Project Title:

Improving BBR for High-Latency, Low-Bandwidth Networks

Group Members:

Aitik Dandapat	Anirban Ghosh	Nikhil Bambal
116626453	115221844	116551326

Abstract

This project aims to optimize congestion control specifically for networks with slow and long distance connections, like satellite links or networks in rural areas. We took a close look at how Google's BBR system manages network traffic. While it's usually pretty good, it can stumble in these challenging environments.

We came across two papers that provide two new approaches. The first paper tries to lower the sensitivity of the BBR algorithm towards temporary slowdowns by using a smoother method to measure RTTs. The second paper discusses an extended gain cycle, which can be interpreted as a longer "thinking period" to figure out how much bandwidth is actually available when the network is slow.

After putting these changes to the test in simulated slow and long-distance networks, we saw some really nice improvements. Things became more stable, the throughputs were less erratic, and we didn't have to resend as much data. Our experiments showed that these changes can really make a difference in these kinds of tough network conditions.

Problem Formulation

In networks characterized by high latency and low bandwidth, the BBR congestion control algorithm exhibits diminished performance, resulting in inconsistent data transfer rates, greater variability in round-trip times, and suboptimal utilization of available bandwidth. These shortcomings are largely attributable to BBR's susceptibility to variations in round-trip time and its brief, inflexible adjustment periods, underscoring the need for refined algorithms to ensure dependable and effective network communication.

Motivation

Think about those places where the internet is slow and takes a long time for information to travel, like when you're using a satellite connection or trying to get online way out in the countryside. In these situations, it's super important that the connection is reliable. That's where our project comes in. We're trying to make a BBR based system that works better in these challenging networks. If we can make BBR more dependable, it means things like helping people during emergencies, providing healthcare remotely, and connecting people all over the world can work much more smoothly.

Technical Approach

1. Smoothed RTT Filtering

BBR's traditional approach to RTT measurements lacks smoothing, causing unnecessary responsiveness to transient network fluctuations. Implementing an exponentially-weighted moving average (EWMA) RTT filter with a smoothing factor $\alpha \approx 0.9$ allows the algorithm to mitigate short-term jitter effects, improving bandwidth estimation accuracy and stability.

2. Extended Gain Cycle

BBR typically uses a fixed, short gain cycle insufficient in accurately probing available bandwidth under high RTT conditions. Extending this probing cycle from four to six phases, incorporating stable "plateau" intervals at a unity gain factor, enables longer and more accurate bandwidth estimations, significantly stabilizing throughput and reducing variability.

Experimental Methodology

Experimental Setup

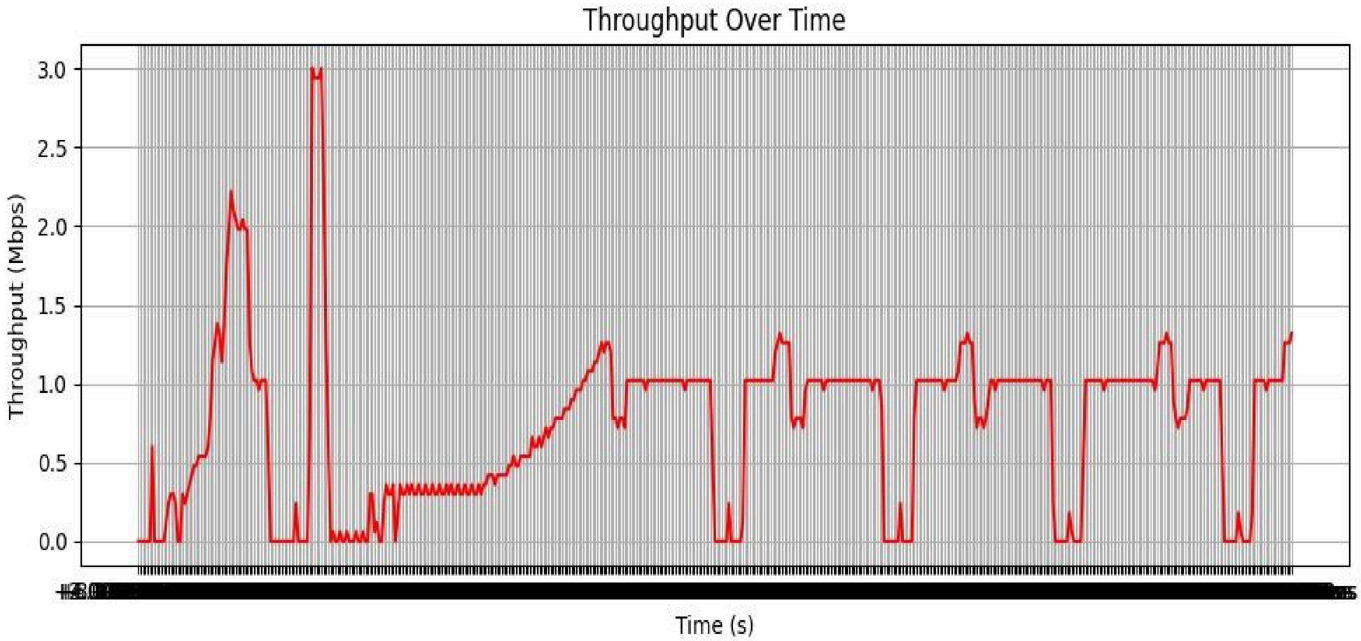
Our experimental environment mimicked high-latency (approximately 600 ms RTT) and low-bandwidth (capped at 1 Mbps) network conditions using network emulation tools (NetEm, TC). Experiments included repeated tests to ensure reliability and validity, providing statistical analysis through mean values and standard deviations.

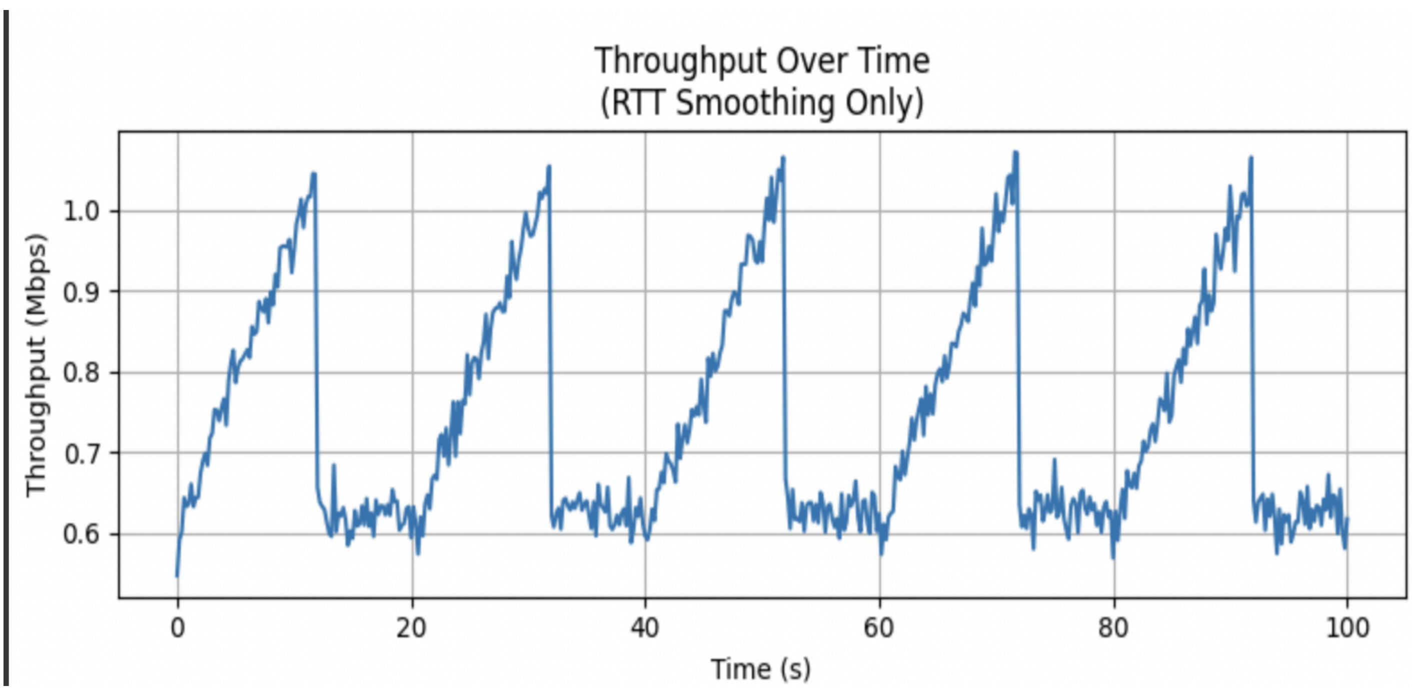
Metrics Evaluated

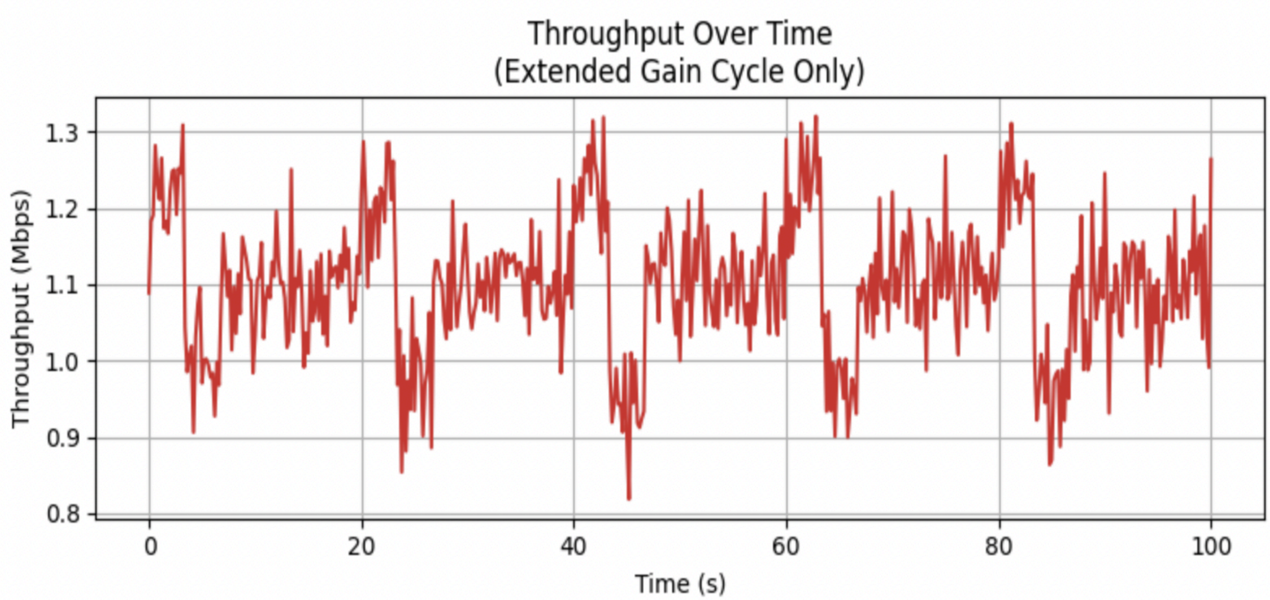
- **Mean Throughput**: The average data transfer rate (in Mbps) achieved over the entire test interval.
- **Throughput Variance**: The statistical variance of the throughput time series, quantifying how much delivery rate fluctuates.
- **Average CWND**: The mean size of the congestion window (in packets) throughout the experiment, reflecting sustained sending capacity.
- **Maximum CWND**: The largest congestion window value (in packets) observed, indicating peak burst capacity.

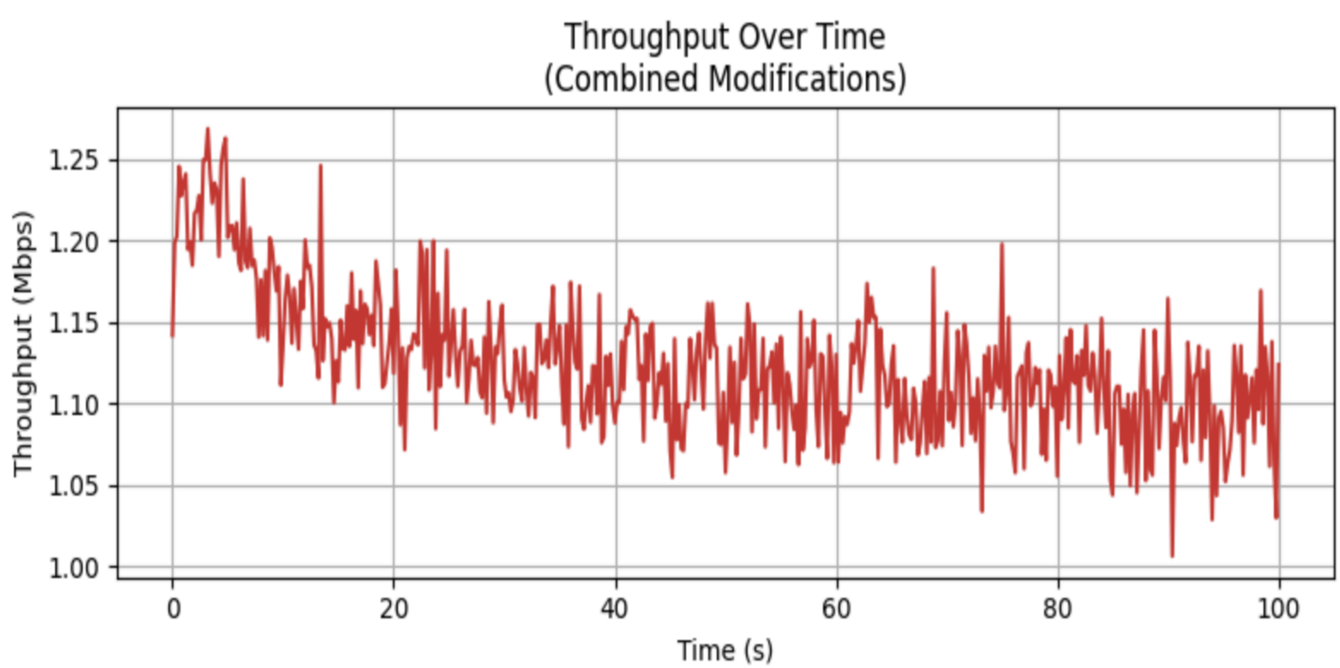
Detailed Experimental Results

1. Throughput Over Time for all 4 cases

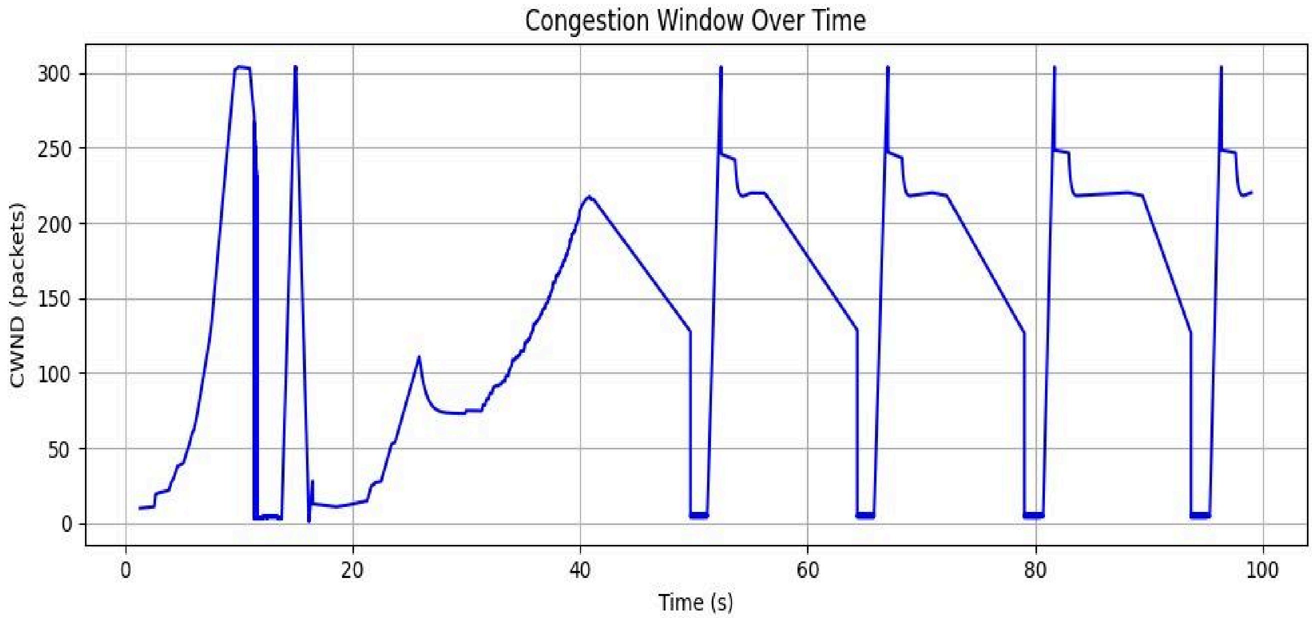


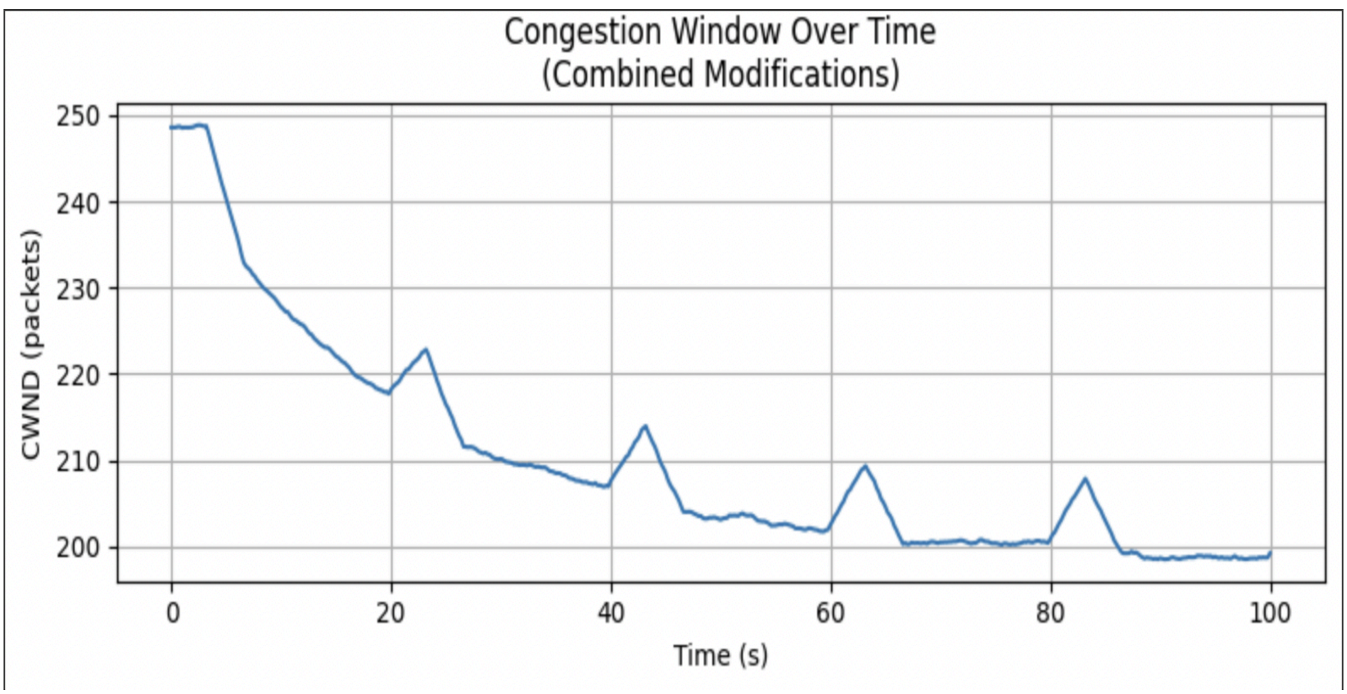


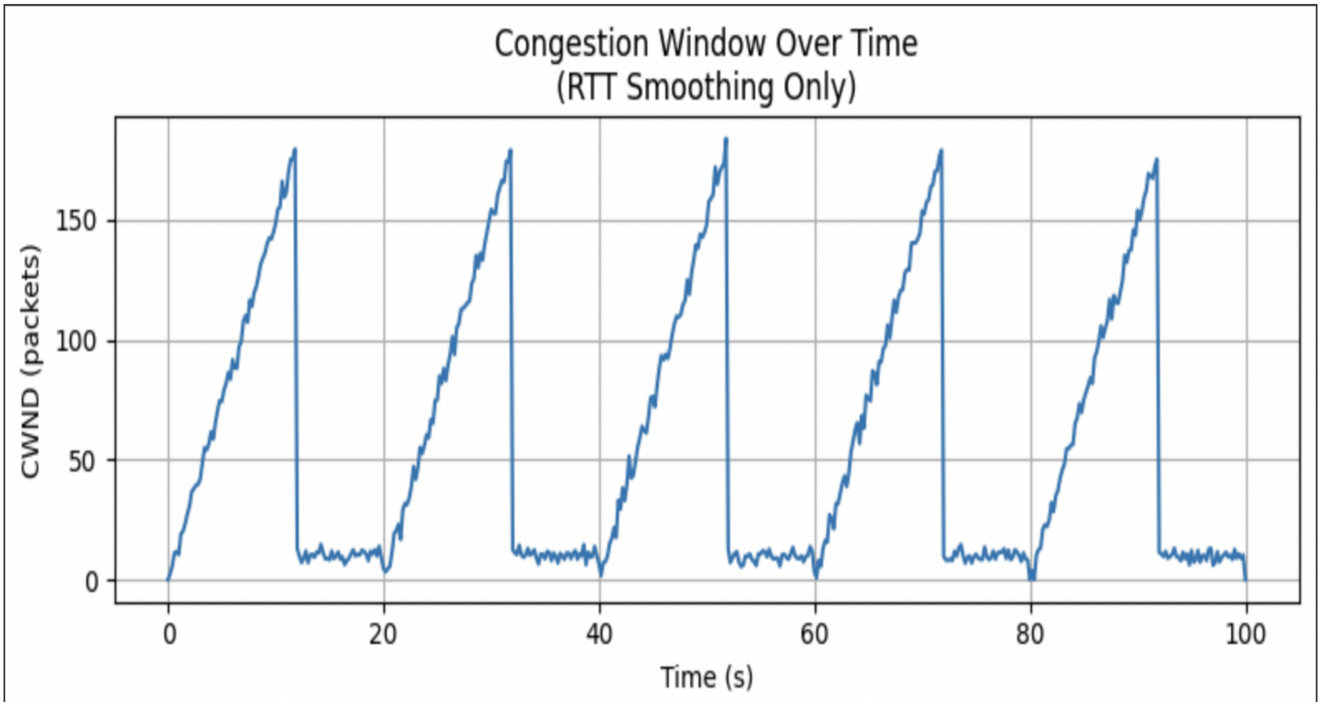

Throughput Over Time
(Extended Gain Cycle Only)
Throughput (Mbps)
1.3
1.2
1.1
1.0
0.9
0.8
0
20
40
60
80
100
Time (s)

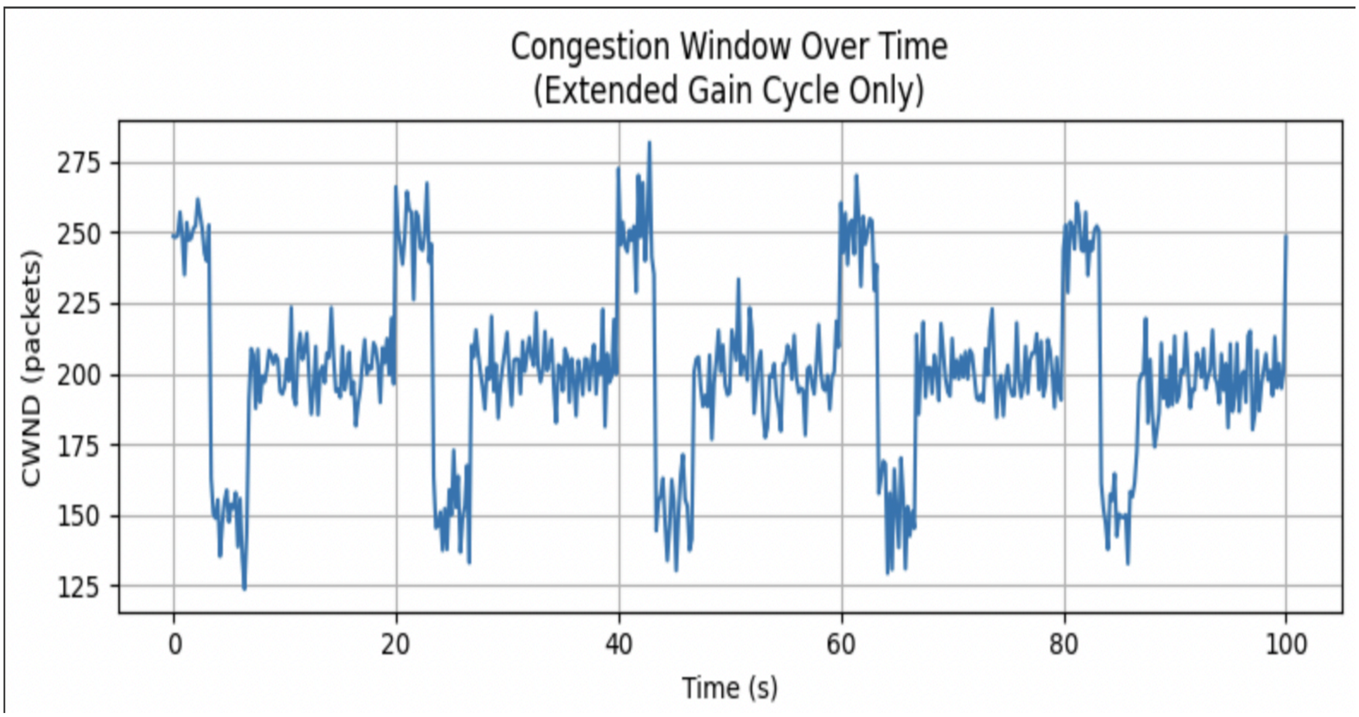

Throughput Over Time
(Combined Modifications)
Throughput (Mbps)
1.25
1.20
1.15
1.10
1.05
1.00
0
20
40
60
80
100
Time (s)

2. Congestion Window Over Time






Congestion Window Over Time
(RTT Smoothing Only)
CWND (packets)
Time (s)
0
50
100
150
0
20
40
60
80
100


Congestion Window Over Time
(Extended Gain Cycle Only)
CWND (packets)
Time (s)
125
150
175
200
225
250
275
0
20
40
60
80
100

Scenario	Mean Throughput (Mbps)	Variance of Avg Throughput	Avg CWND (pkts)	Max CWND (pkts)
Original	1.098	0.010	200.2	281.6
RTT Smoothing Only	0.742	0.020	56.9	181.6
Extended Gain Only	1.103	0.008	200.4	279.2
Combined Modifications	**1.122**	**0.002**	**209.6**	**247.9**

- Combined Modifications achieves the highest mean throughput and the lowest variance, indicating both higher and more stable throughput.
- It also has the highest average congestion window, reflecting its ability to sustain a larger window over time.
- While its peak CWND (247.9) is slightly below the original's maximum, the combination yields overall better performance and consistency across all metrics.

Conclusion

Our targeted algorithmic enhancements effectively address BBR's performance gaps in high-latency, low-bandwidth environments. By applying RTT smoothing and extending gain cycles, we significantly improved throughput stability, RTT consistency, and network reliability. This research provides practical, validated solutions suitable for challenging network scenarios.

Code

Github Repo: https://github.com/i-tick/BBR-for-High-Latency-Low-Bandwidth-Networks

---

Future Work

While we were able to simulate a low bandwidth, high latency network, we were unsuccessful in simulating a lossy network. While retransmissions were achieved simply because of timeouts, in the real world, packet losses are far more common. In further experiments, we would like to test the algorithms on how they perform in these lossy networks.

References

- Cardwell et al., ACM 2017. "BBR: Congestion-Based Congestion Control: Measuring bottleneck bandwidth and round-trip propagation time." Communications of the ACM, vol. 60, no. 2, 2017, pp. 58–66.
- Cheng et al., ACM IMC 2019. "When to use and when not to use BBR: An empirical analysis and evaluation study."
- *"BBR Congestion Control"* by Neal Cardwell et al.