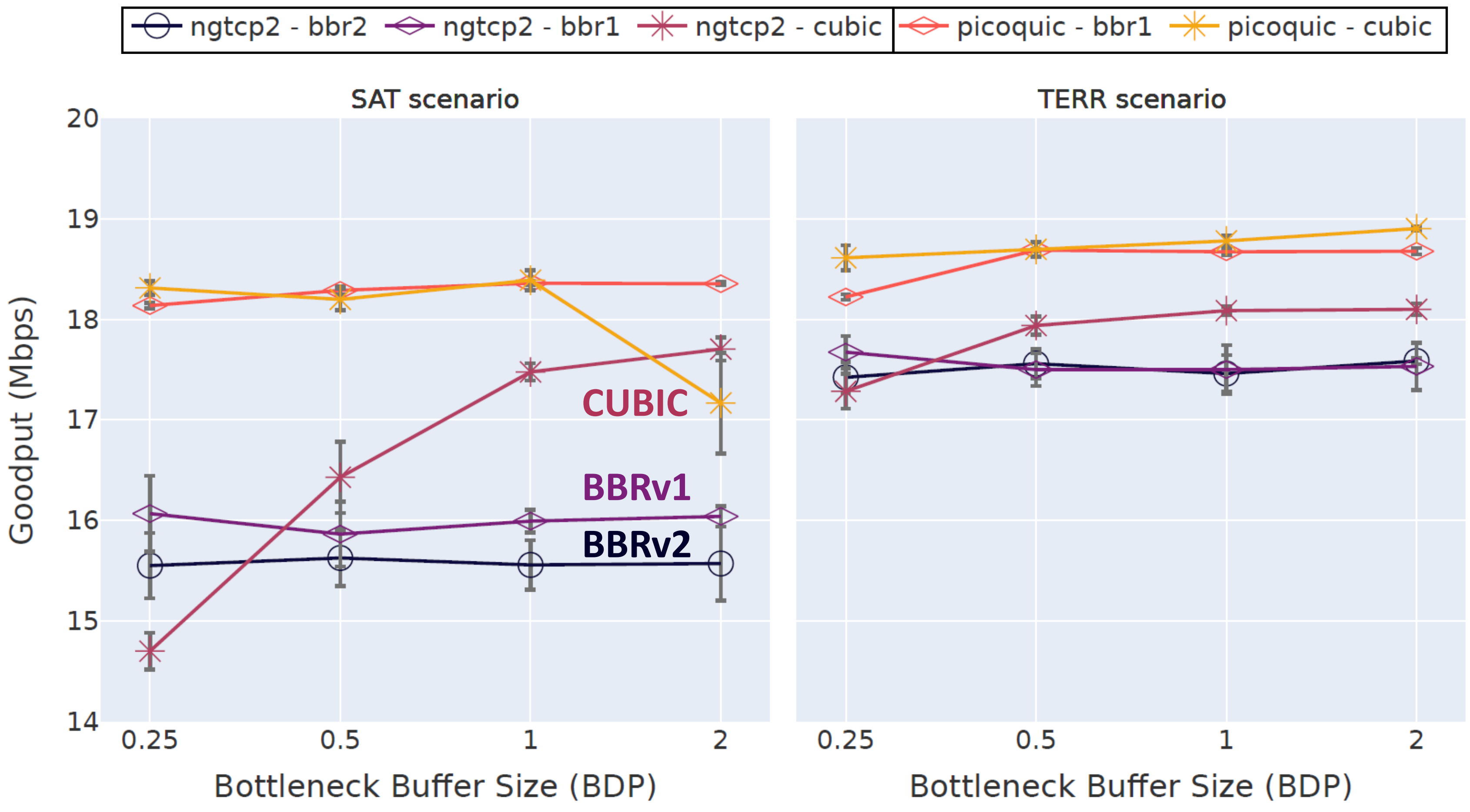
Symmetric link (20/20), no packet loss



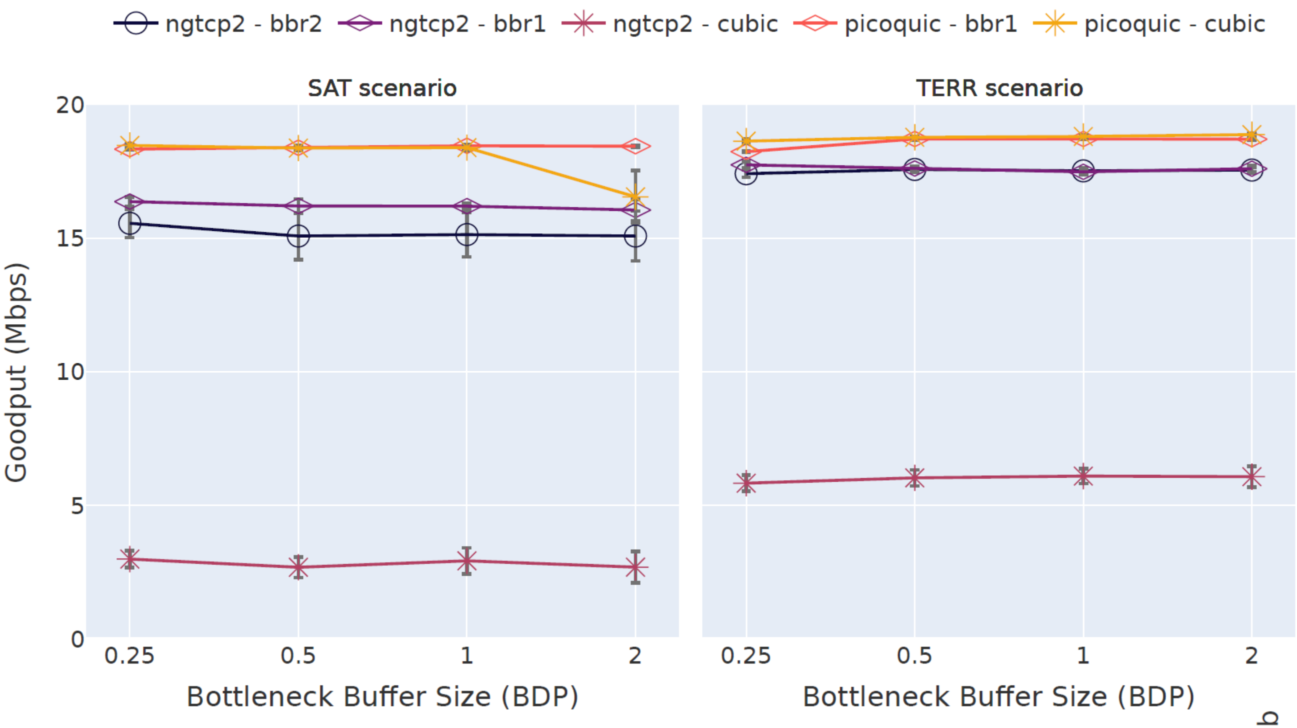
4 Results: Bulk Download

University of Stavanger

Symmetric link (20/20), no packet loss

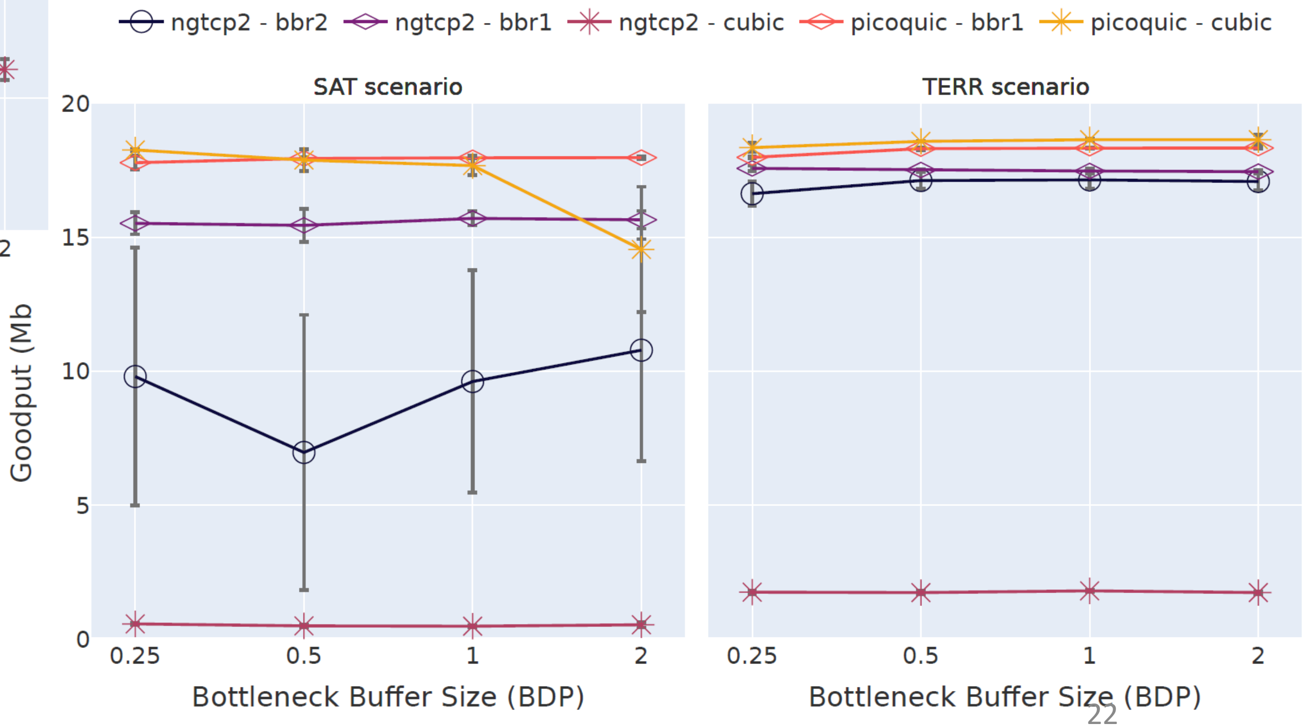


4 Results: Bulk Download with packet loss



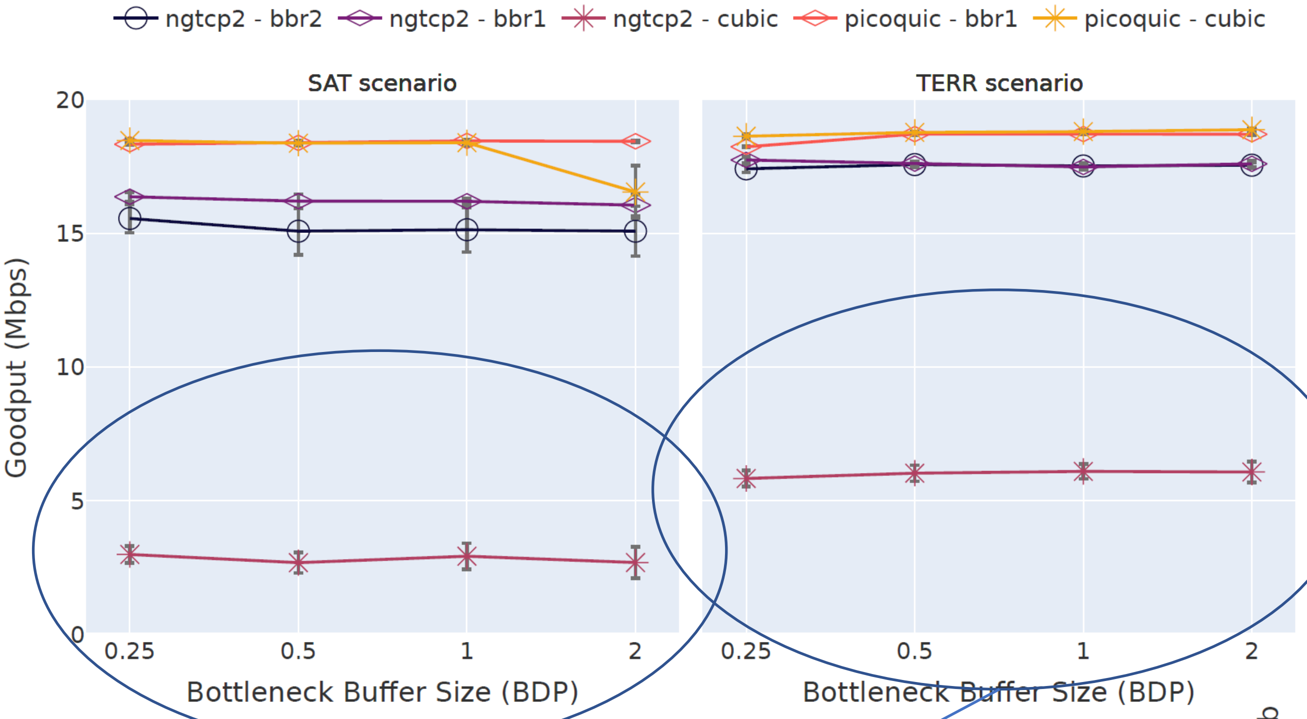
Symmetric link (20/20), PLR = 0,1%

Symmetric link (20/20), PLR = 1%



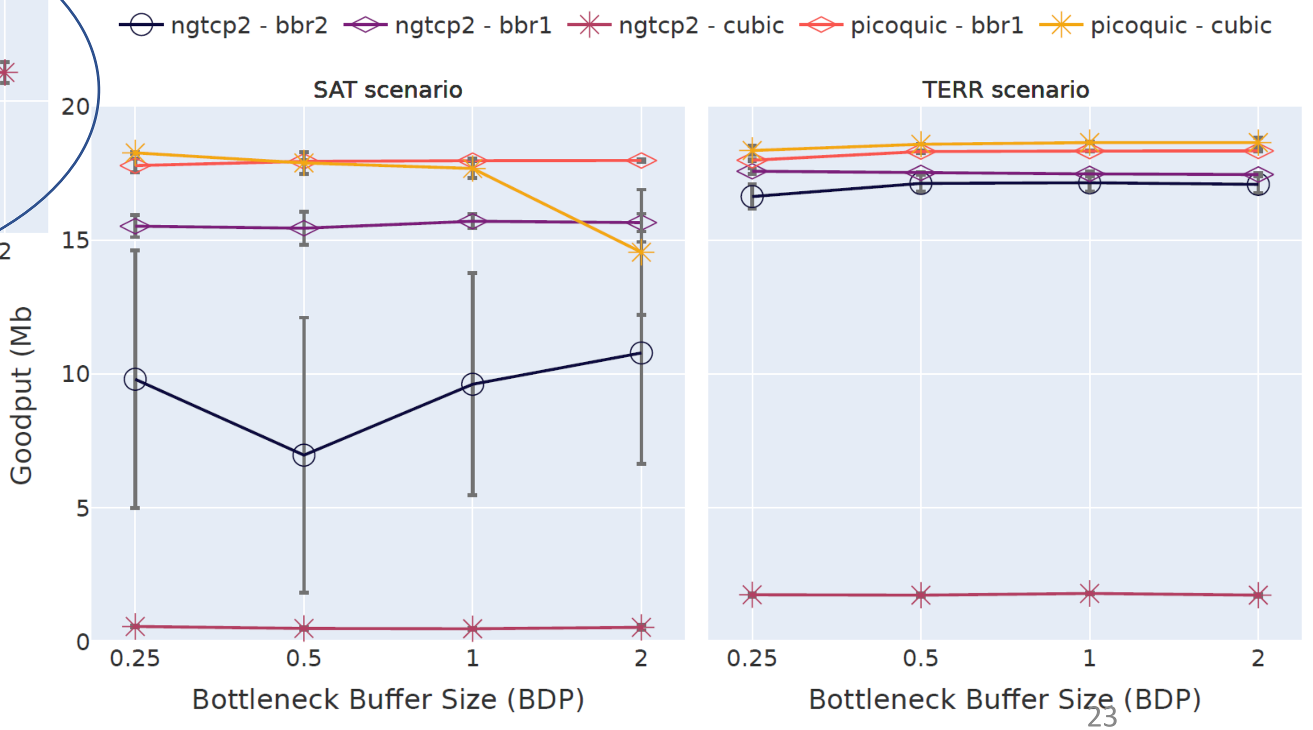
4 Results: Bulk Download with packet loss

University of Stavanger



Symmetric link (20/20), PLR = 0,1%

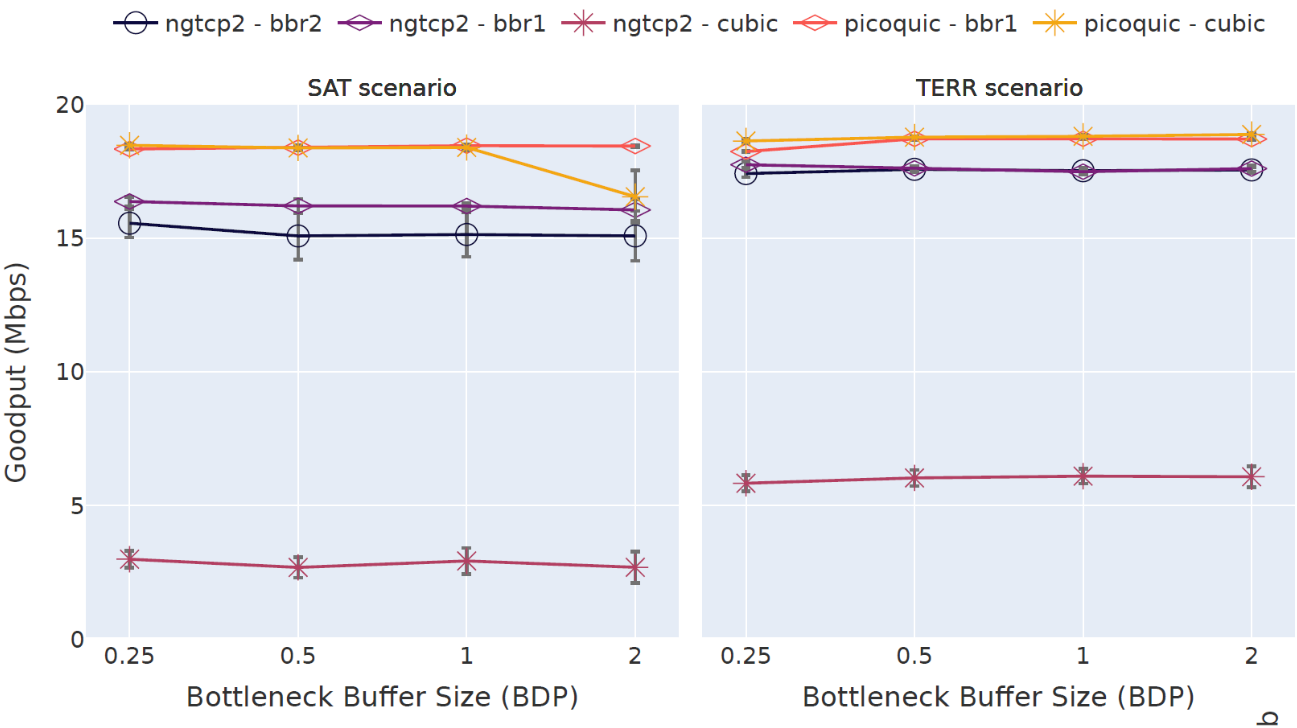
Symmetric link (20/20), PLR = 1%



ngtcp2 – cubic performance drops dramatically!

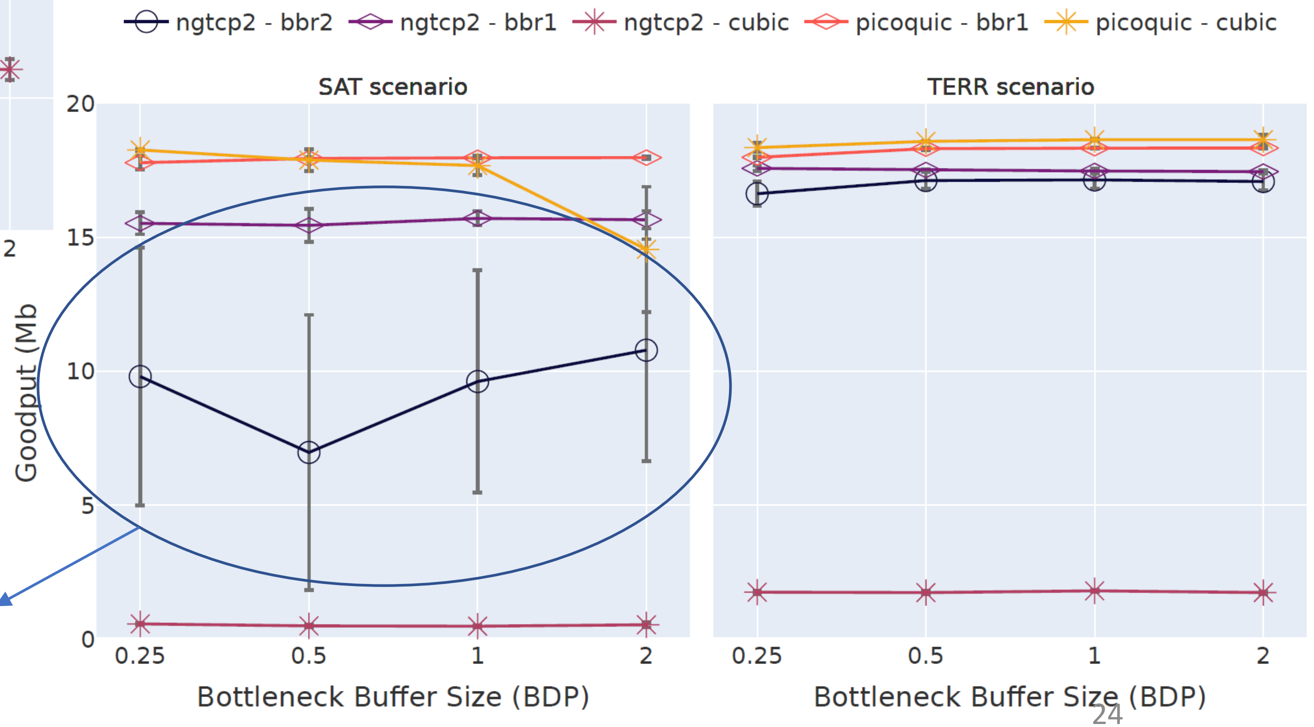
4 Results: Bulk Download with packet loss

University of Stavanger



Symmetric link (20/20), PLR = 0,1%

Symmetric link (20/20), PLR = 1%



Long RTT + high packet loss reduce **BBRv2** performance!

4 Results: Bulk Download with uplink traffic

University of Stavanger

Now, we introduce **cross-traffic in the uplink**, and measure forward goodput over symmetric and asymmetric bandwidth SATCOM setups

Average Forward Goodput (Mbps)

ngtcp2	**Symmetric 20/20**			**Asymmetric 20/2**		
Down | Up	BBRv2	BBRv1	CUBIC	BBRv2	BBRv1	CUBIC
BBRv2	14.73	14.94	13.56	5.31	8.74	6.36
BBRv1	15.10	14.97	14.66	8.03	8.25	5.25
CUBIC	14.90	13.64	11.70	7.38	8.64	7.74
picoquic	**Symmetric 20/20**			**Asymmetric 20/2**		
Down | Up	BBRv2	BBRv1	CUBIC	BBRv2	BBRv1	CUBIC
BBRv2	-	-	-	-	-	-
BBRv1	-	18.36	18.34	-	18.35	18.35
CUBIC	-	18.38	18.25	-	18.34	18.35

4 Results: Bulk Download with uplink traffic

University of Stavanger

Now, we introduce **cross-traffic in the uplink**, and measure forward goodput over symmetric and asymmetric bandwidth setups.

Average Forward Goodput (Mbps)

ngtcp2	Symmetric 20/20			Asymmetric 20/2		
Down | Up	BBRv2	BBRv1	CUBIC	BBRv2	BBRv1	CUBIC
BBRv2	14.73	14.94	13.56	5.31	8.74	6.36
BBRv1	15.10	14.97	14.66	8.03	8.25	5.25
CUBIC	14.90	13.64	11.70	7.38	8.64	7.74
picoquic	Symmetric 20/20			Asymmetric 20/2		
Down | Up	BBRv2	BBRv1	CUBIC	BBRv2	BBRv1	CUBIC
BBRv2	-	-	-	-	-	-
BBRv1	-	18.36	18.34	-	18.35	18.35
CUBIC	-	18.38	18.25	-	18.34	18.35

ngtcp2 performance drops on asymmetric links

4 Results: Bulk Download with uplink traffic

University of Stavanger

Now, we introduce **cross-traffic in the uplink**, and measure forward goodput over symmetric and asymmetric bandwidth setups.

Average Forward Goodput (Mbps)

ngtcp2	Symmetric 20/20			Asymmetric 20/2		
Down | Up	BBRv2	BBRv1	CUBIC	BBRv2	BBRv1	CUBIC
BBRv2	14.73	14.94	13.56	5.31	8.74	6.36
BBRv1	15.10	14.97	14.66	8.03	8.25	5.25
CUBIC	14.90	13.64	11.70	7.38	8.64	7.74

picoquic	**Symmetric 20/20**			**Asymmetric 20/2**		
Down | Up	BBRv2	BBRv1	CUBIC	BBRv2	BBRv1	CUBIC
BBRv2	-	-	-	-	-	-
BBRv1	-	18.36	18.34	-	18.35	18.35
CUBIC	-	18.38	18.25	-	18.34	18.35

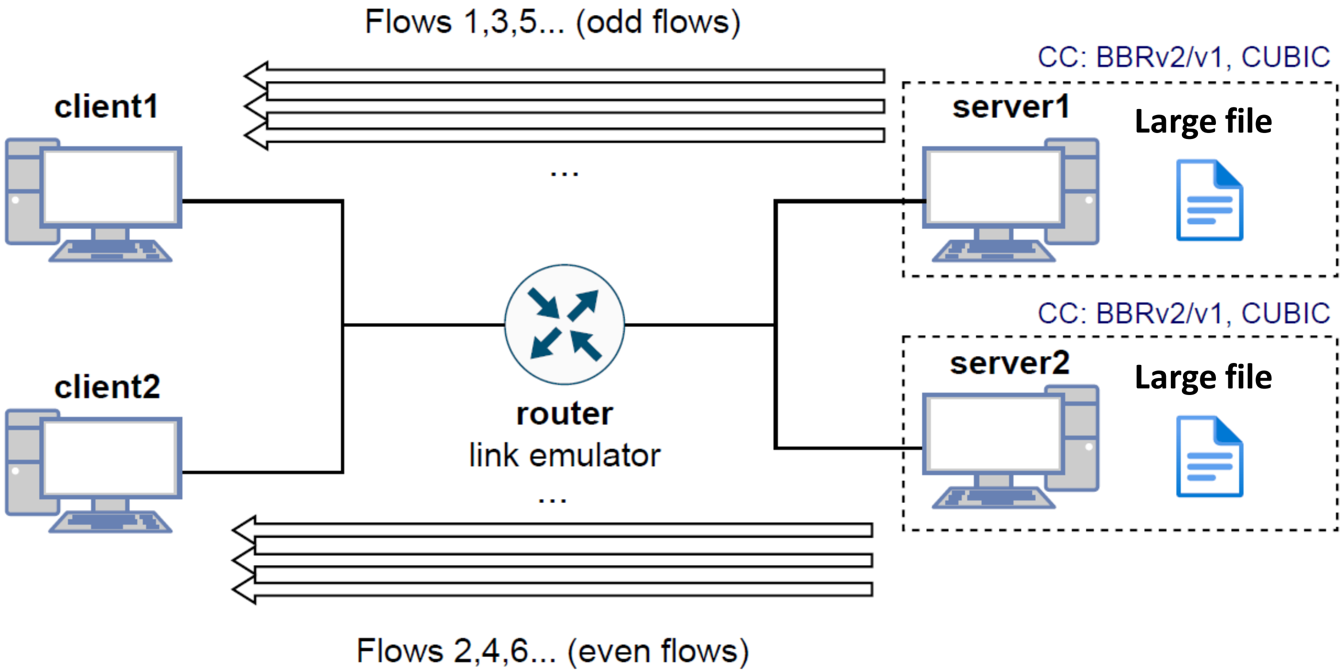
picoquic remains stable on asymmetric links

4 Results

We set up four different experimental scenarios:

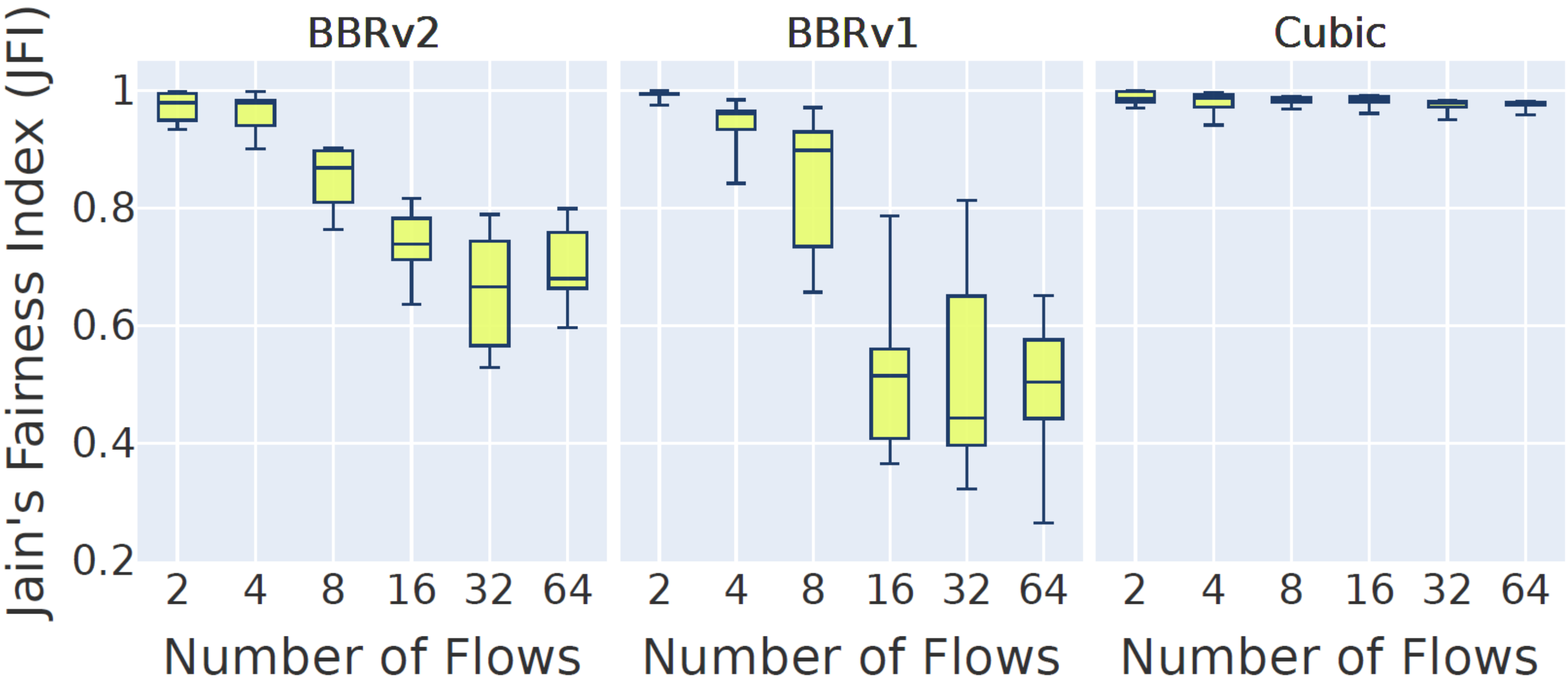
1. Single-Flow Bulk Download
2. **Multi-Flow Fairness**
3. Latecomer Issue
4. Mice versus Elephant flows

We use **Jain's Fairness Index (JFI)** to measure fairness between parallel flows

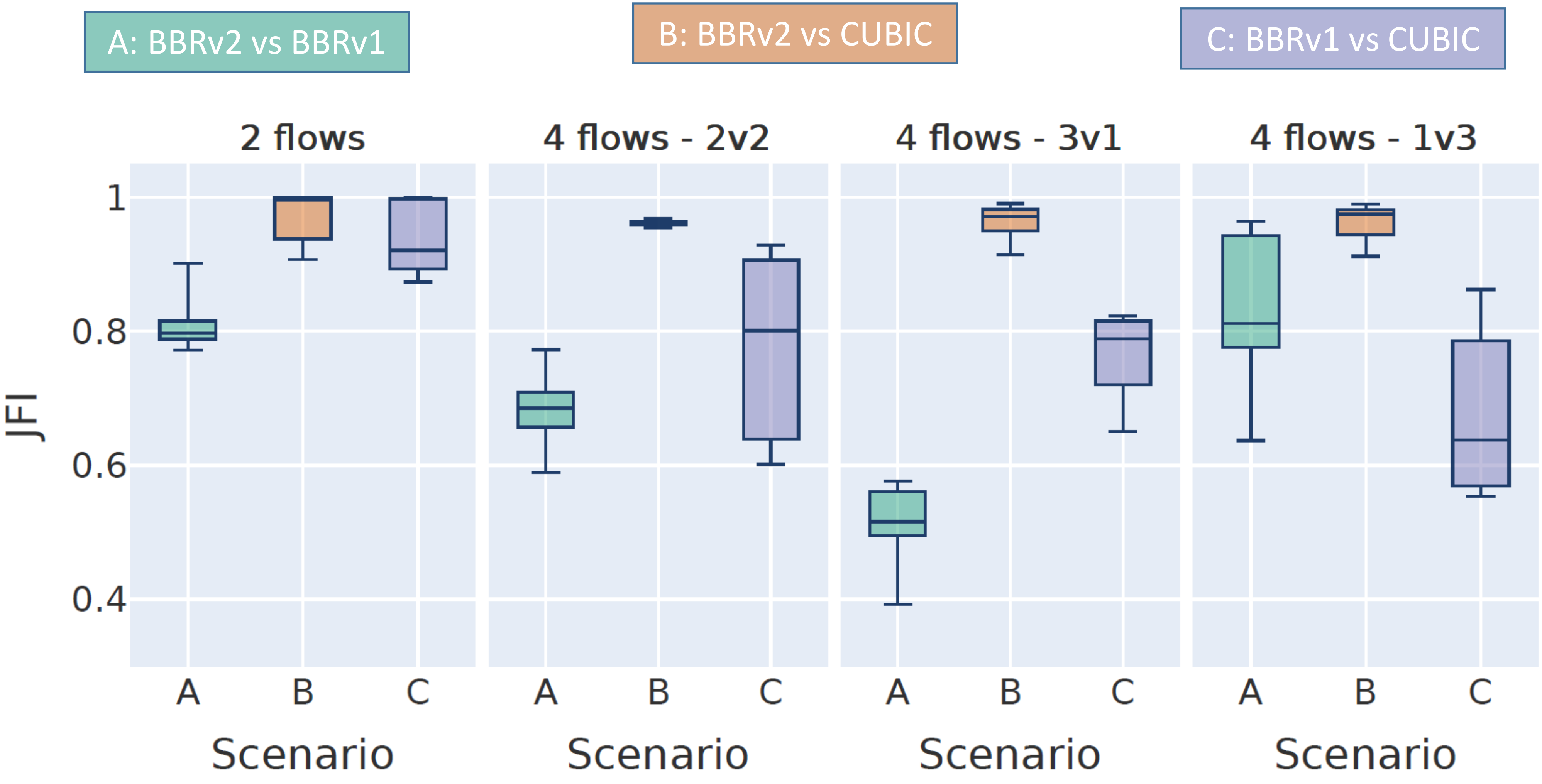


Run duration: 300 seconds
Implementation: ngtcp2

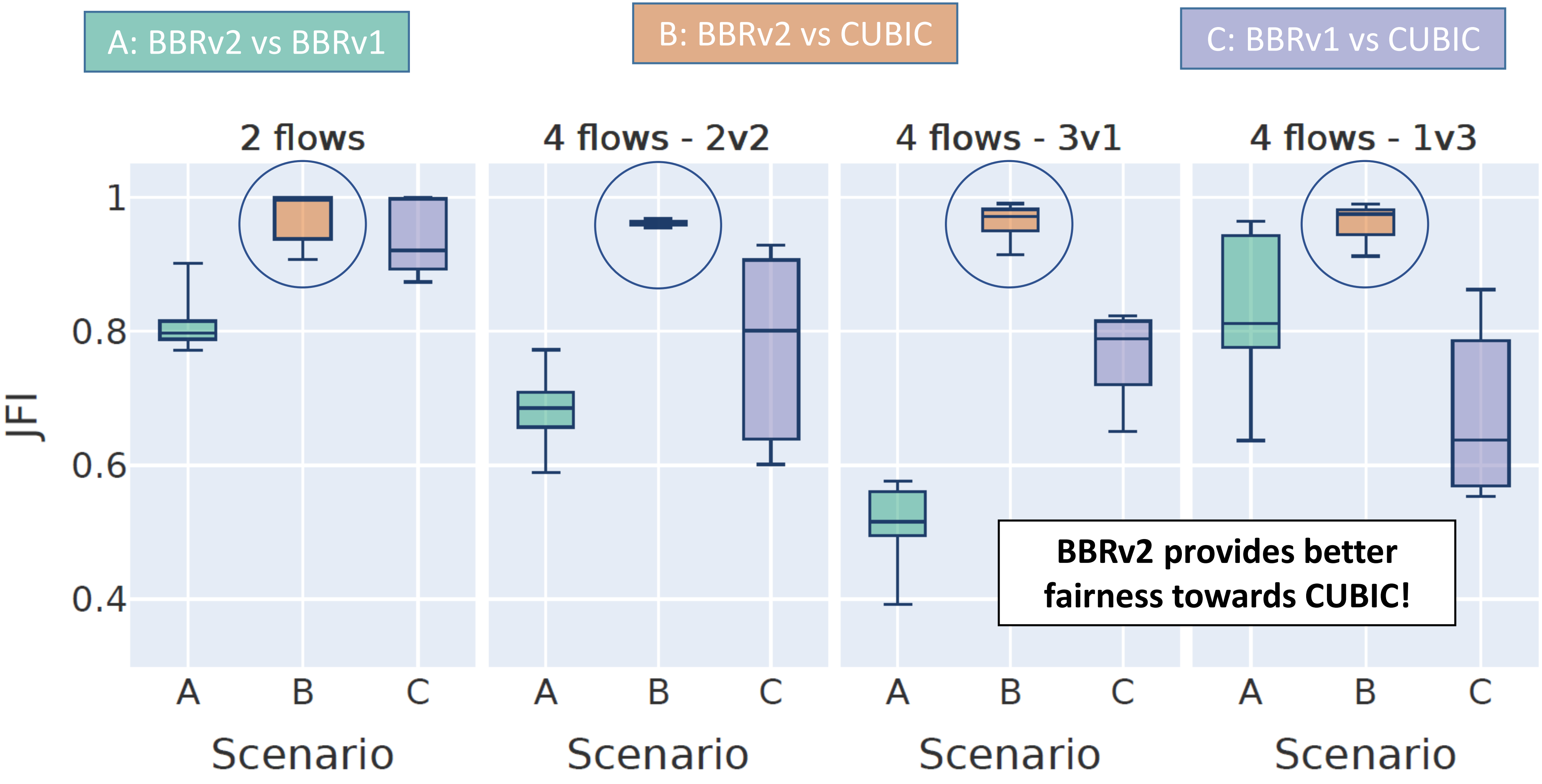
4 Results: Intra-protocol fairness



4 Results: Inter-protocol fairness



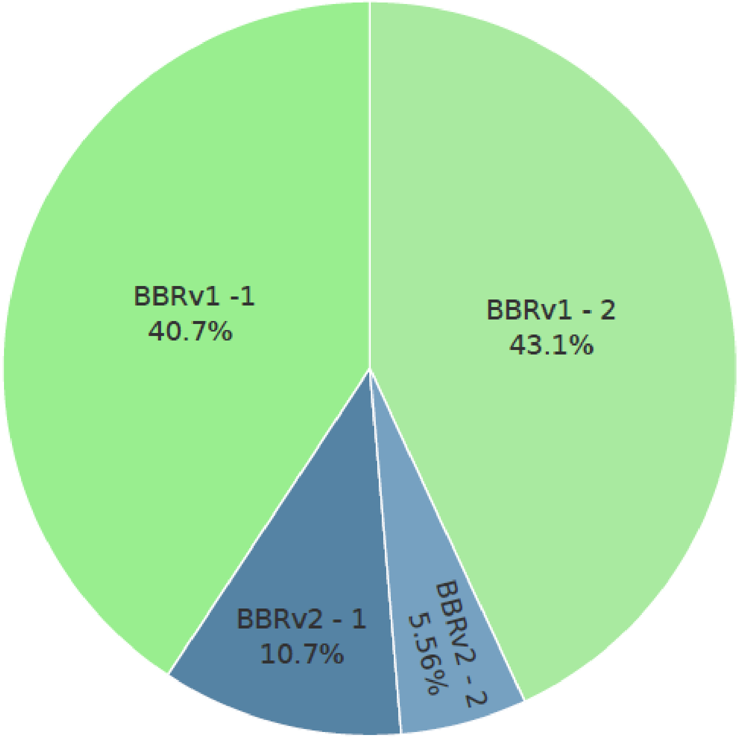
4 Results: Inter-protocol fairness



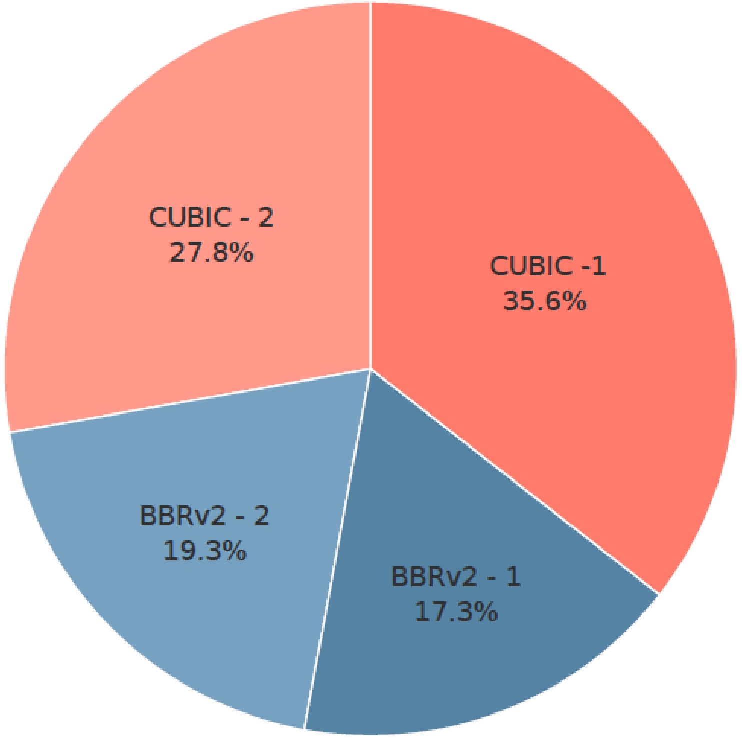
4 Results: Inter-protocol fairness

Average goodput ratio achieved by each flow

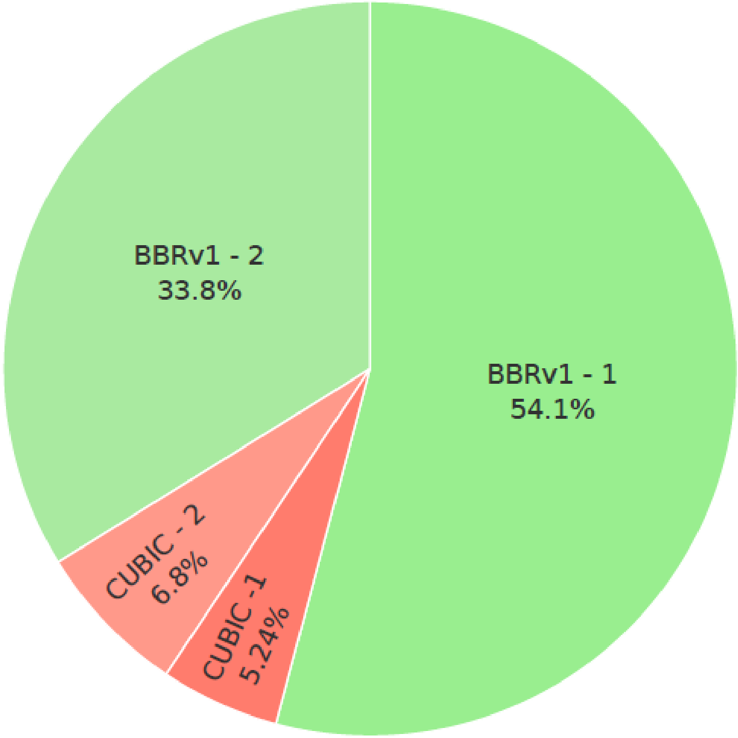
BBRv2 vs BBRv1



BBRv2 vs CUBIC



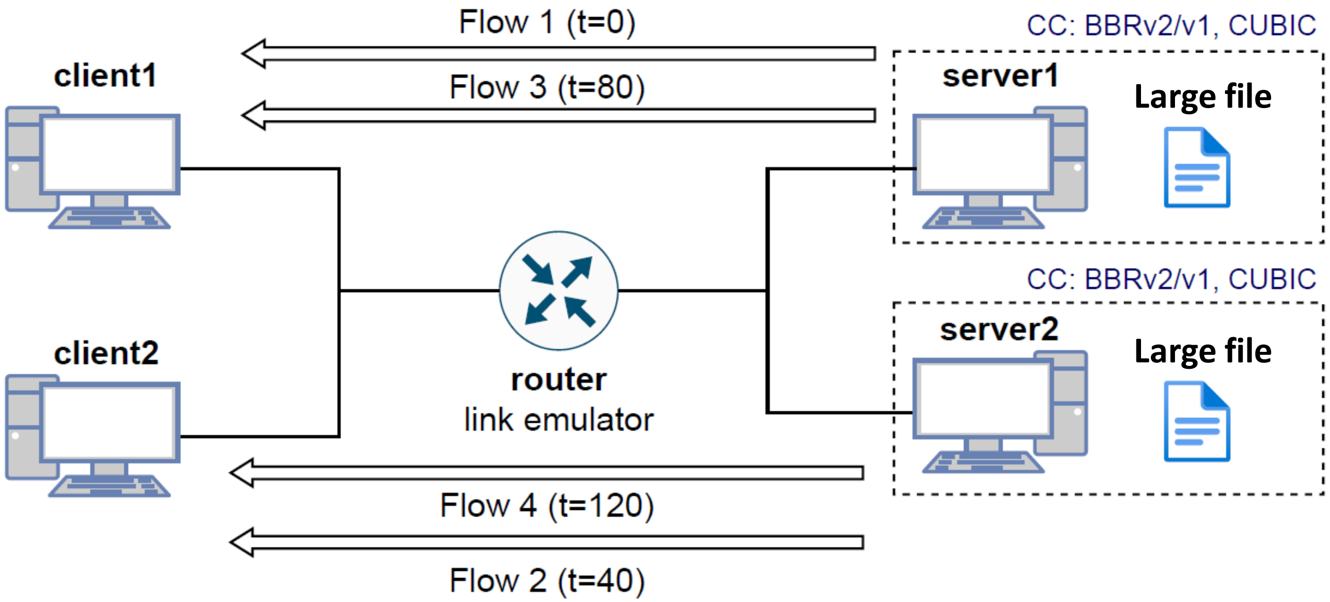
BBRv1 vs CUBIC



Results

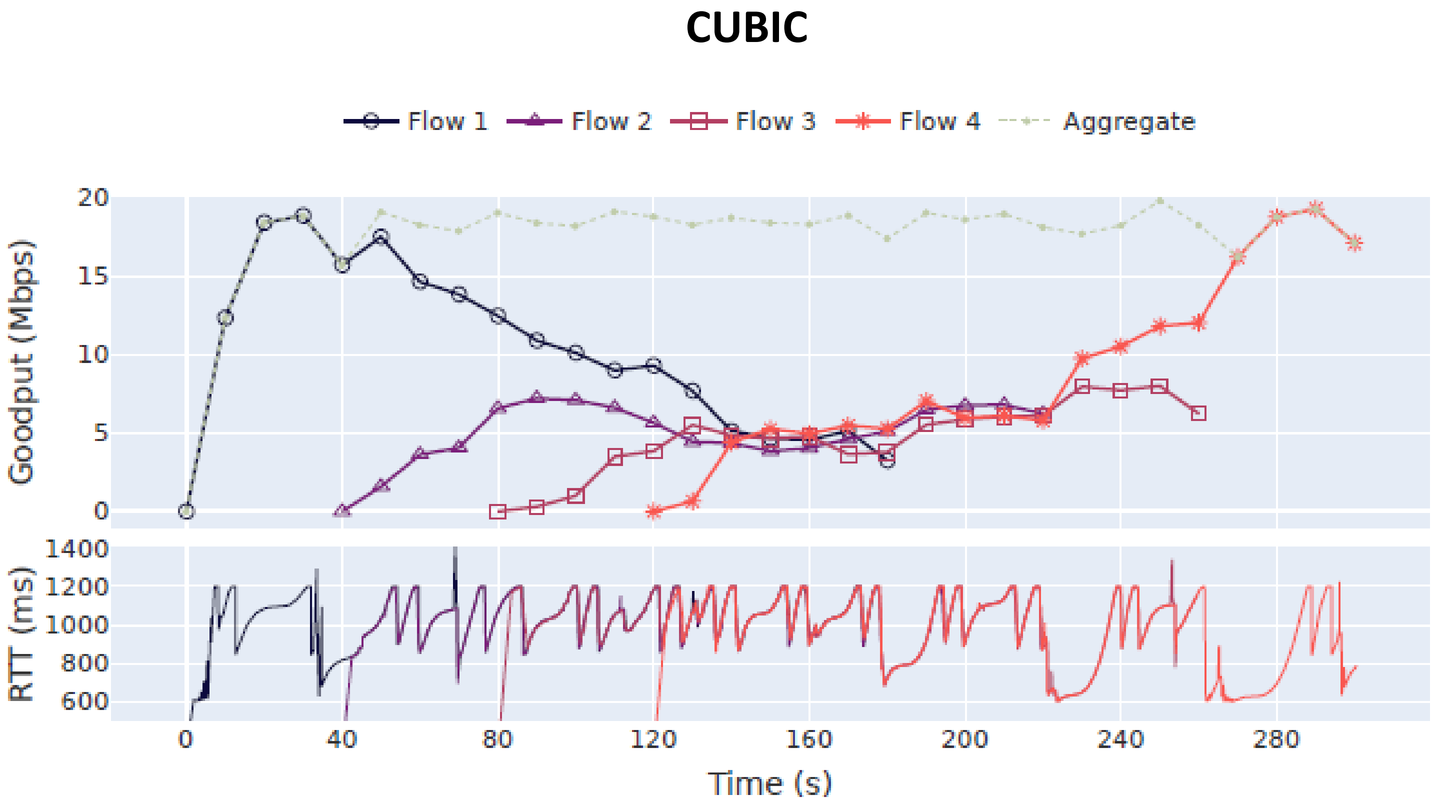
We set up four different experimental scenarios:

1. Single-Flow Bulk Download
2. Multi-Flow Fairness
3. **Latecomer Issue**
4. Mice versus Elephant flows



Run duration: 300 seconds
Implementation: ngtcp2

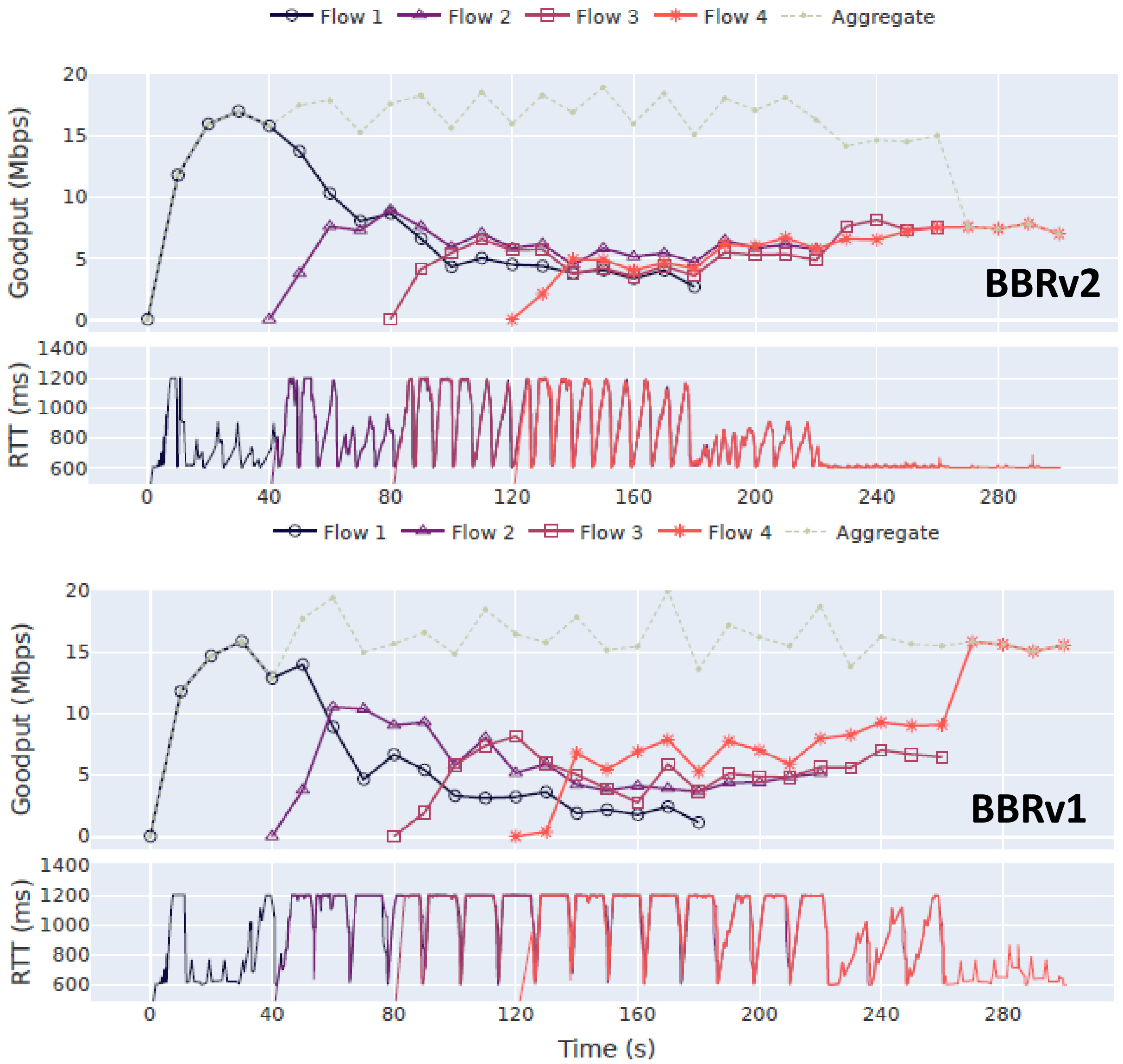
4 Results: Latecomer fairness with CUBIC



CUBIC latecomers **converge very slowly on long RTT paths**

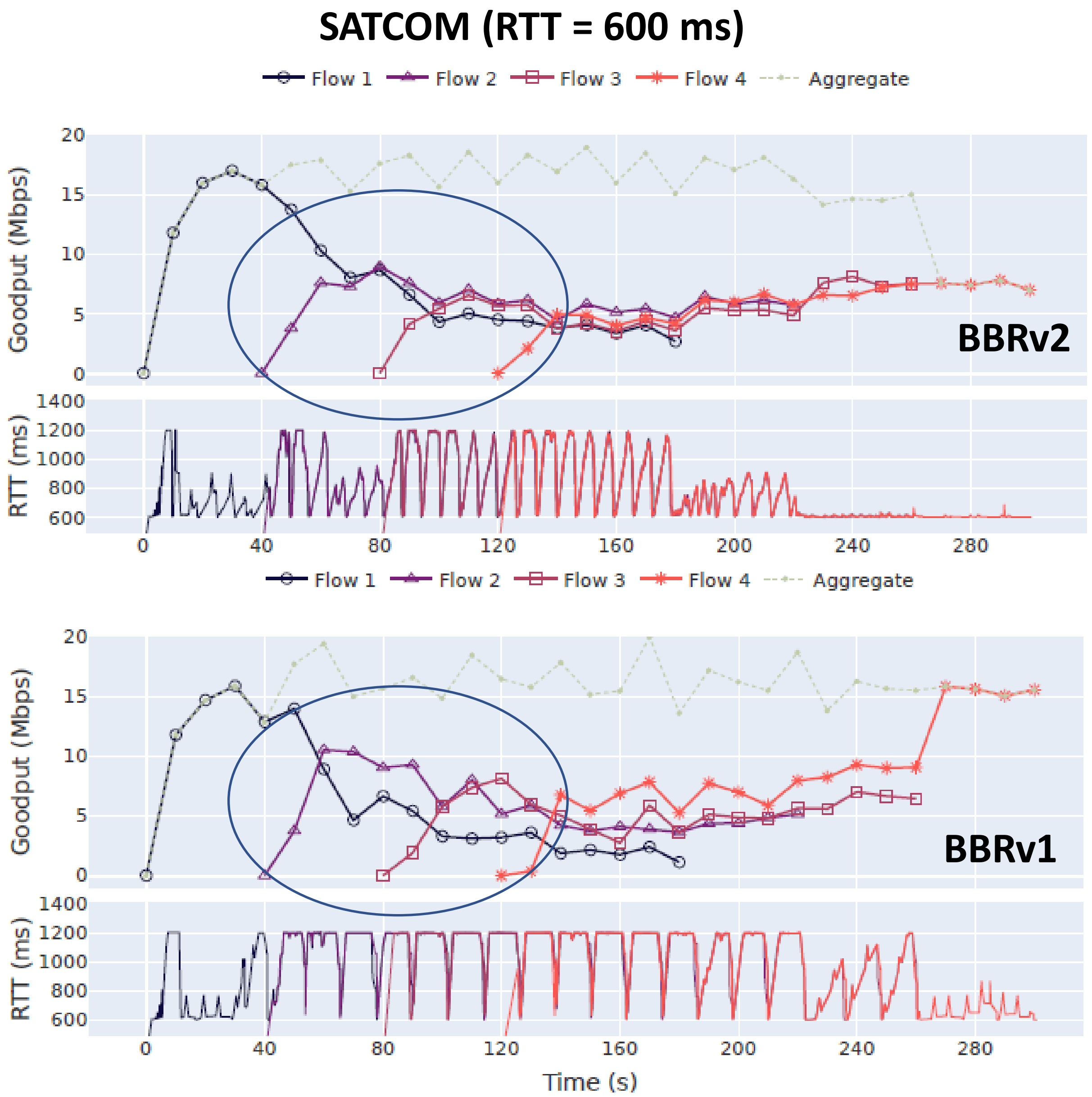
4 Results: Latecomer fairness with BBR

SATCOM (RTT = 600 ms)



BBR latecomers converge faster!

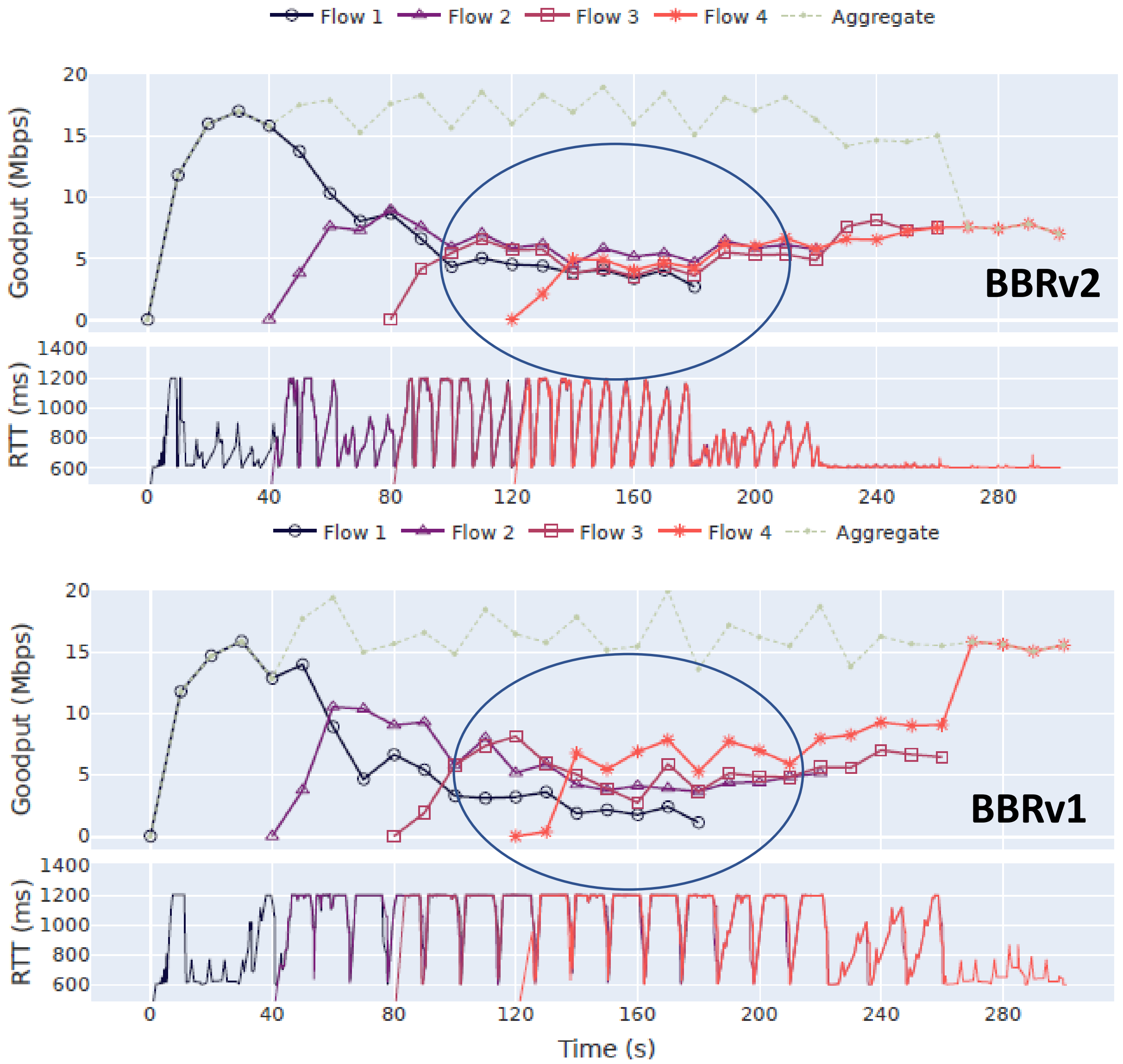
Results: Latecomer fairness with BBR



BBRv2 latecomers join the link less aggressively

4 Results: Latecomer fairness with BBR

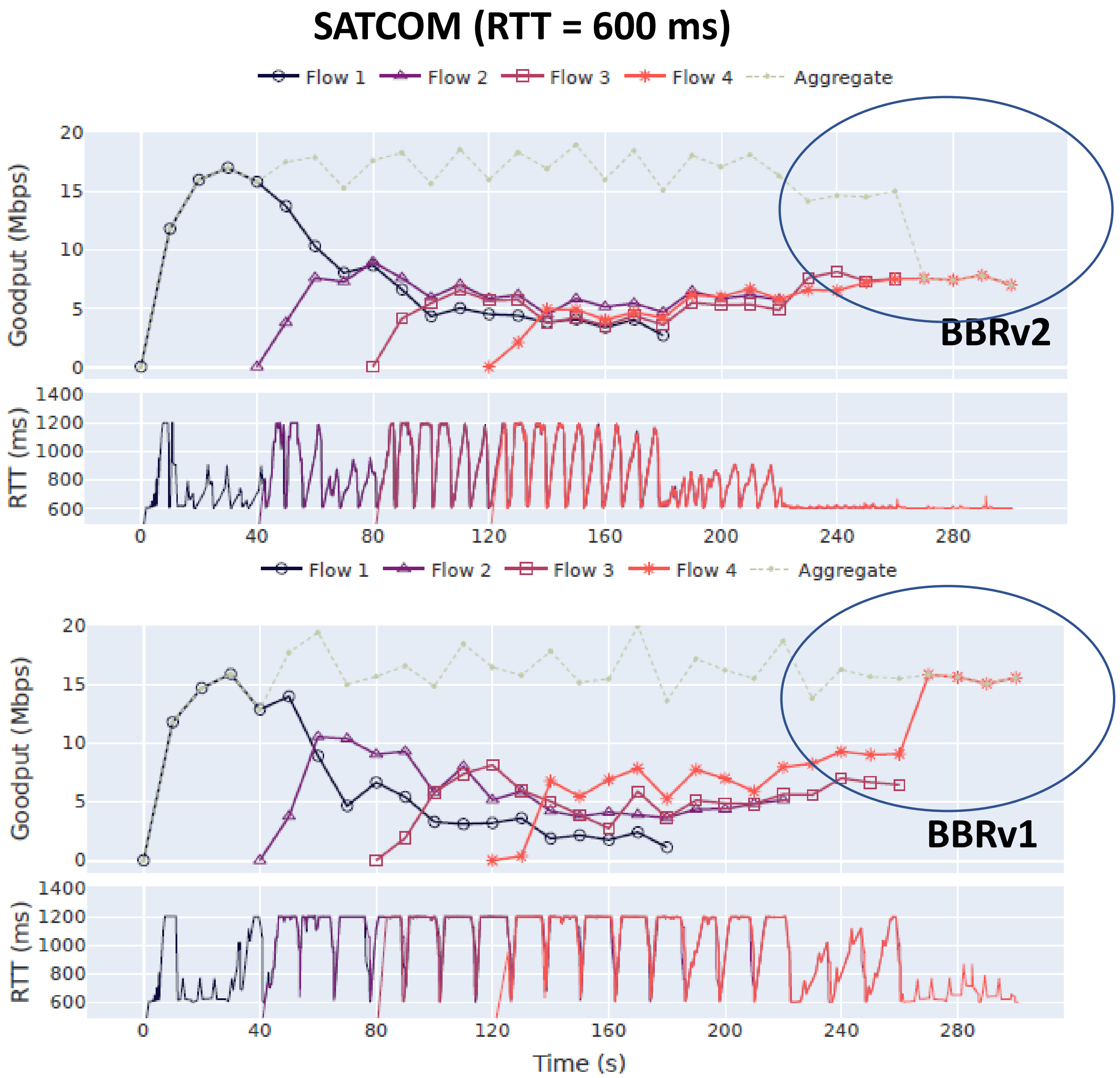
SATCOM (RTT = 600 ms)



BBRv2 achieves <u>better long-term fairness</u>

4 Results: Latecomer fairness with BBR

University of Stavanger



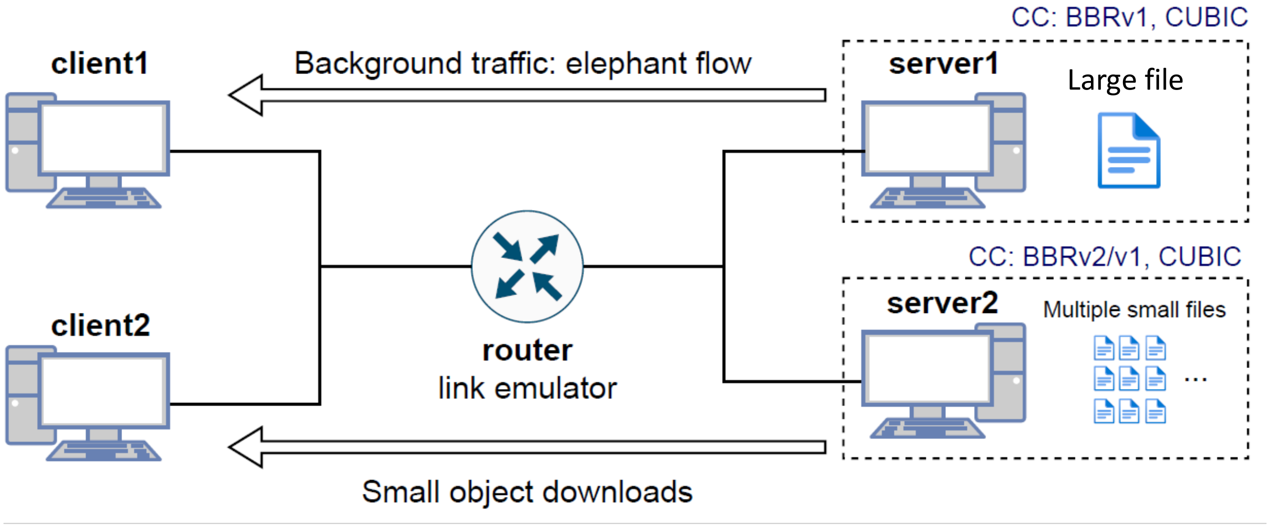
But BBRv2 <u>fails to recover available bandwidth</u> after the rest of the flows end

We have seen that this is a consequence of the long RTT

University of Stavanger

We set up four different experimental scenarios:

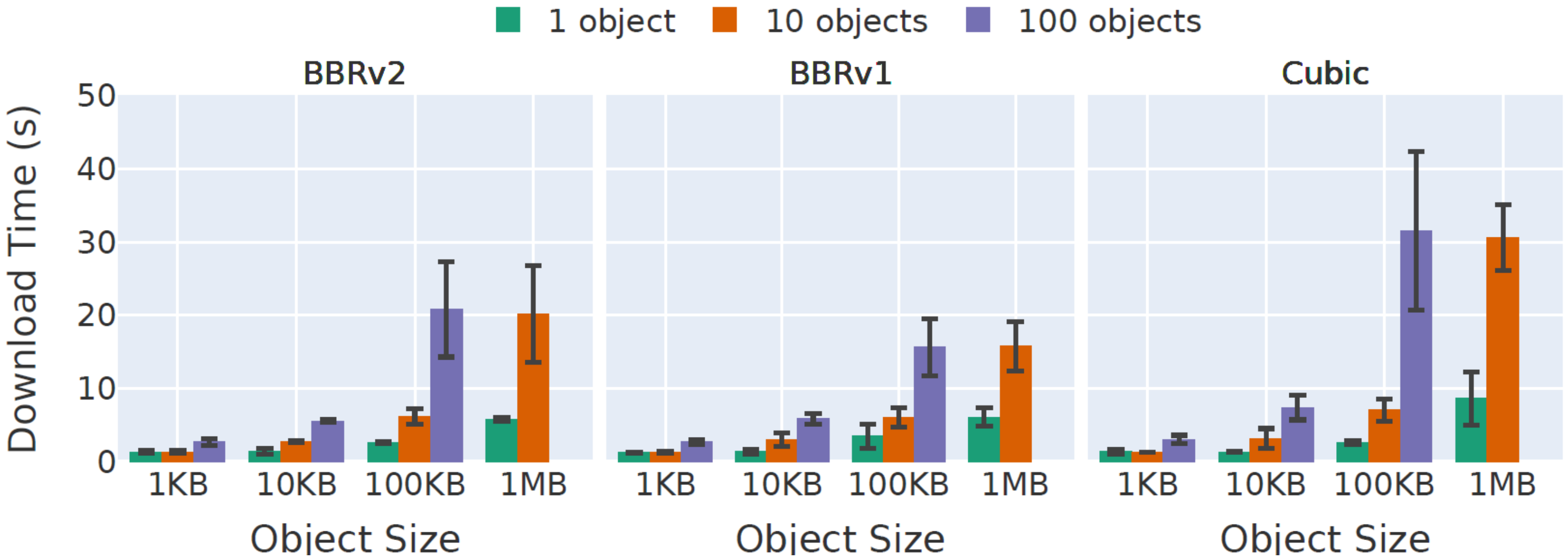
1. Single-Flow Bulk Download
2. Multi-Flow Fairness
3. Latecomer Issue
4. **Mice versus Elephant flows**



Implementation: ngtcp2

Results: Mice vs Elephant Flows

Background Traffic
BBRv1



5

Discussion and Conclusion

University of Stavanger

5 Discussion

Impact of Congestion Control choice

- **BBR provides better performance overall under lossy links**
  - BBRv1 provides the best performance
  - BBRv2 provides better fairness towards itself and towards CUBIC
  - BBRv2 latecomers are less aggressive and still converge fast

- BBRv2 seems to be the on the right path for fairer coexistence with other flows
  - But **BBRv2 performance suffers from the long RTT + packet loss** present in SATCOM links
  - Further BBR iterations could contemplate these long RTT scenarios

University of Stavanger

Impact of bandwidth asymmetry

- A **1:10 asymmetry** has proven to be a great challenge for ngtcp2, with great performance drops
- But results have shown that **an ACK policy such as picoquic's can maintain performance**
  - *picoquic sends around 10 times less ACK frames in our experiment results*
  - This stresses **further research into optimized ACK strategies for SATCOM networks**

University of Stavanger

Impact of QUIC implementation

- **picoquic outperforms ngtcp2 across CC algorithms**
  - Better resilience to packet loss
  - Better performance with bandwidth asymmetry
  - Possible reasons
    - Flow control window mechanism
    - ACK policies

5 Conclusion

University of Stavanger

To summarize:

- **BBR** seems to be a good candidate to yield better performance SATCOM networks
  - BBRv2 adds great improvements to fairness
  - But BBRv2 fairness and performance could be further improved for SATCOM-like scenarios (i.e. high BDP and packet loss)
- **Bandwidth asymmetry** is a problem in the absence of satellite-optimized ACK policies
- Picoquic's satelite-optimizations seem to be key for QUIC

Future directions:

- Improve the experimental setup:
  - Introduce a more realistic satellite model (L1-L2 mechanisms, packet loss models)
  - Use a wider set of QUIC implementations
- Propose and study different ACK strategies under various asymmetric setups
  - Could this be better implemented using MASQUE?
  - How do different CC coexist with these ACK policies?

Thank you! Questions?

Aitor Martin

Naeem Khademi

Department of Electrical Engineering and Computer Science

University of Stavanger, Norway