# Understanding BBR-v3 Dynamics over Starlink: An Experimental Evaluation of Satellite Internet

Rakshitha De Silva* and Shiva Raj Pokhrel†
*†School of Information Technology, Deakin University, Geelong, Australia
Email: rakshitha.desilva@deakin.edu.au*, shiva.pokhrel@deakin.edu.au†

***Abstract*—We develop testbed to study Google's Bottleneck Bandwidth and Round-trip propagation time (BBR)-v3 on SpaceX's Starlink, the world's leading Low Earth Orbit (LEO) satellite internet provider. By deploying servers across the globe, we compared BBR with well-known loss and delay-based Transmission Control Protocol (TCP) Congestion Control Algorithms (CCAs), Cubic, and Hybla. The results show that BBR delivers much higher throughput than the other CCAs, making full use of Starlink's network capacity. However, this comes at the cost of higher retransmissions due to its aggressive pacing, pointing to a trade-off between speed and stability. Overall, the combination of Starlink and BBR appears promising, but it requires careful tuning to balance performance and reliability for the future of satellite-based internet.**

***Index Terms*—BBR-v3, LEO satellites, Starlink, TCP**

## I. Introduction

Bottleneck Bandwidth and Round-trip propagation time (BBR)[1] Congestion Control Algorithm (CCA) builds an explicit path model from recent measurements of delivery rate, Round Trip Time (RTT), and loss, using it to pace sending and limit inflight data [1]. Compared to loss-based CCAs, it is known that BBR achieves higher throughput on shallow/random-loss bottlenecks and lower queuing delay on deep buffers. Building on this model-based design, BBR refines bandwidth/RTT estimation, fairness, and probe scheduling through multi-phase gain cycling and adaptive inflight control [2]. Explicitly targeting the path Bandwidth-Delay Product (BDP) with paced sending and a congestion window $\approx B_\theta \times RTT_{\min}$, while improving coexistence with loss-based flows [1], [3].

Starlink mega-constellation with more than $6{,}751$ active satellites orbiting at $\approx 550$ km, is transforming the Internet. Serving over 2.7M subscribers and expanding at a rate exceeding 5 Tbps/week, Starlink facilitates service through phased-array Ku user links, Ka feeder links, and interconnects the constellation via laser Inter-Satellite Links (ISLs) [4]. However, LEO-specific dynamics such as high Signal to Noise Ratio (SNR) variance, frequent handovers, variable and high RTT, challenge traditional loss/delay-based congestion control and create non-stationary path conditions that stress transport layer control loops [5]–[7]. This motivates a deeper experimental evaluation of BBR under realistic LEO satellite settings.

To that end, we develop a real distributed testbed as shown in Fig. 1 using a Starlink terminal in Melbourne and globally distributed Amazon Web Services (AWS) endpoints in São Paulo, London, Mumbai, Tokyo, and Sydney cities. Using `iperf3`, we benchmark BBR with Cubic [8] and Hybla [9], by measuring throughput, loss/retransmissions, congestion-window dynamics, and RTT variance [10]. Our study experimentally evaluates and captures how BBR's new congestion control leverages Starlink's capacity while experiencing LEO orbital dynamics.

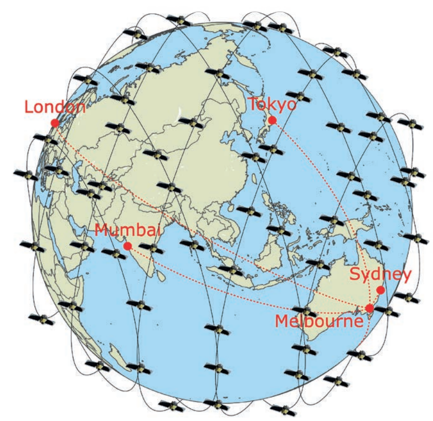

Fig. 1: Experimenting with a Starlink terminal in Melbourne (Australia) connected to AWS endpoints: São Paulo, London, Mumbai, Tokyo, and Sydney

## II. Background & Experiment Setup

### A. Relevant CCAs

BBR [1] fundamentally diverges from traditional loss or delay-based paradigms by explicitly modeling the network path's bottleneck bandwidth and RTT. Unlike other CCAs that infer congestion from loss signals or queue buildup, BBR operates by continuously estimating the maximum delivery rate ($B_\theta$), the minimum RTT ($RTT_{\min}$), and uses these to set its sending rate and congestion window. The sending rate is set as $R_{\text{sending}} = G \cdot B_\theta$ where $G$ is a gain factor that varies across different phases, *i.e.* startup, drain, exploring additional bandwidth, and refreshing the $RTT_{\min}$. The congestion window is calculated to match the BDP: $W_{\text{BBR}} = G_W \cdot B_\theta \cdot RTT_{\min}$ where $G_W$ is the congestion window gain that controls queue occupancy and inflight data [1], [3]. BBR-v3 builds upon BBR-v2 by enhancing queue control through dual inflight limits, improving robustness against Acknowledgement (ACK) aggregation, and formalising congestion window limits for

[1]For the rest of the paper, if not stated otherwise, BBR-v3 is referred to as BBR.

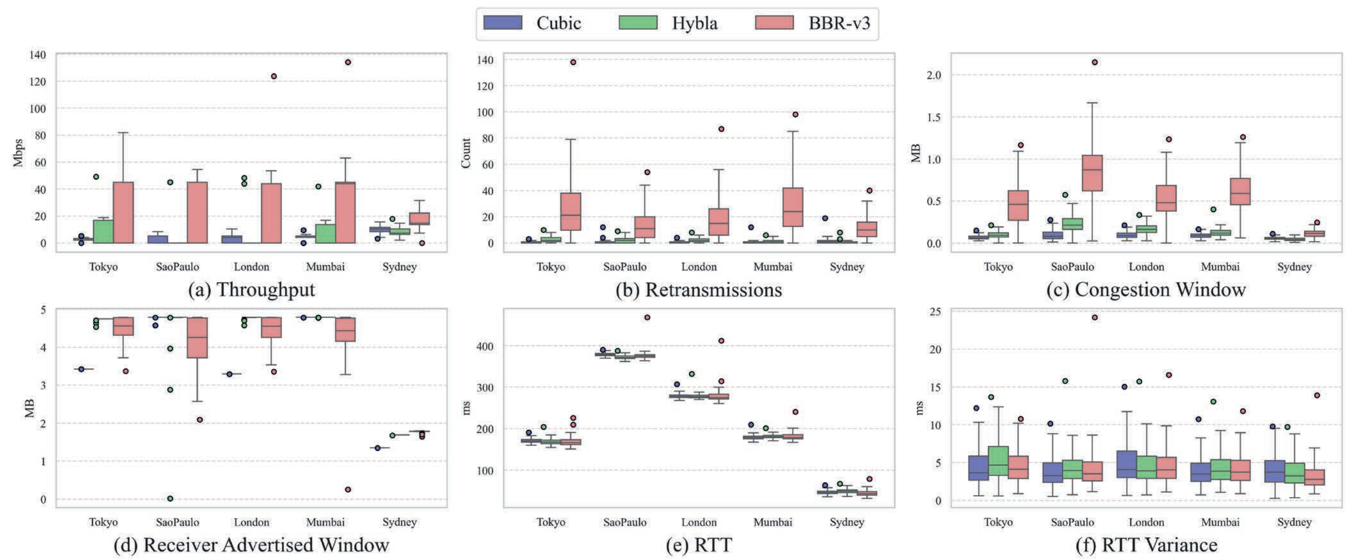


Fig. 2: Summarized illustration of uplink over Starlink for globally distributed server locations with different CCAs.

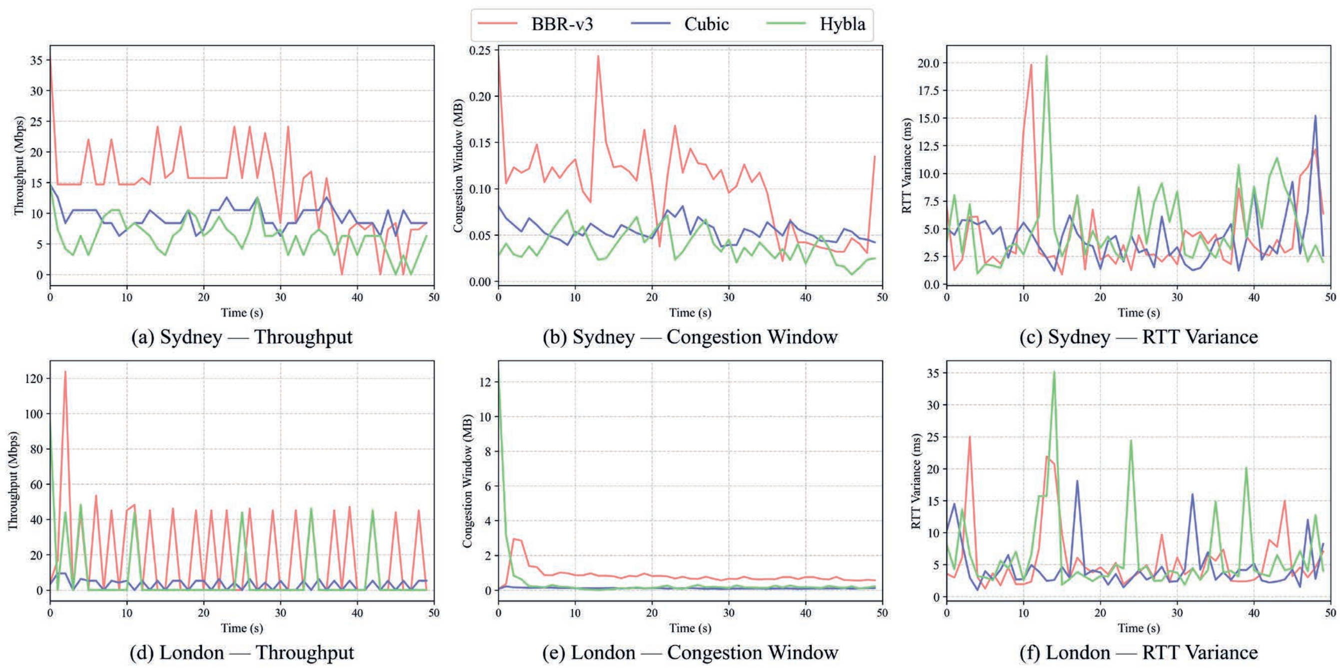


Fig. 3: Variation of throughput, congestion window, and RTT variance over time in the uplink for Sydney and London.

each state, resulting in greater stability and readiness for production use. While BBR-v2 advanced beyond BBR-v1 by incorporating loss and RTT sensitivity, BBR-v3 is the new standardized version designed for large-scale deployment [1]. We use Cubic [RFC 8312] [8] and Hybla [9] for tractability and benchmarking with BBR in our experiments. Cubic grows its congestion window with a time-based cubic function during congestion avoidance and exponentially during slow start, making it more aggressive yet stable after losses [8]. Hybla scales growth by normalizing the experienced RTT to offset high RTT, so that high-latency paths approach the throughput of low-latency ones, making it more suitable for satellite communication [9].

### B. Experiment Setup

As illustrated in Fig. 1, we deployed a geographically distributed experimental testbed comprising AWS cloud servers located in São Paulo, London, Mumbai, Tokyo, and Sydney. Each remote endpoint was provisioned as a Linux-based server instance within the AWS ecosystem with the default operating system (Ubuntu 24.04 LTS). The local component of the testbed was installed at Deakin University (Burwood, Melbourne) and comprised a Starlink next-generation standard user terminal equipped with a UTA-232 phased-array dish. A

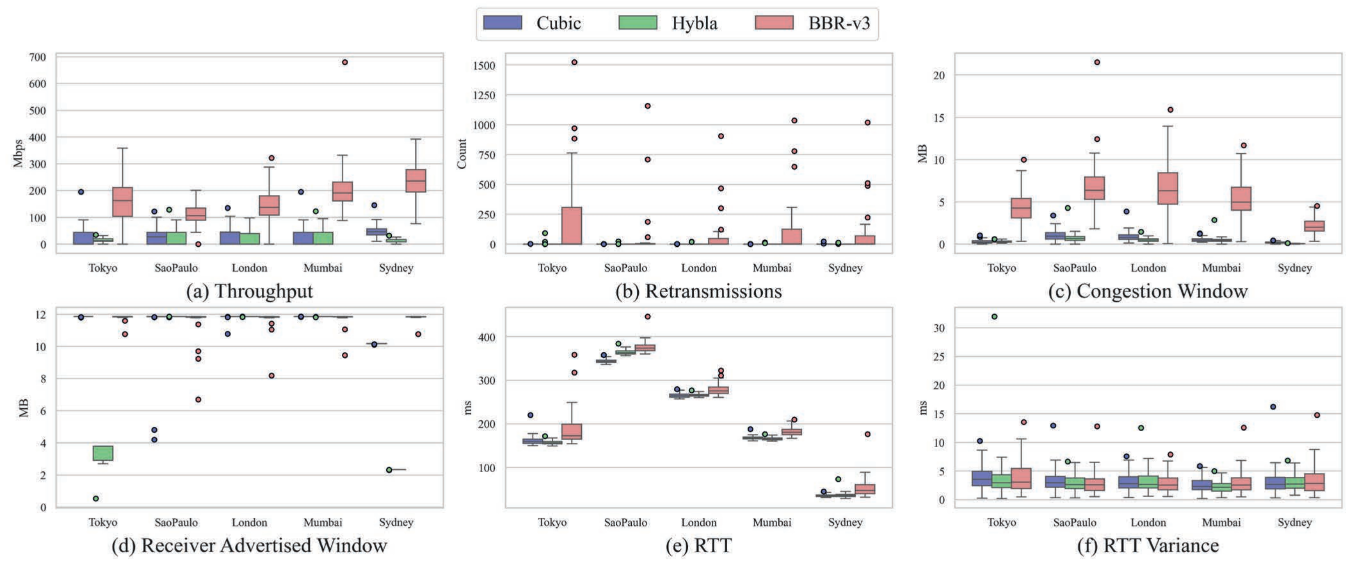


Fig. 4: Summarized illustration of downlink over Starlink for globally distributed server locations with different CCAs.

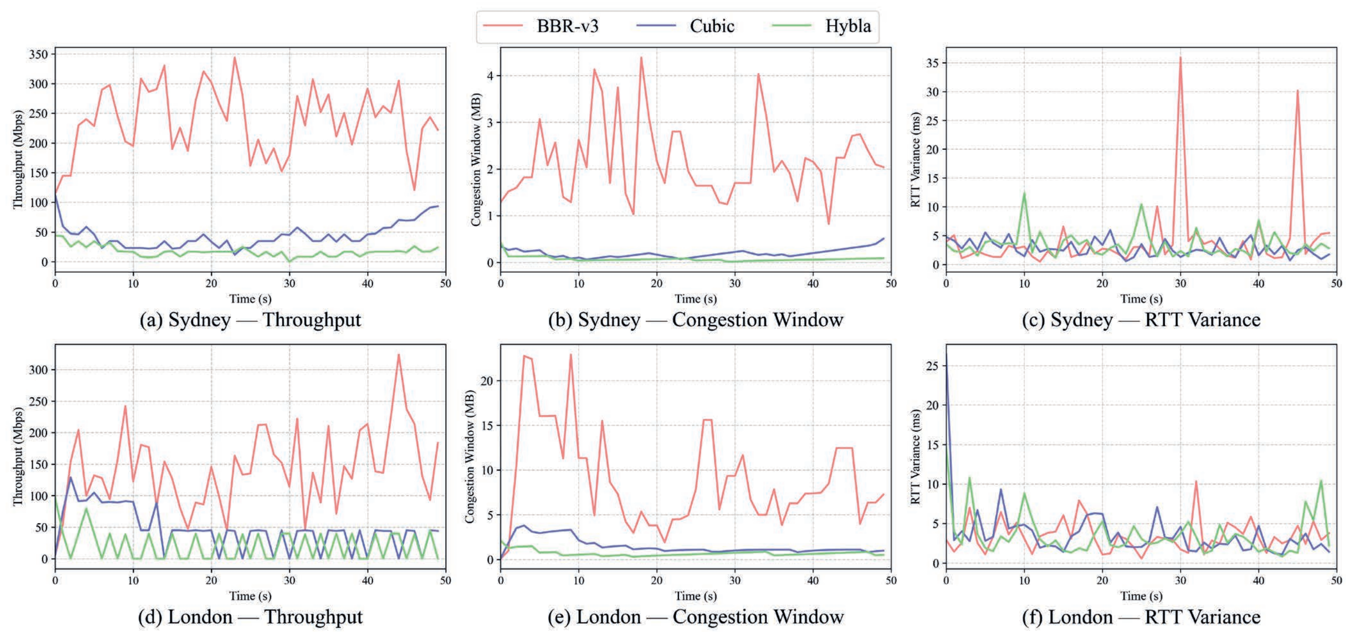


Fig. 5: Variation of throughput, congestion window, and RTT variance over time in downlink for Sydney and London.

dedicated Linux measurement server (Ubuntu 24.04 LTS) was connected to the Starlink router via a Category 5e Ethernet link, providing a consistent measurement environment.

We assume that the AWS cloud infrastructure offers stable and high-availability connectivity at each geographical site, thereby ensuring that observed performance variations predominantly arise from the Starlink access link and long-haul Internet paths rather than server-side dependencies. To evaluate the behavior of different CCAs, we employed `iperf3`, an open-source throughput measurement tool widely used for transport-layer performance analysis. Our automated measurement framework sequentially configured each remote–local pair for the selected CCA and executed tests across all endpoints. For every CCA, we conducted both uplink (forward) and downlink (reverse) measurements over the Starlink connection. Each experiment consisted of a 300-second `iperf3` session, during which transport-layer metrics were recorded, and the tests were conducted during the first week of August 2025.

## III. Results & Discussion

### A. Performance Evaluation of Uploading CCAs

*1) Observations-I:* Our experiment reveals that BBR consistently demonstrated superior throughput performance across all geographical locations tested. In several instances, BBR's upper throughput whiskers exceeded 80 Mbps, while alternative Cubic and Hybla CCAs maintained significantly lower throughput levels, as evidenced in Fig. 2(a), Fig. 3(a), and Fig. 3(d). BBR's congestion window exhibited markedly larger dimensions compared to other CCAs, with minimum second quartile values exceeding $0.4$ MB. This expanded window capacity enables BBR to fully utilize the BDP. Additionally, BBR displayed the highest variance in congestion window size across all tested locations, as illustrated in Fig. 2(c), Fig. 3(b), and Fig. 3(e).

However, BBR incurred substantially higher retransmission rates, with second quartile values ranging between 10 and 30, and maximum third quartile values approaching 70. This elevated retransmission activity was particularly pronounced in high-RTT locations. In contrast, Cubic and Hybla maintained comparatively minimal retransmission rates, as shown in Fig. 2(b). Regarding network latency, São Paulo consistently exhibited the highest RTTs with second quartile values exceeding 370 ms, while Sydney demonstrated the lowest RTTs with second quartile values approximating 50 ms. Notably, RTT variance remained relatively consistent across all tested CCAs, as depicted in Fig. 2(e), Fig.2(f), Fig. 3(c), and Fig. 3(f).

*2) Findings-I:* BBR's model-based design circumvents the RTT-dependence of traditional CCAs, making it highly effective in Starlink's dynamic LEO environment. This architectural edge enables consistently superior throughput under variable conditions. However, the gain comes with a trade-off: elevated *retransmission overhead*, reflecting BBR's tendency to push closer to maximum link capacity at the expense of stability. This performance stability compromise is critical for applications with strict reliability demands.

In contrast, Cubic and Hybla consistently underutilized Starlink's available capacity, underscoring the limitations of traditional loss- and delay-based CCAs in Starlink networks. The higher retransmission rates observed with BBR, despite similar and steady RTT variance across all CCAs, point to its *aggressive bandwidth probing* rather than satellite handover effects. Overall, BBR remains the most suitable option for uplink transport in Starlink, though adapting *retransmission mitigation and probing* strategies will be essential to strengthen robustness for reliability-critical Starlink applications.

### B. Performance Evaluation of Downloading CCAs

*1) Observations-II:* Our results show that BBR consistently delivered the highest downlink throughput across all test sites. In every location, the median throughput was above 100 Mbps, with peaks close to 350 Mbps, clearly outperforming the uplink, and other CCAs, as illustrated in Fig. 4(a), Fig. 5(a), and Fig. 5(d). This advantage came from BBR's much larger congestion window in the downlink, which allowed it to make better use of the available capacity compared to both its uplink behavior and other CCAs. See Fig. 4(c), Fig. 5(b), and Fig. 5(e) for details. However, this gain comes at a cost: retransmissions in the downlink were significantly higher, often above 300 packets, especially in Tokyo and Mumbai (Fig. 4(b)). RTT patterns were similar to the uplink. São Paulo (median RTT $>$ 350 ms) and London showed the highest delays, Sydney the lowest (around 50 ms), while overall RTT variation stayed modest at 2–4 ms. This result is illustrated in Fig. 4(e), Fig. 4(f), Fig. 5(c), and Fig. 5(f).

*2) Findings-II:* BBR unlocked Starlink's high downlink capacity, achieving far greater throughput than competing CCAs, but at the cost of heavy retransmissions. This reflects its *aggressive pacing*, which often overshot available bandwidth. In contrast, Cubic and Hybla remained capped below 60 Mbps, underscoring BBR's dominance. The sharp uplink–downlink contrast shows that while BBR maximizes performance, it suffers from *retransmission inflation* in Starlink downlink channels, highlighting the need for refined pacing under LEO dynamics, which requires further investigations.

## IV. Conclusions and Future Work

We designed and deployed a globally distributed testbed spanning multiple cities to benchmark BBR-v3 against Cubic and Hybla CCAs over SpaceX's Starlink network in both uplink and downlink. Results show that BBR consistently delivers substantially higher throughput, fully exploiting Starlink's network capacity while maintaining comparable RTT. This gain, however, comes with elevated retransmissions due to BBR's aggressive pacing. These findings demonstrate BBR's superiority in LEO satellite environments but also highlight the need for pacing refinements to balance throughput with stability. Future work will extend the testbed to assess fairness, resource utilization, and queue dynamics across Starlink deployments.


## Acknowledgement

This work is supported by SmartSat CRC, whose activities are funded by the Australian Government's CRC Program.


## References


[1] N. Cardwell, I. Swett, and J. Beshay, "BBR Congestion Control," IETF Congestion Control Working Group (CCWG), Internet-Draft draft-ietf-ccwg-bbr-03, Oct 2024, expires 19 April 2025. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-ietf-ccwg-bbr-03

[2] S. R. Pokhrel, J. Choi, and A. Walid, "Fair and efficient distributed edge learning with hybrid multipath TCP," *IEEE/ACM Transactions on Networking*, vol. 31, no. 4, pp. 1582–1594, 2023.

[3] A. Piotrowska, "Performance Evaluation of TCP BBRv3 in Networks with Multiple Round Trip Times," *Applied Sciences*, vol. 14, no. 12, p. 5053, 2024.

[4] Starlink, "Starlink Updates," Aug. 2025. [Online]. Available: https://www.starlink.com/updates/network-update

[5] N. Mohan *et al.*, "A Multifaceted Look at Starlink Performance," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 2723–2734.

[6] L. Izhikevich *et al.*, "A Global Perspective on the Past, Present, and Future of Video Streaming over Starlink," *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 8, no. 3, pp. 1–22, 2024.

[7] R. De Silva and S. Raj Pokhrel, "Adapting Large Language Models for Cognitive Spectrum Allocation in Coexisting GEO-LEO Satellites," in *2025 IEEE 50th Conference on Local Computer Networks (LCN)*, 2025, pp. 1–6.

[8] R. I. L. Goyzueta and Y. Chen, "A Deterministic Loss Model Based Analysis of CUBIC," in *2013 International Conference on Computing, Networking and Communications (ICNC)*. IEEE, 2013, pp. 944–949.

[9] C. Caini and R. Firrincieli, "TCP Hybla: a TCP Enhancement for Heterogeneous Networks," *International Journal of Satellite Communications and Networking*, vol. 22, no. 5, pp. 547–566, 2004.

[10] S. R. Pokhrel, M. Panda, and H. L. Vu, "Fair Coexistence of Regular and Multipath TCP over Wireless Last-Miles," *IEEE Transactions on Mobile Computing*, vol. 18, no. 3, pp. 574–587, 2019.