10.48047/jocaaa.2024.32.01.70

TCP Congestion Control: A Paradigm Shift from Reactive Loss Signals to Model-Driven and Learning-Based Control

[1]Raj Kumar
Assistant Professor, Department: Computer Science and Engineering, Government Engineering College, Vaishali, Bihar, India
Email Id: nit.er.raj@gmail.com

[2]Manoj Kumar Sah
Assistant Professor, Department: Computer Science and Engineering, Government Engineering College, Vaishali, Bihar, India
Email Id: emanojcse@gmail.com,

[3]Md Ehtashamoul Haque
Assistant Professor, Department: Computer Science and Engineering, B. P. Mandal College of Engineering, Madhepura, Bihar, India
Email Id: ehtasham47@gmail.com

Received 20-10-2023, Revised 15-11-2023, Accepted 21-01-2024

Abstract - The Transmission Control Protocol (TCP) has been central to the stability and scalability of the Internet, with its congestion control mechanism playing a pivotal role. The early loss-based framework, epitomized by TCP Reno’s Additive Increase Multiplicative Decrease (AIMD) strategy, relied on packet loss as the primary congestion signal. However, the emergence of high bandwidth–delay product paths, deep buffers, and wireless links has exposed its scalability and latency limitations. This paper traces the evolution of TCP congestion control across major paradigms developed to address these challenges. We examine high-speed loss-based algorithms such as CUBIC, proactive delay-based, and model-based schemes, including TCP Vegas and BBR, and the recent shift toward online learning with Performance-oriented, Congestion Control (PCC). Through comparative analysis of these representative approaches, the paper highlights the transition from reactive, loss-driven mechanisms to intelligent, measurement-driven, and adaptive systems that align with the demands of modern networks.

Index Terms—TCP congestion control, loss-based control, delay-based control, model-based control, learning-based control, TCP CUBIC, BBR, PCC, High-speed networks, Bufferbloat.

1. Introduction

The Transmission Control Protocol (TCP) has served as the backbone of the Internet for decades, ensuring reliable data delivery across a vast and heterogeneous global network. A critical component of TCP's long-lasting success is its congestion control mechanism, a set of algorithms designed to prevent network collapse by adapting a sender's transmission rate to the available capacity of the overall network path. The initial TCP congestion control algorithms, standardized in [1], were built on a foundational principle: packet loss is a reliable indicator of congestion. This loss-based approach, embodied in the Additive Increase, Multiplicative Decrease (AIMD) strategy of TCP Reno [2], was instrumental in stabilizing the early Internet.

10.48047/jocaaa.2024.32.01.70

However, the exponential growth in network capacities and changing application requirements has exposed fundamental limitations of traditional loss-based approaches. As bandwidth-delay products have increased, especially in long-distance and high-speed environments, standard TCP algorithms have struggled to utilize link capacities and maintain low latency fully. For instance, standard TCP's additive increase and multiplicative decrease (AIMD) strategy results in slow convergence to optimal throughput on high-bandwidth, high-latency links, and can be excessively conservative in the presence of non-congestion-related losses. This limitation led to the development of high-speed TCP variants like HighSpeed TCP [3] and, most notably, CUBIC, which modified the AIMD response function to be more aggressive in high Bandwidth-Delay Product (BDP) environments. CUBIC [4], now the default in major operating systems, represents the pinnacle of loss-based congestion control evolution.

Simultaneously, a deeper understanding of network dynamics revealed that the reliance on packet loss itself was a fundamental problem. In networks with deep buffers, loss-based algorithms tend to keep queues persistently full, a phenomenon known as "bufferbloat," leading to high latency and poor user experience. Furthermore, in wireless and other lossy networks, non-congestion-related packet loss can trick these algorithms into unnecessarily reducing their sending rate.

These shortcomings have catalyzed a paradigm shift in congestion control research, moving away from reactive, loss-based signals towards more proactive and model-based approaches.

This paper provides a systematic and structured overview of the evolution of TCP congestion control mechanisms, charting their progression from foundational algorithms to contemporary state-of-the-art approaches. To facilitate this analysis, we classify existing congestion control schemes into four representative categories that serve as key milestones in this evolution.

(i) Traditional loss-based control (Reno [2], NewReno [5]): These algorithms embody the original AIMD framework, in which packet loss serves as the primary congestion signal.

(ii) High-speed loss-based control (HighSpeed TCP [3], CUBIC [4]): These schemes extend AIMD to improve scalability in high-bandwidth, high-delay product (BDP) networks.

(iii) Delay and model-based control (TCP Vegas [6], BBR [7]): These approaches leverage queuing delay measurements or explicit network models to detect and mitigate congestion proactively, often before packet loss occurs.

(iv) Online learning-based control (PCC [8]): An emerging class that adopts empirical, performance-oriented strategies.

Through a detailed analysis of these representative algorithms, this paper seeks to present a comprehensive perspective on the continuing research toward designing congestion control protocols capable of delivering consistently high performance across the modern Internet. The paper is structured as follows: Section II provides an overview of the foundational loss-

10.48047/jocaaa.2024.32.01.70

based mechanisms. Sections III, IV, and V explore the evolution toward high-speed, model-based, and learning-based paradigms, respectively. Section VI reviews explicit feedback mechanisms for wireless networks. Section VII presents a comparative analysis, and Section VIII concludes the paper.

2. Background: The Foundational Loss-Based Framework

The foundational framework for modern TCP congestion control was established to address the threat of network collapse by reacting to congestion signals from the network. As standardized in RFC 5681 [1], this framework is predicated on the core assumption that packet loss is a reliable indicator of network congestion. It comprises four intertwined algorithms that govern the sender's behavior: Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery. Collectively, these algorithms implement an Additive Increase, Multiplicative Decrease (AIMD) strategy. Initially, a connection probes for available bandwidth aggressively during the Slow Start phase, doubling its congestion window (cwnd) approximately every Round-Trip Time (RTT). Upon reaching a slow start threshold, it transitions to a more conservative Congestion Avoidance phase, where the cwnd is increased linearly by approximately one segment per RTT.

The detection of congestion, inferred from a retransmission timeout or duplicate acknowledgments (Fast Retransmit), triggers a multiplicative decrease in the sending rate. A significant refinement to this model is the NewReno modification (RFC 6582) [5]. Standard Reno's Fast Recovery algorithm was often inefficient in scenarios with multiple packet losses. NewReno addresses this deficiency by introducing a more robust recovery mechanism that interprets a "partial acknowledgment" as an indication of a subsequent packet loss, allowing it to repair multiple losses without resorting to a timeout. This Reno/NewReno framework, which directly couples the event of packet loss to a predetermined control response, represents the classic loss-based paradigm. The characteristic "saw-tooth" behavior of the TCP Reno congestion window is illustrated in Figure 1.

10.48047/jocaaa.2024.32.01.70

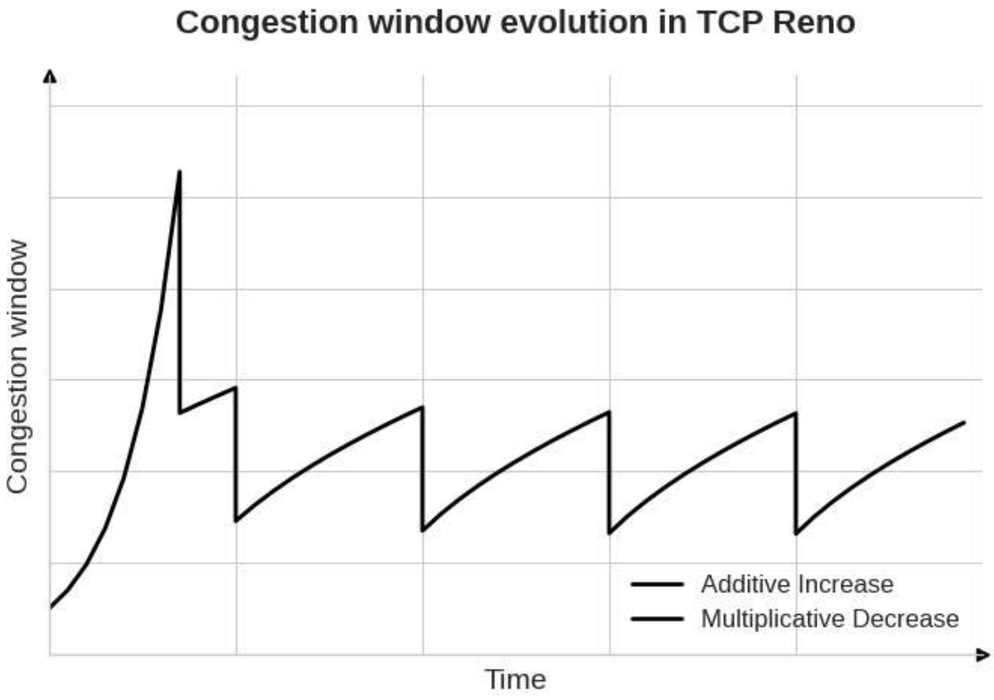

Figure 1:Saw-tooth behavior of the TCP Reno congestion window

3. High-Speed, Loss-Based Algorithms

The advent of "long fat networks" (LFNs)—networks with high bandwidth-delay products (BDP) exposed a fundamental scalability limitation in the traditional TCP AIMD algorithm. The conservative linear increase of one segment per RTT, as specified in [1], proved to be insufficient to fully utilize network capacity in these environments, as recovering the congestion window after a loss event could take an impractically long time. As noted in [3], a standard TCP flow on a 10 Gbps link would require an unrealistically low packet drop rate to maintain its congestion window, motivating a new class of "high-speed" TCP variants. These algorithms retain the core loss-based philosophy but modify the response function to be more aggressive at larger window sizes.

An early proposal in this space was HighSpeed TCP (RFC 3649) [3], which defined a response function that employs a more aggressive additive increase and a less severe multiplicative decrease as the window grows. Building upon these principles, CUBIC (RFC 9438) [4] emerged as a more refined and widely deployed solution. CUBIC's primary innovation is the replacement of the linear window growth function with a cubic function of the time elapsed since the last congestion event. As shown in Figure 2, this design results in a window growth that is initially rapid and concave, slows to a stable plateau around the previous congestion point (W_max), and then becomes convex to probe for new available bandwidth. A key design principle is that this time-based growth function makes the window evolution independent of RTT, thereby providing significantly better RTT-fairness than standard TCP.

10.48047/jocaaa.2024.32.01.70

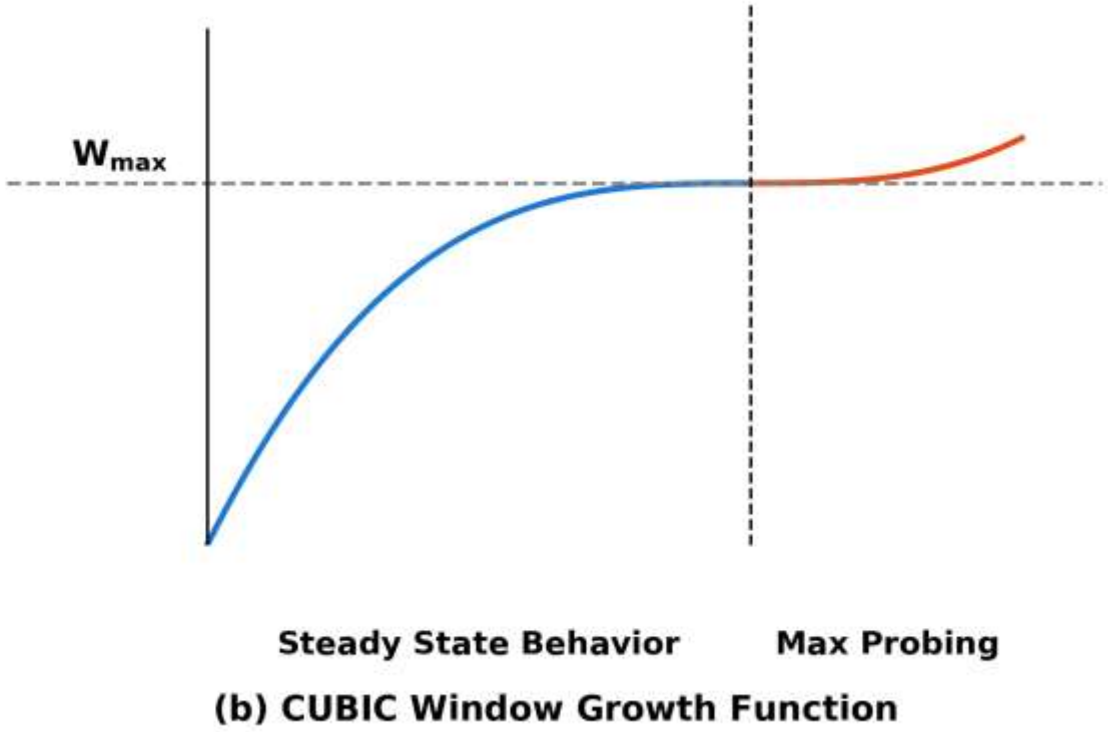

Figure 2: The window growth function of CUBIC.

4. Delay-Based and Model-Based Paradigms

A fundamental limitation of the loss-based paradigm is its reactive nature, which relies on filling network buffers to the point of overflow to detect congestion. This design inherently contributes to the "bufferbloat" problem, inducing high and variable queuing delays that degrade the performance of interactive applications. In response, a new class of algorithms emerged that sought to proactively manage congestion by using queuing delay as a primary signal. A pioneering effort in this domain was TCP Vegas [6], which proposed a novel congestion avoidance mechanism. As detailed by Brakmo and Peterson, Vegas continuously compares the expected throughput (calculated from the minimum observed RTT) with the actual measured throughput. By adjusting its congestion window to keep the difference—a proxy for the number of packets queued in the network—within a small, predefined range, Vegas attempts to operate with high throughput while maintaining low queuing delays and avoiding packet loss altogether.

Building on this proactive philosophy, Google's BBR (Bottleneck Bandwidth and Round-trip propagation time) [7] represents a more comprehensive, model-based approach. The authors of BBR argue that congestion control should not react to intermediary signals like loss or delay, but should instead be governed by an explicit model of the network path's two fundamental physical constraints: the bottleneck bandwidth (BtlBw) and the round-trip propagation time (RTprop). BBR continuously estimates these two parameters. It then paces its data transmission to match the measured BtlBw, ensuring the amount of data in flight is just enough to fill the bandwidth-delay product (BDP) pipe without creating a persistent

10.48047/jocaaa.2024.32.01.70

queue. By explicitly modeling the network and operating at what Kleinrock identified as the optimal point, BBR is designed to achieve both high throughput and minimal latency. This model-driven approach also makes it exceptionally resilient to non-congestion-related packet loss, as it does not use loss as a signal to reduce its rate, thereby overcoming a major deficiency of all loss-based algorithms.

5. A New Architecture: Online Learning-Based Control

A more recent and radical departure from traditional congestion control design is the online learning-based paradigm, exemplified by Performance-oriented Congestion Control (PCC) [8]. The central argument put forth by Dong et al. is that all preceding algorithms, whether loss-based or delay-based, are fundamentally limited by a "hardwired mapping" architecture. This architecture predefines a control response (e.g., halve the window) to a specific network event (e.g., a packet loss) based on a fixed set of assumptions about the underlying cause of that event. When these assumptions are violated by the complex and diverse conditions of real-world networks—such as the presence of non-congestion loss or rapidly changing link characteristics—performance degrades significantly.

PCC re-architects this process by treating congestion control as an online learning problem, entirely abandoning fixed assumptions. The sender continuously conducts "micro-experiments" by sending at a given rate for a short monitoring interval and observing the resulting performance. This performance is quantified by a utility function, which aggregates metrics like throughput, loss rate, and latency into a single numerical score. By comparing the utility scores of slightly different sending rates, the sender empirically discovers which control actions lead to better performance and adjusts its rate in that direction. This approach, which is fundamentally a form of real-time A/B testing for rate control, allows PCC to learn the optimal sending rate for a vast range of network conditions without any preconceived model of the network's behavior. By directly optimizing for an explicit performance objective, PCC can achieve consistently high performance in environments where assumption-based algorithms struggle or fail.

6. A Specialized Approach: Explicit Feedback for Wireless Networks

While end-to-end congestion control algorithms have evolved to be highly sophisticated, their performance can still be suboptimal in environments with rapid, large-scale capacity variations, such as wireless mobile networks. The fundamental challenge for any end-to-end scheme is the inherent delay and ambiguity in inferring the state of the network from signals like packet loss or RTT changes. An alternative paradigm, which sacrifices the end-to-end principle for higher performance, involves the use of explicit feedback from network routers. Accel-Brake Control (ABC) [11] is a recent protocol designed specifically for this purpose in time-varying wireless links.

As proposed by Goyal et al., ABC introduces a simple yet powerful mechanism where the wireless router, which has direct knowledge of the current link rate, provides explicit per-packet feedback to the sender. This feedback is a single bit, representing either an

10.48047/jocaaa.2024.32.01.70

"accelerate" or "brake" command, which is embedded into the existing ECN field, making it incrementally deployable. Upon receiving this feedback via an ACK, the sender makes a small, immediate adjustment to its congestion window. By aggregating these small adjustments over an RTT, the router can precisely and rapidly guide the sender's rate to match the dynamically changing link capacity. This approach avoids the blind probing of end-to-end algorithms, allowing the system to track the wireless link rate with high fidelity, thereby achieving both high throughput and low queuing delay in a challenging environment where traditional end-to-end signals are often misleading.

7. Comparative Analysis and Future Directions

The evolution of TCP congestion control, from the foundational principles of Reno to the learning-based architecture of PCC [8], reflects a continuous search for more accurate congestion signals and more adaptive control strategies. This progression is not linear but rather a branching exploration of different philosophies, each with its own set of strengths, weaknesses, and ideal operating environments. This section provides a comparative analysis of the paradigms undertaken in this paper and outlines potential future directions for the research.

7.1. A Multi-Dimensional Comparison of Paradigms

The used algorithms under different congestion control approaches can be analyzed across several key dimensions: the signal they use to infer congestion, their control mechanism, their performance characteristics in terms of throughput and latency, and their behavior in challenging network environments. Table 1 provides a comprehensive summary of these dimensions, highlighting the distinct trade-offs made by each approach.

Table 1: A Multi-Dimensional Comparison of Congestion Control Paradigms

Paradigm	Algorithm (s)	Primary Congestion Signal	Control Mechanism	Key Strengths	Key Weaknesses / Challenges
Traditional Loss-Based	NewReno [2][14]	Packet Loss	AIMD: Linear increase, 50% decrease	Simple, proven stability, safe	Poor scalability in LFNs, induces bufferbloat, performs poorly on lossy links
High-Speed Loss-Based	CUBIC [4][13], HighSpeed TCP [3]	Packet Loss	Modified AIMD	Excellent scalability in LFNs, RTT-fairness (CUBIC)	Still fundamentally relies on buffer overflow, inherently causes high latency
Delay-Based	TCP Vegas [6]	Queuing Delay (RTT	Proactive window adjustment	Proactively avoids loss, achieves low	Can be too conservative and yield bandwidth to more aggressive loss-

10.48047/jocaaa.2024.32.01.70

		increase)		latency	based flows
Model-Based	BBR [7] [12]	Delivery Rate & Min RTT	Packet pacing based on an explicit network model	High throughput with minimal latency, highly resilient to non-congestion loss	Model accuracy can be challenging, potential fairness issues with deep-buffered loss-based flows
Online Learning-Based	PCC [8]	Empirically Observed Utility	Online learning, rate adjustments based on performance experiments	Highly adaptive to unknown/diverse conditions, no fixed assumptions	Performance is dependent on the utility function, can have slower convergence
Explicit Feedback	ABC [11]	Router-provided Signals	Sender reacts to "accelerate/brake" commands	Extremely fast adaptation to time-varying wireless links	Requires router support (not end-to-end), incremental deployment challenges

7.2. Discussion: The Trajectory of Congestion Signals

The evolution of congestion control is perhaps most clearly reflected in the progressive sophistication of the signals used to infer network state.

- **Reactive Signal — Packet Loss:** Exemplified by TCP Reno and CUBIC, these algorithms increase their sending rate until the network responds with a loss event.
- **Proactive Signal — Queuing Delay:** TCP Vegas marked a critical conceptual shift by introducing queuing delay as a proactive congestion signal, enabling high throughput without excessive queuing delays.
- **Model-Based Signal — Delivery Rate:** BBR represents a further advancement by adopting a model-based perspective, constructing an explicit model of the network’s physical properties.
- **Empirical Signal — Utility:** Finally, PCC introduces the most flexible paradigm by eschewing predefined congestion signals altogether, instead evaluating the empirical outcome of its own actions against a utility function.

7.3. Future Directions

The trajectory of research and deployment points to several key future directions:

1. **The Rise of Model-Based and Learning-Based Control:** The widespread adoption of BBR by major content providers and the compelling performance of PCC in diverse environments suggest that the future of congestion control lies in moving beyond simple loss-based reactions.

10.48047/jocaaa.2024.32.01.70

2. **Hybrid Approaches:** Future algorithms may not be purely model-based or learning-based, but hybrids. For instance, a learning-based system like PCC could be initialized with a baseline model like BBR's to speed up convergence. Similarly, BBR could incorporate learning to adapt its probing behavior more intelligently.
3. **Customizable Performance Objectives:** PCC's concept of a flexible utility function is a powerful one. As networks are used for an increasingly diverse set of applications (from bulk data transfer to real-time augmented reality), the ability for an application to specify its own performance objective (e.g., "minimize latency at all costs" vs. "maximize throughput") will become increasingly important. This opens up a new frontier in application-aware networking.
4. **The Role of Explicit Feedback:** While general-purpose Internet transport will likely remain end-to-end, the success of specialized schemes like ABC in controlled environments (like a cellular operator's own network) suggests a growing role for explicit feedback mechanisms where the network and endpoints can be co-designed. This is particularly relevant with the rise of Software-Defined Networking (SDN) and programmable network hardware.

8. Conclusion

The inherent limitations of using packet loss as a primary signal—namely, bufferbloat and poor performance on lossy links—motivated a philosophical shift toward proactive control. Delay-based approaches, such as TCP Vegas, and the more sophisticated model-based paradigm of BBR, demonstrated that it is possible to achieve the dual goals of high throughput and low latency by utilizing more nuanced network signals. The most recent architectural evolution, exemplified by PCC, abandons fixed assumptions altogether, reframing congestion control as an online learning problem that empirically discovers optimal sending rates. While these advancements have largely adhered to the end-to-end principle, specialized solutions like ABC highlight the performance gains achievable with in-network feedback in challenging environments like wireless links. Although the loss-based CUBIC remains the incumbent, the clear benefits of model-aware and learning-based algorithms are paving the way for a new generation of intelligent, measurement-driven protocols that promise to finally resolve the long-standing trade-off between throughput and latency, delivering a faster and more responsive Internet.

References

[1] M. Allman, V. Paxson, and E. Blanton, "TCP Congestion Control," RFC 5681, Sep. 2009.

[2] W. R. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, Jan. 1997.

[3] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC 3649, Dec. 2003.

[4] S. Ha, I. Rhee, and L. Xu, "CUBIC: A New TCP-Friendly High-Speed TCP Variant," in Proc. International Workshop on Protocols for Fast and Long Distance Networks, 2005.

[5] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 6582, Apr. 2012.

10.48047/jocaaa.2024.32.01.70

[6] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End-to-End Congestion Avoidance on a Global Internet," IEEE J. Sel. Areas Commun., vol. 13, no. 8, pp. 1465-1480, Oct. 1995.

[7] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-Based Congestion Control," Commun. ACM, vol. 60, no. 2, pp. 58-66, Jan. 2017.

[8] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, "PCC: Re-architecting Congestion Control for Consistent High Performance," in Proc. 12th USENIX Symp. Netw. Syst. Des. Implementation (NSDI '15), 2015, pp. 395-408.

[9] K. Ramakrishnan, S. Floyd, and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP," RFC 3168, Sep. 2001.

[10] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, "An Extension to the Selective Acknowledgment (SACK) Option for TCP," RFC 2883, Jul. 2000.

[11] P. Goyal, A. Agarwal, R. Netravali, M. Alizadeh, and H. Balakrishnan, "ABC: A Simple Explicit Congestion Controller for Wireless Networks," in Proc. 17th USENIX Symp. Netw. Syst. Des. Implementation (NSDI '20), 2020.

[12] Dominik Scholz, Benedikt Jaeger, Lukas Schwaighofer, Daniel Raumer, Fabien Geyer, and Georg Carle. Towards a Deeper Understanding of TCP BBR Congestion Control. IFIP Networking Conference (IFIP Networking) and Workshops. 14-16 May 2018.

[13] Rhee, I., Xu, L., Ha, S., Zimmermann, A., Eggert, L., & Scheffenegger, R. (2018). CUBIC for fast long-distance networks (No. RFC 8312).

[14] M. Allman, S. Floyd, C. Partridge. Increasing TCP's Initial Window. RFC 3390. October 2002.