

# BBRv3 Startup Behavior: Analysis and Fairness Enhancements

Imtiaz Mahmud

Lawrence Berkeley National Laboratory Lawrence Berkeley National Laboratory Lawrence Berkeley National Laboratory  
Berkeley, CA, USA Berkeley, CA, USA Berkeley, CA, USA  
imahmud@lbl.gov kwu@lbl.gov asim@lbl.gov

Kesheng Wu

Alex Sim

Anirban Mandal

RENCI, University of North Carolina  
Chapel Hill, NC, USA  
anirban@renci.org

Ewa Deelman

USC Information Sciences Institute  
Marina del Rey, CA, USA  
deelman@isi.edu

**Abstract**—BBRv3, the latest iteration of Google’s BBR congestion control algorithm, has shown significant performance improvements in high-bandwidth networks. However, our analysis reveals that BBRv3’s startup phase can lead to persistent fairness issues, where flows that initially acquire a larger bandwidth share maintain their advantage throughout the connection lifetime. To address this, we propose three enhanced BBRv3 variants that modify the startup behavior. Our evaluation on the FABRIC testbed demonstrates that these variants significantly improve fairness metrics, with BBRv3e1 achieving up to 15% improvement in Jain’s fairness index while preserving over 95% link utilization across diverse network scenarios. These improvements enable more equitable resource allocation in high-bandwidth networks, ensuring that BBRv3 can be deployed with confidence.

**Index Terms**—BBR, congestion control, fairness, RTT fairness, network protocols, high-bandwidth networks

## I. INTRODUCTION

The evolution of Internet infrastructure has led to the proliferation of high-bandwidth, low-latency networks that challenge traditional congestion control algorithms (CCAs). In such environments, conventional loss-based CCAs like TCP Reno and CUBIC often fail to fully utilize available bandwidth because they rely on packet loss as the primary congestion signal [1], [2]. This limitation is particularly problematic for large “elephant” flows over long-haul networks, where adopting a more suitable CCA can substantially improve end-to-end performance [3]–[7].

Google’s BBR (Bottleneck Bandwidth and Round-Trip-Time (RTT)) CCA family was designed to address these challenges by using bottleneck bandwidth and RTT estimates rather than packet loss as the primary congestion signals [5]–[7]. The algorithm operates through four phases—startup, drain, probe bandwidth, and probe RTT—with the startup phase allowing new flows to rapidly acquire their bandwidth share by sending at an aggressive rate.

BBRv3, the latest iteration of the BBR family, incorporates several improvements over its predecessors, including enhanced bandwidth probing mechanisms and better handling of

network variations [7]. However, fairness remains a persistent challenge when multiple flows compete for shared resources. The startup phase, while intended to enable fair bandwidth acquisition, can paradoxically lead to unfair outcomes when flows start at different times or under different network conditions [8].

Our comprehensive experimental analysis on the FABRIC testbed [9] reveals a critical fairness issue in BBRv3: flows that acquire a larger bandwidth share during startup tend to maintain this advantage throughout their lifetime, effectively suppressing competing flows. This behavior, also observed qualitatively in prior BBR studies, becomes more pronounced with newer BBR iterations and suggests that optimizations for individual flow performance may inadvertently compromise inter-flow fairness.

The startup phase in BBRv3 operates with a pacing gain of  $4 \times \ln(2) \approx 2.77$  and doubles the congestion window for each RTT [7]. It terminates when either (1) packet loss or explicit congestion notification (ECN) exceeds a threshold, (2) three rounds complete without significant bandwidth increase, or (3) the in-flight data exceeds the estimated bandwidth-delay product (BDP), indicating that the path is full. While this aggressive design enables rapid convergence to available bandwidth, it also creates an inherent advantage for flows that establish higher rates early in startup.

This paper makes the following key contributions:

- **Empirical Analysis:** We provide an experimental characterization of fairness issues in BBRv3, demonstrating how startup behavior affects long-term flow dynamics.
- **Enhanced Variants:** We propose three BBRv3 variants (BBRv3e1, BBRv3e2, BBRv3e3) with targeted startup-phase modifications that improve fairness through increased aggressiveness, reduced loss responsiveness, or their combination.
- **Evaluation of Fairness–Throughput Trade-offs:** We evaluate our variants on the FABRIC testbed across diverse RTTs (35–145 ms), bandwidths (1–25 Gbps), and buffer sizes (1–4 BDP), and quantify fairness and

utilization using Jain’s fairness index and link utilization metrics.

- **Startup-Focused Fairness Enhancements:** We isolate BBRv3’s startup phase as a primary mechanism behind persistent unfairness and argue that low-complexity adjustments to pacing gains and exit conditions can restore fairness, in contrast to more intrusive RTT-aware proposals such as BBR-EFRA [10] and BBR-ACD [11].

The remainder of this paper is organized as follows: Section II reviews related work in congestion control fairness. Section III presents our experimental observations of BBRv3 fairness issues. Section IV describes our enhanced variants and evaluation methodology. Section V presents experimental results and discussion, and Section VI concludes with future research directions.

## II. RELATED WORK

The introduction of BBR marked a shift from loss-based congestion control to model-based control using bottleneck bandwidth and RTT measurements [5]–[7]. Initial evaluations demonstrated significant throughput improvements, but subsequent studies revealed fairness and coexistence issues when BBR competes with traditional loss-based algorithms [8], [12], [13].

### A. Fairness in Congestion Control

Fairness in network congestion control has long been studied using metrics such as Jain’s fairness index, which quantifies the equality of resource allocation among competing flows [14]. Factors such as RTT differences, buffer sizing, queue management policies, and algorithm-specific behaviors can all create fairness problems [15], [16].

### B. BBR Evolution and Fairness Challenges

The original BBR algorithm achieved high throughput but exhibited notable fairness issues in multi-flow scenarios [17], [18], motivating the development of BBRv2 with improved loss detection and modified probing behavior [19]. Subsequent work proposed additional fairness enhancements: BBR-EFRA modifies the bandwidth probing mechanism to improve inter-flow equity [20], Pan et al. use an adaptive congestion window to improve RTT fairness [10], and Mahmud et al. introduce BBR-ACD, which selectively responds to retransmissions to mitigate high retransmission rates and fairness issues [11].

These approaches show that BBR’s fairness can be improved, but they often rely on more intrusive RTT-aware mechanisms and broader redesigns of the congestion control logic. Our goal is instead to narrow the root cause of persistent unfairness in BBRv3 to its startup behavior and to explore how small, localized parameter changes during startup can recover fairness while keeping the rest of the BBRv3 design intact.

Despite these efforts, BBRv3’s fairness characteristics remain understudied in realistic network environments, where most evaluations have focused on simulation studies or limited experimental setups.

## III. BBR EVOLUTION: EXPERIMENTAL OBSERVATIONS

To understand the fairness characteristics of BBR variants, we conducted comprehensive experiments using the FABRIC testbed. Our experimental setup, illustrated in Fig. 1, employs a dumbbell topology spanning four distinct geographic locations to achieve realistic RTT variations (35 ms, 56 ms, and 145 ms). The bottleneck router allows precise control over bandwidth and queue parameters using traffic control (tc) commands.

### A. Experimental Methodology

Our evaluation focuses on two-sender scenarios where flows compete over a shared bottleneck. We systematically varied the following network parameters:

- Bottleneck bandwidth: 1 Gbps, 10 Gbps, 25 Gbps
- Queue sizes:  $1 \times \text{bandwidth-delay product (BDP)}$ ,  $2 \times \text{BDP}$ ,  $4 \times \text{BDP}$
- Round-trip times: 35 ms, 56 ms, 145 ms
- Traffic load: The number of flows was scaled with bandwidth, set to 10 flows per sender for every 1 Gbps of capacity.
- Queue management: First-in, first-out (FIFO)

Each experiment ran for 150 seconds to allow the flows to reach a steady state. To observe interactions during the startup phase, flows were initiated near-simultaneously. We note that minor, hardware-induced delays in flow start times occurred due to the distributed nature of the FABRIC testbed, but we did not systematically vary start-time offsets (e.g., starting flows 1 s apart). Quantifying how controlled start-time offsets impact Jain’s fairness index and long-term throughput remains an important direction for future work.

### B. Fairness Issues in BBRv3

Fig. 2 demonstrates the core fairness problem observed across all BBR variants, with the issue becoming more pronounced in newer versions. The figure shows congestion window (CWND) evolution for competing flows under identical network conditions (1 Gbps bandwidth,  $1 \times \text{BDP}$  buffer). A critical observation emerges: whichever flow manages to acquire a larger share during the initial startup phase successfully retains that advantage throughout the connection lifetime.

This behavior is rooted in BBR’s model-driven approach and its startup dynamics. When a new flow attempts to join an already-established flow, it faces several disadvantages:

- 1) **Less Responsive to Losses:** Unlike loss-based algorithms (e.g., CUBIC, Reno) that aggressively reduce their rate upon loss, BBR seeks to sustain the maximum observed delivery rate. This makes it comparatively non-yielding, preventing new flows from claiming their fair share of the bandwidth.
- 2) **Startup Suppression:** The established flow’s ongoing traffic creates additional congestion signals that can cause the new flow to exit startup prematurely, preventing it from acquiring a fair share.
- 3) **Persistent Inequality:** Once the startup phase concludes, both flows enter bandwidth probing phases

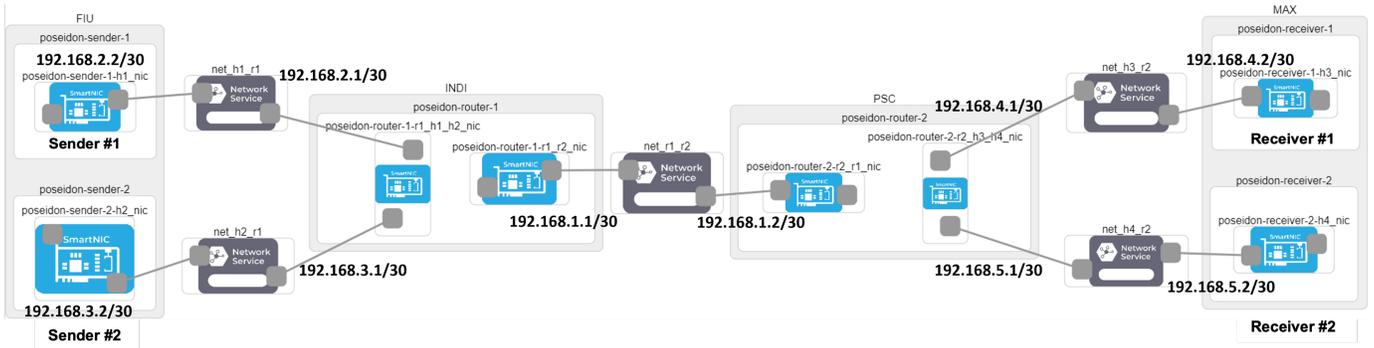


Fig. 1. Dumbbell network topology used for data generation on the FABRIC testbed [9]. The private Layer 2 network is an overlay on underlying physical networks spanning four distinct geographic locations. We vary the geographical locations to obtain round-trip times (RTTs) of 145 ms, 35 ms, and 56 ms, and we configure the bottleneck router (poseidon-router-1) using the Linux `tc` command to control both bandwidth and queue size.

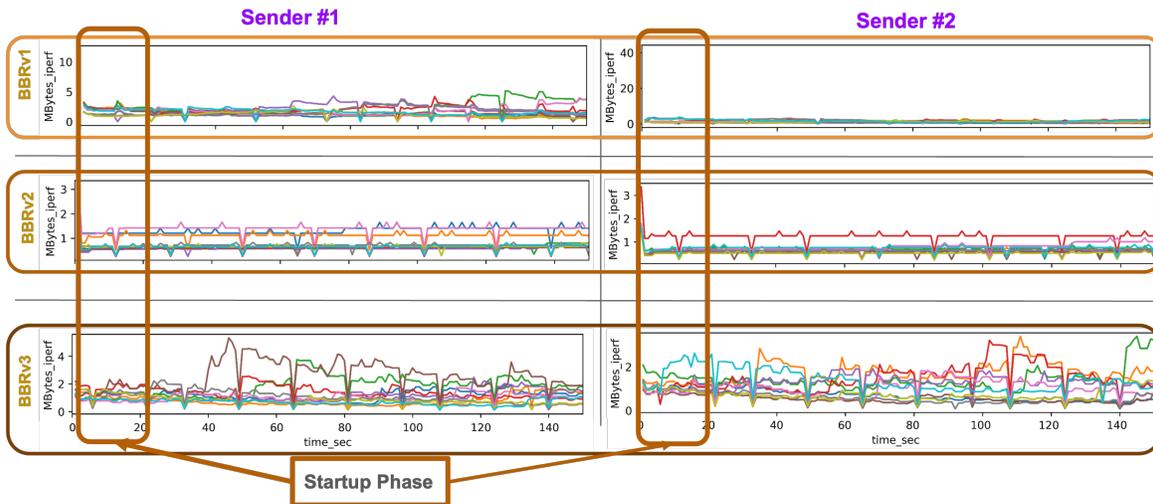


Fig. 2. Evolution of congestion window (CWND) for competing flows using different BBR variants. Under identical network conditions (1 Gbps bandwidth and  $1 \times \text{BDP}$  buffer), the flow that acquires a larger share during the startup phase tends to retain that advantage for the entire experiment, illustrating the persistence of startup-induced unfairness.

where they make only incremental adjustments, insufficient to correct the initial imbalance.

### C. Root Cause Analysis

Our analysis identifies the startup behavior as one of the primary source of fairness issues in BBRv3. The algorithm introduced a less aggressive pacing gain of  $4 \times \ln(2) \approx 2.77$  and doubles the sending rate each RTT during startup. It also introduced new early exit criteria that can cause premature departure from the startup phase. In combination, these mechanisms create several problematic scenarios:

- **Early Exit Conditions:** An overly sensitive response to Loss or ECN signals during startup can trigger premature startup termination, preventing new flows from acquiring a fair share when competing with established ones.
- **Insufficient Aggressiveness:** In highly competitive scenarios, the new moderate pacing gain may be inadequate for new flows to successfully challenge established flows for bandwidth.
- **Timing Sensitivity:** The startup success becomes heavily dependent on arrival timing relative to other flows, creating unfair advantages for earlier arrivals.

These observations motivate our proposed modifications to the BBRv3 startup behavior, targeting these specific mechanisms while preserving the algorithm’s performance benefits.

## IV. SOLUTION: BBRV3 ENHANCED VARIANTS

Based on our analysis of fairness issues in BBRv3’s startup behavior, we propose three enhanced variants that address different aspects of the problem. Each variant modifies specific parameters of the startup phase while maintaining compatibility with the core BBR framework.

### A. Standard BBRv3 Startup Behavior

Before presenting our enhancements, we formally characterize the standard BBRv3 startup behavior [7]:

- **Pacing Gain:**  $4 \times \ln(2) \approx 2.77$
- **Rate Increase:** Congestion window doubles each RTT
- **Exit Conditions:**
  - 1) Packet loss or ECN signal exceeds threshold
  - 2) Three consecutive rounds without significant bandwidth increase
  - 3) Maximum startup duration reached

## B. Enhanced Variants

1) *BBRv3e1: Increased Aggressiveness*: BBRv3e1 addresses scenarios where the standard startup pacing gain is insufficient for new flows to compete effectively with established flows. To address this, we reinstated the previous pacing gain of  $2/\ln(2) \approx 2.89$  used in BBRv1 and BBRv2 [5], [6], while maintaining the standard loss-responsive exit conditions:

- **Pacing Gain**:  $2/\ln(2) \approx 2.89$
- **Rate Increase**: Congestion window doubles each RTT
- **Exit Conditions**: Same as standard BBRv3

This approach enables new flows to probe more aggressively during startup, increasing their likelihood of acquiring fair bandwidth shares when competing with established flows.

2) *BBRv3e2: Reduced Loss Responsiveness*: BBRv3e2 targets scenarios where premature startup termination due to loss signals prevents fair bandwidth acquisition. This variant maintains the standard pacing gain but removes loss-based exit conditions:

- **Pacing Gain**:  $4 \times \ln(2) \approx 2.77$
- **Rate Increase**: Congestion window doubles each RTT
- **Exit Conditions**:

- 1) Three consecutive rounds without significant bandwidth increase
- 2) Maximum startup duration reached

By ignoring loss and ECN signals during startup, this variant allows flows to complete their bandwidth probing process even in high-congestion scenarios.

3) *BBRv3e3: Combined Approach*: BBRv3e3 combines the benefits of both previous variants, employing increased aggressiveness with reduced loss responsiveness:

- **Pacing Gain**:  $2/\ln(2) \approx 2.89$
- **Rate Increase**: Congestion window doubles each RTT
- **Exit Conditions**:
  - 1) Three consecutive rounds without significant bandwidth increase
  - 2) Maximum startup duration reached

This variant represents the most aggressive approach, designed for scenarios with high competition and significant fairness challenges.

## C. Fairness Metrics and Evaluation Framework

We employ two primary metrics to evaluate the performance of our enhanced variants:

1) *Link Utilization*: To measure resource efficiency, we define link utilization as:

$$\text{Link Utilization} = \frac{\sum_{i=1}^n G_i}{\beta_\tau} \quad (1)$$

where  $\beta_\tau$  represents the total available bandwidth at the bottleneck for a given scenario,  $G_i$  denotes the goodput achieved by flow  $i$ , and  $n$  indicates the total number of flows. A value of 1 signifies complete link utilization, while lower values indicate suboptimal resource usage.

2) *Jain's Fairness Index*: We assess bandwidth distribution equity using Jain's fairness index [14]:

$$\text{Jain's Fairness Index} = \frac{(\sum_{i=1}^n T_i)^2}{n \cdot \sum_{i=1}^n T_i^2} \quad (2)$$

where  $T_i$  denotes the throughput achieved by flow  $i$ . This metric ranges from  $1/n$  (completely unfair) to 1 (perfectly fair), providing a quantitative measure of resource allocation equality among competing flows.

## D. Experimental Configuration

Our comprehensive evaluation encompasses multiple network scenarios as summarized in Table I. We systematically test each proposed variant against both itself (measuring intra-protocol fairness) and CUBIC (measuring inter-protocol coexistence) across varying network conditions.

TABLE I  
SUMMARY OF EXPERIMENTAL SCENARIOS CONSIDERED FOR THE COMPARISON.

| CCA1 vs CCA2   | AQM                 | Queue Length                                       | Bottleneck Bandwidth   | RTT                        |
|--|---------------------|--|------------------------|----------------------------|
| BBRv1 vs BBRv1<br>BBRv1 vs CUBIC<br>BBRv2 vs BBRv2<br>BBRv2 vs CUBIC<br>BBRv3 vs BBRv3<br>BBRv3 vs CUBIC                   | FIFO                | $1 \times \text{BDP}$                              | 1 Gbps                 | 145 ms                     |
| BBRv3e1 vs BBRv3e1<br>BBRv3e1 vs CUBIC<br>BBRv3e2 vs BBRv3e2<br>BBRv3e2 vs CUBIC<br>BBRv3e3 vs BBRv3e3<br>BBRv3e3 vs CUBIC | FQ-CoDel<br><br>RED | $2 \times \text{BDP}$<br><br>$4 \times \text{BDP}$ | 10 Gbps<br><br>25 Gbps | 35 ms<br><br>35 ms / 56 ms |

Each experiment ran for 150 seconds to allow the flows to reach a steady state. In addition to this FIFO baseline, we also evaluated active queue management (AQM) schemes, specifically FQ-CoDel and RED, as summarized in Table I. To observe interactions during the startup phase, flows were initiated near-simultaneously. We note that minor, hardware-induced delays in flow start times occurred due to the distributed nature of the FABRIC testbed, but we did not systematically vary start-time offsets (e.g., starting flows 1 s apart). Quantifying how controlled start-time offsets impact Jain's fairness index and long-term throughput remains an important direction for future work.

## V. EXPERIMENTAL RESULTS AND ANALYSIS

This section presents comprehensive experimental results evaluating our proposed BBRv3 enhanced variants across multiple network scenarios. The evaluation demonstrates significant fairness improvements while maintaining high link utilization.

### A. Detailed Performance Analysis

Fig. 3 presents detailed performance results across nine different network scenarios, organized by RTT characteristics and buffer depths. The results are averaged over three bandwidth scenarios (1, 10, and 25 Gbps) to provide representative performance metrics.

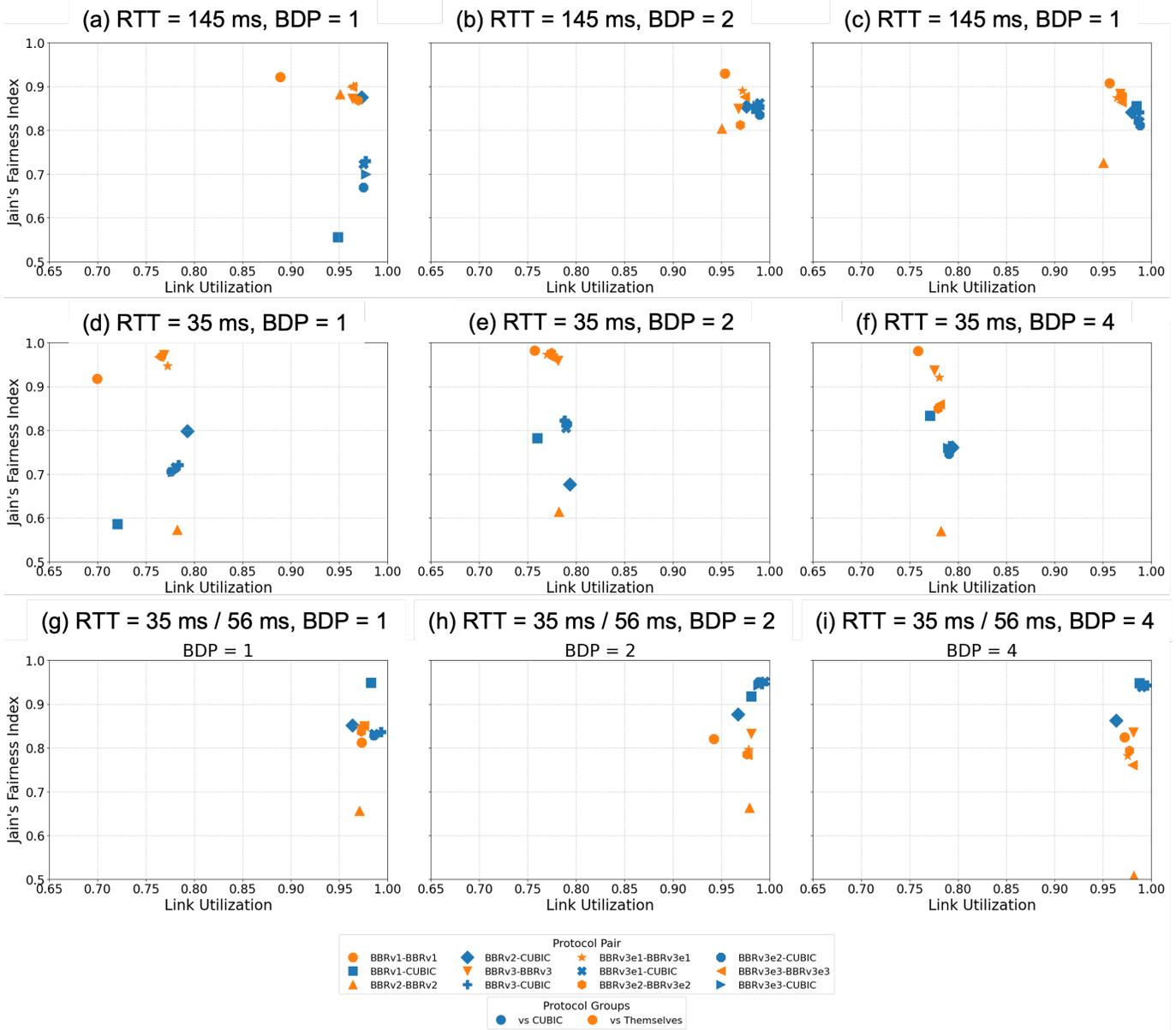


Fig. 3. Detailed performance of BBR variants in terms of Jain's fairness index and link utilization. Each subplot shows averages over the 1, 10, and 25 Gbps bandwidth scenarios for a given RTT and buffer configuration: (a)–(c) RTT 145 ms with BDP 1, 2, and 4, (d)–(f) RTT 35 ms with BDP 1, 2, and 4, and (g)–(i) mixed RTT where one sender has RTT 35 ms and the other 56 ms with BDP 1, 2, and 4, respectively.

1) *Long RTT Scenarios (145 ms)*: For long RTT scenarios (Fig. 3(a)–(c)), several key observations emerge:

**1×BDP Buffer**: All BBR variants struggle to achieve fairness when competing with CUBIC, with BBRv1 performing notably worse. BBRv3e1 demonstrates superior performance in both fairness and link utilization metrics, achieving approximately 10–15% improvement in Jain's fairness index compared to standard BBRv3.

**2×BDP Buffer**: The same pattern persists, with BBRv3e1 maintaining its advantage. The increased buffer size provides more room for bandwidth probing, but the fundamental fairness issues remain. BBRv3e1's increased aggressiveness during startup proves effective in acquiring fair shares.

**4×BDP Buffer**: Interestingly, in deep buffer scenarios,

standard BBRv3 performs slightly better than the enhanced variants. This suggests that with ample buffering, the aggressive startup modifications may be less beneficial, and the standard algorithm's more conservative approach suffices.

The results indicate that in moderate to low buffer scenarios with long RTTs, increasing startup aggressiveness (BBRv3e1) significantly improves fairness outcomes without sacrificing link utilization.

2) *Short RTT Scenarios (35 ms)*: Short RTT scenarios (Fig. 3(d)–(f)) reveal different performance characteristics:

**Intra-protocol Fairness**: BBRv2 exhibits the worst fairness when competing with itself in 1×BDP scenarios, but performs significantly better against CUBIC. This counterintuitive result suggests that BBRv2's behavior is highly sensitive to the

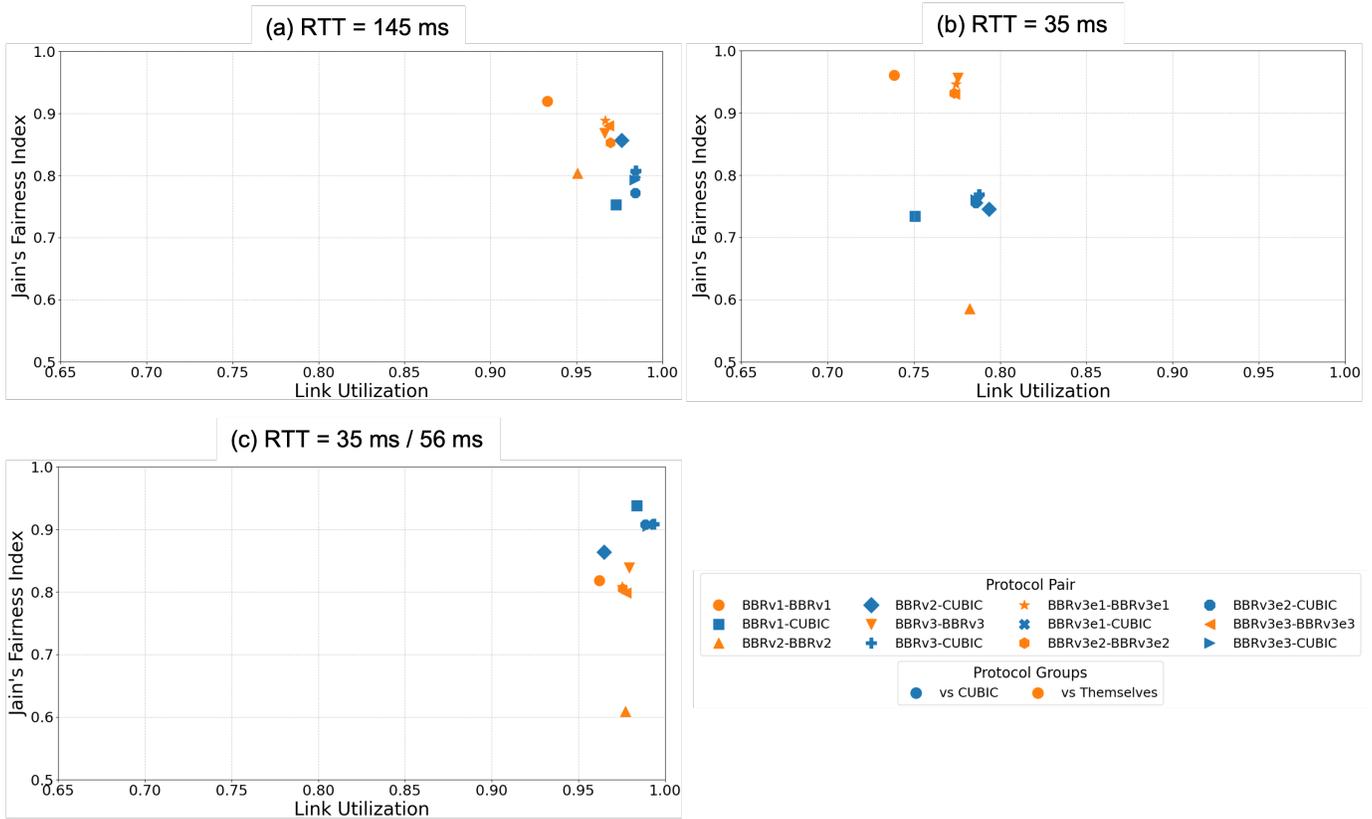


Fig. 4. Overall average performance of BBR variants in terms of Jain's fairness index and link utilization across all scenarios, grouped by RTT conditions. Subfigure (a) aggregates long-RTT scenarios (145 ms), (b) aggregates short-RTT scenarios (35 ms), and (c) aggregates mixed-RTT scenarios where one sender has RTT 35 ms and the other 56 ms.

competing algorithm type.

**Link Utilization Challenges:** All algorithms struggle to achieve high link utilization (below 80%) in short RTT scenarios. This phenomenon may result from the reduced opportunity for bandwidth probing in networks with short feedback loops.

**Variant Performance:** BBRv3 and BBRv3e3 perform comparably across all buffer sizes, suggesting that the combined modifications in BBRv3e3 are particularly effective in short RTT environments.

3) *Mixed RTT Scenarios (35 ms / 56 ms):* Mixed RTT scenarios (Fig. 3(g)-(i)) simulate realistic Internet conditions where flows experience different path characteristics:

**RTT Unfairness:** BBRv2 again demonstrates poor intra-protocol fairness, confirming its sensitivity to RTT differences. The algorithm appears to struggle with managing flows having different RTT characteristics.

**Enhanced Variant Performance:** BBRv3 and BBRv3e1 perform competitively, with BBRv3e1 showing marginal improvements in fairness, particularly when competing with CUBIC in 1×BDP and 2×BDP scenarios.

**Robustness:** The results demonstrate that our enhanced variants maintain robust performance across RTT variations, an important characteristic for real-world deployment.

## B. Overall Performance Comparison

Fig. 4 summarizes the average performance across all scenarios for each RTT configuration, providing a holistic view

of algorithm behavior:

1) *Long RTT Performance (145 ms):* BBRv3e1 clearly outperforms standard BBRv3 in fairness when competing with CUBIC, demonstrating that modest increases in startup aggressiveness yield significant benefits. The improvement is particularly pronounced in inter-protocol scenarios, suggesting better coexistence characteristics.

2) *Short RTT Performance (35 ms):* BBRv3 and BBRv3e1 exhibit nearly identical performance, indicating that in short RTT environments, the startup modifications have minimal impact. The rapid feedback loops in these scenarios may naturally limit the advantages that enhanced startup aggressiveness can provide.

3) *Mixed RTT Performance (35 ms / 56 ms):* The results again show competitive performance between BBRv3 and BBRv3e1, with slight advantages for the enhanced variant. This suggests that our modifications provide benefits without introducing significant drawbacks in diverse RTT environments.

## C. Discussion and Limitations

Our evaluation is conducted entirely on the FABRIC testbed, a powerful but still controlled research infrastructure. FABRIC lets us exercise fine-grained control over RTT, bandwidth, buffer size, and AQM, but it cannot capture the full diversity of public or cloud-based networks, including background traffic mixes, failure modes, and concurrent application workloads.

The reported fairness and utilization trends should therefore be interpreted as indicative rather than exhaustive, and validating BBRv3 and our enhanced variants on production networks remains important future work.

The proposed startup variants are primarily tuned for buffer sizes up to  $2 \times \text{BDP}$  and RTT ranges common in wide-area research and education networks. In deep-buffer scenarios such as  $4 \times \text{BDP}$ , standard BBRv3 can sometimes match or exceed the fairness of our variants, and mixed-RTT scenarios remain challenging, indicating that no single parameterization is universally optimal across all buffering and RTT regimes. Designing a more general solution that behaves robustly across arbitrary buffer sizes and RTT combinations is left as open work.

Our experiments also focus on long-lived flows on relatively low-loss paths. Increasing startup aggressiveness may affect short or small flows that complete within a few RTTs, or paths with higher random loss rates, by increasing traffic burstiness or triggering more retransmissions. Although our FABRIC experiments did not reveal severe problems in these regimes, a systematic study of short-flow latency and behavior under higher loss is an important direction for future investigation.

## VI. CONCLUSION AND FUTURE WORK

This paper demonstrates that targeted modifications to BBRv3's startup behavior can address key fairness issues without sacrificing link utilization. Through extensive evaluation on the FABRIC testbed, we showed that increasing startup aggressiveness, as implemented in our BBRv3e1 variant, improves Jain's fairness index by up to 15% across diverse network conditions, particularly in long RTT and moderate buffer scenarios, while maintaining high throughput.

Building on the success of these startup-phase modifications, our future work will investigate how tuning parameters in other BBRv3 phases—such as drain, probe bandwidth, and probe RTT—can further enhance performance and fairness, providing a path toward more equitable and efficient congestion control mechanisms.

## ACKNOWLEDGMENT

This work was funded by the U.S. Department of Energy under the Integrated Computational and Data Infrastructure (ICDI) for Scientific Discovery, grant number DE-SC0022328. Experimental data was collected on the FABRIC testbed, which is supported by the National Science Foundation. The authors thank the FABRIC team for their support and infrastructure.

## REFERENCES

- [1] Van Jacobson. Congestion avoidance and control. In *Proceedings of SIGCOMM '88*, pages 314–329, Stanford, CA, USA, Aug 1988. ACM.
- [2] Sally Floyd and Van Jacobson. Random early detection gateways for congestion avoidance. *IEEE/ACM Transactions on Networking*, 1(4):397–413, 1993.
- [3] Sangtae Ha, Injong Rhee, and Lisong Xu. Cubic: A new tcp-friendly high-speed tcp variant. *ACM SIGOPS Operating Systems Review*, 42(5):64–74, 2008.

- [4] Douglas Leith and Robert Shorten. H-tcp: Tcp for high-speed and long-distance networks. In *Proceedings of PFLDnet*, volume 2004. Citeseer, 2004.
- [5] Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Van Jacobson, and Jerry Vickers. Bbr: Congestion-based congestion control. In *Communications of the ACM*. ACM, 2016.
- [6] Neal Cardwell, Yuchung Cheng, Soheil Hassas Yeganeh, Priyaranjan Jha, Yousuk Seung, Kevin Yang, Ian Swett, Victor Vasiliiev, Bin Wu, Luke Hsiao, et al. Bbrv2: A model-based congestion control performance optimization. In *Proc. IETF 106th Meeting*, pages 1–32, 2019.
- [7] Neal Cardwell, Y Cheng, K Yang, D Morley, SY Hassas, P Jha, Y Seung, V Jacobson, I Swett, B Wu, et al. Bbrv3: Algorithm bug fixes and public internet deployment. *Presentation in CCGW at IETF*, 117, 2023.
- [8] Danesh Zeynali, Emilia N Weyulu, Seifeddine Fathalli, Balakrishnan Chandrasekaran, and Anja Feldmann. Promises and potential of bbrv3. In *International Conference on Passive and Active Network Measurement*, pages 249–272. Springer, 2024.
- [9] Ilya Baldin, Anita Nikolich, James Griffioen, Indermohan Inder S Monga, Kuang-Ching Wang, Tom Lehman, and Paul Ruth. Fabric: A national-scale programmable experimental network infrastructure. *IEEE Internet Computing*, 23(6):38–47, 2019.
- [10] Wansu Pan, Haibo Tan, Xiru Li, and Xiaofeng Li. Improved rtt fairness of bbr congestion control algorithm based on adaptive congestion window. *Electronics*, 10(5):615, 2021.
- [11] Imtiaz Mahmud, Geon-Hwan Kim, Tabassum Lubna, and You-Ze Cho. Bbr-acd: Bbr with advanced congestion detection. *Electronics*, 9(1):136, 2020.
- [12] Danesh Zeynali, Emilia N Weyulu, Seifeddine Fathalli, Balakrishnan Chandrasekaran, and Anja Feldmann. Bbrv3 in the public internet: a boon or a bane? In *Proceedings of the 2024 Applied Networking Research Workshop*, pages 97–99, 2024.
- [13] Jose Gomez, Elie F Kfoury, Jorge Crichigno, and Gautam Srivastava. Evaluating tcp bbrv3 performance in wired broadband networks. *Computer Communications*, 222:198–208, 2024.
- [14] Rajendra K Jain, Dah-Ming W Chiu, William R Hawe, et al. A quantitative measure of fairness and discrimination. *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 21, 1984.
- [15] Agnieszka Piotrowska. Performance evaluation of tcp bbrv3 in networks with multiple round trip times. *Applied Sciences*, 14(12):5053, 2024.
- [16] Imtiaz Mahmud, George Papadimitriou, Cong Wang, Mariam Kiran, Anirban Mandal, and Ewa Deelman. Elephants sharing the highway: Studying tcp fairness in large transfers over high throughput links. In *Proceedings of the SC'23 Workshops of the International Conference on High Performance Computing, Network, Storage, and Analysis*, pages 806–818, 2023.
- [17] Yuechen Tao, Jingjie Jiang, Shiyao Ma, Luping Wang, Wei Wang, and Bo Li. Unraveling the rtt-fairness problem for bbr: A queueing model. In *2018 IEEE global communications conference (GLOBECOM)*, pages 1–6. IEEE, 2018.
- [18] Geon-Hwan Kim, Imtiaz Mahmud, and You-Ze Cho. Fairness improvement of bbr congestion control algorithm for different rtt flows. In *2019 International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–2. IEEE, 2019.
- [19] Furong Yang, Qinghua Wu, Zhenyu Li, Yanmei Liu, Giovanni Pau, and Gaogang Xie. Bbrv2+: Towards balancing aggressiveness and fairness with delay-based bandwidth probing. *Computer Networks*, 206:108789, 2022.
- [20] Charles Kihungi Njogu, Wang Yang, Humphrey Waita Njogu, and Adrian Bosire. Bbr-with enhanced fairness (bbr-efra): A new enhanced rtt fairness for bbr congestion control algorithm. *Computer communications*, 200:95–103, 2023.