Congestion Control Working Group
Internet-Draft
Intended status: Informational
Expires: 1 September 2026

Z. Lai
Z. Li
Q. Wu
H. Li
Tsinghua University
Q. Zhang
Zhongguancun Laboratory
28 February 2026

Analysis for the Adverse Effects of LEO Mobility on Internet Congestion Control

draft-lai-ccwg-lsncc-01

Abstract

Low-earth-orbit (LEO) satellite networks (LSNs) are increasingly used to carry Internet traffic. However, the fast time-varying conditions induced by LEO mobility can significantly impair the performance of end-to-end Internet congestion control algorithms (CCAs). This document summarizes and analyzes how mobility-driven (often non-congestive) variations in capacity, delay, and loss can mislead representative classes of CCAs, using Starlink as an operational case study. In addition, this document highlights recently-validated measurement and design considerations that treat connection reconfiguration as a phase boundary observable at endpoints, which can improve the interpretation of network signals without requiring changes to the network infrastructure. This document also proposes a transport-agnostic "Path Phase Boundary" (PPB) abstraction and corresponding endpoint behavior considerations for safely using such hints across diverse CCAs. Finally, it outlines a reference endpoint probing profile that can be used to infer PPBs in a deployable, best-effort manner. The goal of this document is informational: to clarify the problem space and to inform future standardization and engineering efforts for congestion control over LSN paths.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Low-earth-orbit (LEO) satellite networks (LSNs) are evolving rapidly, enabled by the deployment of mega-constellations such as SpaceX Starlink, Eutelsat OneWeb, and Amazon LEO Project. These systems provide broadband connectivity with global coverage and are carrying an increasing amount of Internet traffic. For example, Starlink has been reported to serve more than ten million subscribers as of early 2026.

Internet congestion control algorithms (CCAs) are expected to perform robustly across diverse Internet paths, including paths containing satellite links. However, operational LSNs exhibit fast and frequent path changes due to LEO mobility and dynamic network control. Such changes can introduce abrupt variations in available capacity, baseline RTT, and packet loss characteristics. Importantly, many of these variations are not primarily caused by queue build-up at a persistent bottleneck, but by mobility-driven dynamics (e.g., handovers and connection reconfiguration). As a result, the end-to-end signals observed by a transport sender (loss, delay, and delivery rate) can become ambiguous: the sender cannot reliably distinguish congestion-induced signals from mobility-induced signals, and therefore may react inappropriately.

This document provides an informational analysis of the adverse effects of LEO mobility on representative classes of CCAs: loss-based (e.g., Reno[RFC5681]/Cubic[RFC9438]), delay-based (e.g., Vegas[vegas_cc]/Copa[copa_cc]), model-based (e.g., BBR[cardwell2016bbr]), and learning-based CCAs. We use Starlink as a case study to illustrate characteristic dynamics and the resulting performance pathologies.

Beyond problem characterization, this document summarizes recently-validated measurement and design considerations that can help interpret end-to-end observations over LSN paths[lai2025leocc]. In particular, we highlight connection reconfiguration as a phase boundary: when reconfiguration occurs, path conditions can shift abruptly, making historical estimates (e.g., bandwidth and minimum RTT filters) stale. Recent endpoint measurements show that reconfiguration can be detected in a timely manner using lightweight probing to stable network anchors (e.g., a point-of-presence address) and outlier detection on probe-response intervals. This observation supports an informational design principle: endpoints should avoid carrying over congestion-control state across reconfiguration boundaries, and should treat the path as operating in piecewise “phases” separated by reconfiguration events.

From a standardization perspective, this document focuses on the semantics and safe usage of a phase-boundary hint rather than on any single concrete signaling mechanism. We refer to such a hint as a Path Phase Boundary (PPB): an advisory indication that path conditions may have changed discontinuously such that previously-learned bandwidth and baseline-delay estimates may become stale. A PPB does not imply the presence or absence of congestion; instead, it informs how endpoints SHOULD scope estimation and re-convergence logic.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Characteristics of LEO Satellite Networks

```
                        Outgoing Satellite  Incoming Satellite
 high-speed movement       +-----------+   +-----------+   +-----------+
     related to    <--------| Satellite |---| Satellite |---| Satellite |---
the earth surface(~7.5km/s) +-----+-----+   +------+----+   +-----+-----+
                                      |  \  /  |
             frequent space-ground    |   \/   |
             handovers due to LEO     |   /\   |
              satellite mobility      |  /  \  |
                                      | /    \ |
 +----+----+    +---+---+    +----+----+    +---+---+    +--------+--------+
 |  User   |--->|  Home |--->|Satellite|    |Ground |--->|Point-of Presence|
 | Terminal|<---| Router|<---| Terminal|    |Station|<---|   and Internet  |
 +---------+    +---+---+    +----+----+    +---+---+    +--------+--------+
```

Figure 1: Emerging LSN architecture.

2.1. LSN Architecture

Figure 1 plots a typical architecture of emerging LSNs. Overall, the entire LSN can be divided into a space segment which consists of a considerable number of LEO satellites, together with a terrestrial segment that contains many geo-distributed ground stations around the world. On the user side, to access the LSN, a user needs to purchase and deploy a dedicated satellite terminal (i.e. a dish), together with a home router which connects to the user’s terminal (e.g. a smartphone or a laptop) via a WiFi or Ethernet interface. On the ground station side, the LSN exchanges traffic with the terrestrial Internet through a set of geographically distributed gateways behind ground stations. When the LSN provides Internet services for terrestrial users, user traffic is forwarded to LEO satellites via the satellite terminal, then to a ground station, and finally to the gateway and terrestrial Internet (and vice versa). If the user is close to an available ground station, satellites can use the well-known "bent-pipe" routing mechanism to transparently forward user traffic to the corresponding ground station. Otherwise, for users in remote areas far away from available ground stations, the LSN can exploit inter-satellite links (ISLs) to route user traffic to the ground station.

2.2. Low Orbit Altitude, Low Latency and High Throughput

Low latency in LSNs results from their proximity to earth, minimizing the ground-to-satellite distance data packets must travel. This allows for quicker response time, essential for applications like gaming and video conferencing. High throughput is achieved through the use of high-speed Ka-/Ku-band spectrum, and the use of multiple satellites working together in a mesh network, enabling large amounts of data to be transmitted simultaneously. This combination of low latency and high throughput positions LEO networks as a competitive alternative to traditional broadband options, especially in remote areas.

2.3. LEO Mobility

Unlike traditional geostationary (GEO) satellite networks, one key property of emerging LSNs is that: a portion of the network infrastructure, i.e. LEO satellite routers, are moving at a high velocity related to the earth. Such unique LEO mobility can introduce abrupt, non-congestive variations in (i) available capacity, (ii) baseline RTT (due to geometry changes and routing changes), and (iii) packet loss (due to transient outages, link-layer behavior, and handover disruption).

A particularly salient mechanism in operational LSNs is connection reconfiguration. Reconfiguration events can act as phase boundaries: in this document, we refer to these boundaries as Path Phase Boundaries (PPBs). within a phase (between consecutive reconfigurations), path conditions may be relatively stable but noisy; across a boundary, key path parameters can change abruptly. This piecewise structure is critical when interpreting transport-level observations, especially for CCAs that rely on windowed extrema filters (e.g., max bandwidth or min RTT over a history window). Endpoints can sometimes observe mobility-driven phase boundaries without any explicit signal from the network. Recent measurements suggest that reconfiguration can induce short disruptions (e.g., transient outages) that manifest as outliers in probe-response timing. Such observations motivate measurement and control designs that are reconfiguration-aware while remaining purely end-to-end.

3. Impacts of LEO Mobility on Internet Congestion Control

3.1. Principles of Today’s Internet Congestion Control

Internet congestion control is vital for maintaining network performance and stability. Typically, congestion control uses feedback loops to manage data flow. The sender adjusts its sending rate based on time-varying network conditions, ensuring efficient use of available bandwidth without overwhelming the network. In particular, congestion control algorithms (CCAs) detect network congestion through monitoring the performance changes observed on the sender, based on certain indicators such as packet loss, increased latency. Based on the different principles used for congestion detection and rate adaptation, existing CCAs generally can be classified by the following categories.

3.1.1. Loss-based CCAs

Loss-based CCAs, such as TCP Reno [RFC5681] and Cubic [RFC9438], primarily detect packet loss as a signal of network congestion. Loss-based CCAs identify congestion through packet loss, often using acknowledgments (ACKs) from the receiver. If packets are not acknowledged within a certain timeframe, the sender infers congestion. Upon detecting packet loss, loss-based CCAs reduce the congestion window size, which limits the amount of unacknowledged data in transit, thus decreasing the sending rate.

3.1.2. Delay-based CCAs

Delay-based CCAs such as Vegas [vegas_cc] and Copa [copa_cc] focus on monitoring changes in network delay as a signal of congestion, rather than relying solely on packet loss. In delay-based CCAs, the sender continuously measures round-trip time (RTT) to detect increases in latency, which can indicate congestion before packet loss occurs. When increased delay is detected, the sender adjusts its transmission rate, often by decreasing the congestion window, to alleviate potential congestion. Delay-based CCAs aim to react to congestion signals proactively, adjusting data rates to prevent packet loss rather than responding to it after the fact.

3.1.3. Model-based CCAs

Model-based CCAs employ mathematical and statistical models to predict network behavior and optimize data transmission. They often begin with modeling the network’s dynamics, including bandwidth, delay, and packet loss, to understand how traffic interacts with these factors. Continuous data collection on network performance helps update the model, allowing for dynamic adjustments based on current conditions. Instead of reacting solely to congestion signals like packet loss, model-based control predicts congestion based on trends in delay and other metrics, allowing for proactive rate adjustments. For example, Google’s BBR[cardwell2016bbr] models the bottleneck bandwidth and round-trip time to estimate the optimal sending rate. At runtime, BBR continuously monitors the conditions of the network path, adjusting its model based on real-time measurements of bandwidth and delay. By maintaining the sending rate close to the estimated bandwidth without causing excessive queuing delays, BBR optimizes throughput while minimizing latency.

3.1.4. Learning-based CCAs

Recent learning-based CCAs such as PCC-VIVACE[dong2018pcc] leverages machine learning techniques to optimize data transmission by predicting and adapting to network conditions. Instead of relying solely on predefined algorithms, learning-based CCAs analyze historical network data to identify patterns and make informed decisions. Relevant features include packet loss, throughput, and round-trip time (RTT) extracted from network measurements to serve as input for machine learning models. The trained model can adjust the sending rate in real time based on current network conditions, continuously learning and refining its predictions as it receives new data.

3.2. LEO Mobility Breaks Fundamental Assumptions of Congestion Control

LEO mobility challenges congestion control by breaking two implicit assumptions commonly relied upon for end-to-end inference: stationarity over estimation windows and congestion-dominated signal semantics. For the former, many CCAs assume that key path parameters (bottleneck bandwidth, baseline RTT) are approximately stationary over the timescale of their filters or control loops. In operational LSNs, connection reconfiguration can abruptly change these parameters, making history samples stale across a boundary. For the latter, classic interpretations assume that increases in RTT or the occurrence of loss primarily reflect congestion-induced queueing or dropping. In LSNs, delay spikes, loss bursts, and throughput drops can be dominated by mobility-induced effects (e.g., transient disruptions during reconfiguration), and therefore the semantics of these signals become ambiguous. These assumption breaks explain why CCAs may either (i) become overly conservative (underutilizing available capacity) due to frequent non-congestive loss/delay events, or (ii) become overly aggressive (creating persistent queues) due to biased bandwidth/baseline-delay estimation. An informational design implication is that endpoints SHOULD treat reconfiguration events as phase boundaries and SHOULD avoid combining measurements across boundaries when maintaining extrema-based estimates (e.g., maximum delivery rate or minimum RTT).

4. Measurements in Starlink

4.1. CCA Performance in Starlink

To quantitatively understand the performance of different CCAs in real-world operational LSNs, we conduct a performance study of several kinds of representative CCAs based on an operational LSN. Specifically, we evaluate: (i) loss-based CCAs, Reno [RFC5681] and Cubic[RFC9438] which use packet losses as the signal for adjusting data sending rate; (ii) delay-based CCAs, Vegas[vegas_cc] and Copa[copa_cc], which exploit measured delay to estimate network congestion and adjust sending rate; (iii) model-based CCAs, Google BBRv1/v3[I-D.ietf-ccwg-bbr], which frequently measure the bottleneck bandwidth and minimal RTT to model the bandwidth-delay product (BDP) of the path, and accordingly regulates sending rate; and (iv) learning-based CCA, PCC-VIVACE[dong2018pcc], which can automatically adapt itself to various conditions based on a utility function without manually tuning. We describe our case-by-case observations and corresponding analysis as follows.

Algorithm	Average Throughput (Mbps)	Average RTT (ms)	90th RTT (ms)	95th RTT (ms)
Reno	10.89	26.81	30.08	31.89
Cubic	10.56	27.27	30.90	32.77
Vegas	4.53	28.32	31.77	33.31
Copa	6.85	39.87	43.46	44.71
BBRv1	22.79	47.90	73.02	89.79
BBRv3	16.52	26.13	35.35	48.29
PCC-Vivace	17.15	97.08	171.33	207.26

Table 1

Reno and Cubic. End-to-end connections experience non-congestion packet losses over the unstable LEO satellite links. It is a well-known limitation that Cubic and Reno can not discriminate such non-congestion packet losses. As a result, TCP Reno and Cubic mistakenly think network is congested and shrink their congestion window conservatively when non-congestion packet losses occur, causing self-limited throughput.

Vegas and Copa. Delay-based CCAs rely on a basic assumption that the increase in RTT observed by the sender may reflect queuing at the bottleneck link. However, delay-based CCAs can be seriously misled in LSNs because it is difficult for them to distinguish whether the observed RTT changes are caused by congestive queuing or by path fluctuations due to LEO mobility. Specifically, Vegas detects congestion by increasing RTTs, and we observe that Vegas is frequently misled by these non-congestion RTT increases in LSNs, resulting in severe throughput degradation. Similarly, Copa is a recent delay-based CCA that converges on a target sending rate 1/(sigma * Dq) where Dq is the measured queuing delay and sigma is a constant. Copa adjusts the congestion window in the direction of this target rate, and estimates the queuing delay as Dq = RTTstanding - RTTmin, where RTTstanding is the smallest RTT observed over a recent short time-window and RTTmin is the smallest RTT observed over a long period of time (e.g. 10 seconds). We find that as the environmental RTT fluctuates frequently and drastically, Copa usually overestimates Dq and then limits its sending rate. When the environmental RTT suddenly increases to a new level, although Copa’s

RTTstanding estimation can be updated in time, it still takes a long time for Copa’s RTTmin estimation to converge to the correct value. Therefore, as the environmental RTT changes drastically, Copa frequently underestimates RTTmin, and then overestimates Dq which is calculated by RTTstanding - RTTmin. As a result, Copa mistakenly infers that there is congestion in the network and limits its sending rate, and achieves low link utilization and self-limited throughput.

BBR frequently probes the network for its propagation RTT (pRTT) and bottleneck bandwidth (bBW), and then adjusts sending rate to match the bandwidth-delay product (BDP). We identify several issues in different versions of BBR.

BBRv1 experiences bBW overestimation and pRTT underestimation under the drastic network variations caused by LEO mobility. First, BBRv1 estimates bBW by the maximum delivery rate (deliveryRate) over a 10-RTT window. When the link capacity fluctuates drastically, such a maximum filter always over-estimates bBW. Note that BBRv1’s sending rate is set by the estimated bBW multiplied by a factor called pacing_gain and the data in flight is capped by cwnd=2 * BDP. When the link capacity fluctuates, because bBW is overestimated, BBRv1 overshoots the link capacity until the data in flight reaches 2 * BDP, resulting in high queuing delay especially when the link capacity significantly slumps. Second, BBRv1 estimates pRTT by the minimum observed RTT over a 10-second window. Thus, when the RTT increases due to LEO mobility rather than congestion, BBRv1 under-estimates pRTT. However, because bBW is overestimated most of the time, while pRTT is underestimated much less often, in our experiments we observe that in most cases the BDP is still overestimated.

BBRv3 has made several modifications upon BBRv1. One important aspect is that BBRv3 estimates bBW as the minimum value of two new parameters bw_high and bw_low. Specifically, bw_high is calculated by the maximum delivery rate over a short window, while bw_low is set to an extremely high value when there is no packet loss, but is set to max(latest_deliveryRate, 0.7*bw_high) if packet loss >0. In other words, BBRv3 suppresses the sending rate in case of packet loss. The original intention of this change is that when packet loss occurs, it may indicate congestion, so BBRv3 should reduce the sending rate. However, in our experiments, we observe that due to random packet losses in LEO satellite links, BBRv3 avoids overshooting the link but is less resilient to non-congestion loss as compared to BBRv1. As a result, BBRv3 can only achieve about 60-70% link utilization under lossy Starlink condition.

PCC-VIVACE. Recent CCAs like VIVACE try to learn from the observed network conditions based on a utility function, and accordingly estimate a proper sending rate. Specifically, VIVACE’s utility function is calculated based on the sending rate contribution, latency penalty (calculated by RTT gradient) and loss penalty in each measurement interval. We observe two performance issues in VIVACE. First, non-congestion RTT increase caused by LEO mobility can amplify latency penalty and result in inaccurate utility estimation. Second, VIVACE incorporates a dynamic change boundary omega to limit the rate change in a certain range. The original intention of omega is to avoid drastic rate change that overshoots the link capacity, but such a boundary also leads to slow rate convergence in Starlink as the link capacity changes rapidly. As a result, VIVACE: (i) under-utilizes the network when the link capacity drastically increases or when the propagation RTT suddenly increases due to LEO mobility; and (ii) overshoots the network when the link capacity drastically decreases, causing high queuing delay.

Based on our analysis we find that it is quite challenging for existing CCAs to detect network congestion promptly and accurately in an LSN with drastic, multi-dimensional network variations induced by LEO mobility. Essentially, every CCA relies on certain network models and assumptions, based on which the CCA infers network conditions and whether congestion occurs. However, these fundamental assumptions they used become inaccurate in emerging LSNs. Link capacity, RTT and loss rate can change frequently and drastically in LSNs, mixing both congestion and non-congestion variations, and existing CCAs can easily be misled by these non-congestion signals. Considering the fundamental challenge is that it is hard for end-to-end CCAs to discriminate whether the observed performance changes are caused by congestion or not, we argue that CCAs in LSNs require some effective indicators which can implicitly help end host discriminate non-congestion performance changes.

4.2. Connection Reconfiguration as an Observable Phase Boundary

Motivated by the above observation that CCAs are often misled by mobility-induced, non-congestive variations, we next characterize connection reconfiguration as an endpoint-observable phase boundary. To isolate LSN-induced dynamics from variations in the wider Internet, one useful approach is to probe a stable point-of-presence (PoP) address that is consistently reachable from the terminal; probing to such an anchor largely confines measurement to the LSN access segment and improves attribution.

Using lightweight periodic probes (e.g., ICMP echo requests) to this anchor, recent measurements indicate that connection reconfiguration can manifest as short transient disruptions (e.g., brief outages) and abrupt shifts in baseline RTT and available capacity. Reconfiguration boundaries can be detected in a timely manner by analyzing the distribution of consecutive probe-response intervals: while RTT spikes may be misled when probes are lost, outliers in the response interval can still reveal disruption events, providing an endpoint-observable indicator of phase boundaries. Such boundaries imply that estimates aggregated across reconfiguration events (e.g., bandwidth or minimum-RTT filters) can become stale immediately after reconfiguration, which helps explain the performance pathologies observed above.

5. Potential Mitigations

In addition to the broad directions discussed below, a key takeaway is that LSN dynamics often exhibit phase-boundary behavior. The PPB abstraction captures this behavior at the transport layer: it is a best-effort hint that path conditions may have changed discontinuously, and therefore endpoints SHOULD avoid carrying over certain extrema-based estimates (e.g., bandwidth and minimum RTT) across the boundary unless validated. This document does not mandate a specific mechanism to obtain PPBs; endpoints may infer them and networks may explicitly signal them.

5.1. Explicit notifications for network variance discrimination

Explicit signals from the network can, in principle, help endpoints distinguish mobility-induced variations from congestion-induced signals. Examples include explicit reconfiguration markers, path-change notifications, or segment-specific congestion feedback. However, standardizing such signals requires careful consideration of deployment feasibility, security, privacy, and interoperability across administrative domains. In addition, the signal semantics must be robust: false positives or delayed notifications may worsen performance. In the absence of explicit notifications, endpoints may still infer likely phase boundaries using end-to-end observation. Standardization efforts may explore how explicit and implicit signals can complement each other.

5.2. Cross-layer optimization

In conventional networks like WiFi and cellular networks in which the link capacity may rapidly change over time, one classic optimization method is to directly use the underlying channel information to estimate bandwidth changes more accurately and timely, thereby improving the performance of end-to-end congestion control. For example, ExLL [park2018exll] is a congestion control for LTE networks and it exploits cellular bandwidth inference to achieve low latency. CQIC [lu2015cqic], CLAW [xie2017accelerating] and piStream [xie2015pistream] use PHY-layer information to accurately and timely estimate traditional cellular and WiFi networks. PropRate [leong2017tcp] adjusts sending rate by directly monitoring the bottleneck buffer size in cellular networks. PBE-CC [xie2020pbe] leverages PHY-layer measurements to precisely react to capacity variations. The similar cross-layer optimizations might also be available if the satellite network operators expose sufficient programmable interfaces for system and application developers.

5.3. Multipath enhancement

Multi-path transmission can also significantly enhance CCAs in LSNs by improving reliability, throughput, and resilience. By utilizing multiple LSN paths or an LSN path and a cellular path simultaneously, networks can aggregate bandwidth, leading to higher overall data rates and improved performance for users. If one path encounters issues (packet loss), data can still be transmitted via alternative paths, enhancing overall network reliability.

6. Conclusion

Operational LEO satellite networks introduce fast and structured dynamics that can severely impair end-to-end congestion control. Mobility-driven changes in capacity, baseline RTT, and loss can break key assumptions behind congestion inference, leading to both underutilization and excessive queueing. This document provided an informational analysis of these adverse effects and used Starlink as a case study. We also highlighted recently-validated measurement and design considerations that treat connection reconfiguration as a phase boundary observable at endpoints. This suggests a standardization opportunity around the semantics and safe usage of phase-boundary hints (PPBs), independent of how such hints are obtained or signaled. Such considerations can inform future research, engineering, and potential standardization efforts for congestion control over LSN paths.

7. IANA Considerations

This document includes no request to IANA.

8. Security Considerations

This document does not represent a change to any aspect of the TCP/IP protocol suite and therefore does not directly impact Internet security.

9. References

9.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC5681] Allman, M., Paxson, V., and E. Blanton, "TCP Congestion Control", RFC 5681, DOI 10.17487/RFC5681, September 2009, <https://www.rfc-editor.org/rfc/rfc5681>.

[RFC9438] Xu, L., Ha, S., Rhee, I., Goel, V., and L. Eggert, Ed., "CUBIC for Fast and Long-Distance Networks", RFC 9438, DOI 10.17487/RFC9438, August 2023, <https://www.rfc-editor.org/rfc/rfc9438>.

9.2. Informative References

[I-D.ietf-ccwg-bbr] Cardwell, N., Swett, I., and J. Beshay, "BBR Congestion Control", Work in Progress, Internet-Draft, draft-ietf-ccwg-bbr-04, 20 October 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-ccwg-bbr-04>.

[copa_cc] Arun, V. and H. Balakrishnan, "Copa: Practical Delay-Based Congestion Control for the Internet", 2018.

[vegas_cc] Brakmo, L. S, O’malley, S. W, and L. L Peterson, "TCP Vegas: New techniques for congestion detection and avoidance", 1994.

[cardwell2016bbr] Cardwell, N., Cheng, Y., Gunn, C S., Yeganeh, S. H., and V. Jacobson, "BBR: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time", 2016.

[dong2018pcc] Dong, M., Meng, T., Zarchy, D., Arslan, E., Gilad, Y., Godfrey, B., and M. Schapira, "PCC Vivace:Online-Learning Congestion Control", 2018.

[park2018exll]
Park, S., Lee, J., Kim, J., Lee, J., Ha, S., and K. Lee, "ExLL: An extremely low-latency congestion control for mobile cellular networks", 2018.

[lu2015cqic]
Lu, F., Du, H., Jain, A., Voelker, G. M, Snoeren, A. C, and A. Terzis, "CQIC: Revisiting cross-layer congestion control for cellular networks", 2015.

[xie2017accelerating]
Xie, X., Zhang, X., and S. Zhu, "Accelerating mobile web loading using cellular link information", 2017.

[xie2015pistream]
Xie, X., Zhang, X., Kumar, S., and L. E. Li, "piStream: Physical layer informed adaptive video streaming over LTE", 2015.

[leong2017tcp]
Leong, W. K., Wang, Z., and B. Leong, "TCP congestion control beyond bandwidth-delay product for mobile cellular networks", 2017.

[xie2020pbe]
Xie, Y., Yi, F., and K. Jamieson, "PBE-CC: Congestion control via endpoint-centric, physical-layer bandwidth measurements", 2020.

[lai2025leocc]
Lai, Z., Li, Z., Wu, Q., Li, H., Li, J., Xie, X., Li, Y., Liu, J., and J. Wu, "LeoCC: Making Internet Congestion Control Robust to LEO Satellite Dynamics", 2025.

Acknowledgements

Contributors

Thanks to all of the contributors.

Authors' Addresses

Zeqi Lai
Tsinghua University
30 ShuangQing Ave
Beijing
100089
China

Email: zeqilai@tsinghua.edu.cn

Zonglun Li
Tsinghua University
30 ShuangQing Ave
Beijing
100089
China
Email: lzl24@mails.tsinghua.edu.cn

Qian Wu
Tsinghua University
30 ShuangQing Ave
Beijing
100089
China
Email: wuqian@cernet.edu.cn

Hewu Li
Tsinghua University
30 ShuangQing Ave
Beijing
100089
China
Email: lihewu@cernet.edu.cn

Qi Zhang
Zhongguancun Laboratory
Beijing
China
Email: zhangqi@zgclab.edu.cn