On The Effectiveness of Congestion Control Algorithms on MPT-GRE Networks

Naseer Al-Imareen, Gábor Lencse
Department of Telecommunications
Széchenyi István University
Győr, Hungary
al-imareen.naseer@sze.hu

***Abstract*— Contemporary academic research is seeing a notable surge in interest in investigating the multifaceted aspects associated with the advancement of multipath technologies, an area of prominence within ongoing research efforts. This burgeoning interest is exemplified by the prominence of protocols such as Multipath TCP (MPTCP) and Multipath UDP (MPT), which have appeared as focal points within contemporary research trends. The continual evolution of networking protocols, containing diverse iterations of the Transmission Control Protocol (TCP), including (CUBIC, Reno, Vegas, BBR, etc.) has been characterized by a sustained effort to congestion detection and control algorithms. This paper demonstrates the effectiveness of TCP congestion control algorithms within a network operating under the MPT-GRE network layer multipath technology. Various factors contributing to congestion were added to one of the two paths, including delay or packet loss. The study illustrates the contribution of congestion control algorithms to the increase in network throughput by resolving transient period issues in MPT-GRE multipath networks. The main objective is to evaluate the effectiveness and efficiency of congestion control algorithms within the MPT network architecture. Through systematic analysis and experimental testing, this study provides valuable insights into the performance of congestion detection algorithms. It gives evidence of their significant positive effect in multipath MPT-GRE networks.**

***Keywords*—congestion control; delay; Multipath; MPT-GRE; packet loss; throughput.**

I. INTRODUCTION

Modern communication technology encompasses a wide range of devices that support several networks and applications with multiple interfaces. However, the effectiveness of their communication sessions is dependent on the architecture of the TCP/IP protocol, which only allows single-session handling by default. In addition to improving the throughput and accuracy of the user experience [1]. The requirement to use multiple interfaces simultaneously during communication sessions provides increased flexibility in handling network disruptions.

In response to this demand, several multipath solutions have appeared, such as MPT-GRE [2] and MPTCP [3]. MPT-GRE enables the creation of a logical tunnel across various physical paths, distinguishing it from alternatives such as MPTCP and Huawei's Generic Routing Encapsulation (GRE) Tunnel Bonding Protocol. Exceeding the capacity of network resources, whether through single or multiple paths, MPT throughput performance is often degraded due to delays and congestion [4]. To reduce such issues, various TCP congestion control algorithms have appeared, such as (Reno, CUBIC, Vegas, TCP-Illinois, BBR, etc.). These algorithms operate by detecting and preemptively controlling congestion and avoiding packet loss. Their effectiveness lies in their ability to monitor packet flow from source to destination. While some algorithms adjust congestion window size based on round-trip time (RTT), others, especially those designed for high-bandwidth networks, expand the window size to optimize throughput. Efforts to manage congestion within multipath networks not only support network stability but also ensure packets arrive in the correct order, thereby minimizing packet loss. The fair distribution of network resources among packet flows prevents any single flow from monopolizing bandwidth to prevent throughput degradation [5].

Congestion control techniques enable the most efficient use of network resources, facilitate throughput aggregation, and minimize bandwidth waste. Congestion management is still a popular area for research because of its essential function in preserving network integrity by controlling traffic flow between source and destination. Some research has emphasized the side effects of multipath, such as the lack of TCP friendliness. This problem prevents the uncoupled congestion control method in Multipath TCP (MPTCP), where each subflow independently captures bandwidth, acting like a separate TCP connection [6].

Currently, 95% of internet data transmission is controlled by the TCP protocol. However, when communication sessions deteriorate because of packet loss and uncontrolled traffic carried on by delays and congestion, the user experience frequently degrades [4]. The limitations of conventional single-path techniques can be avoided by diversifying data transmission between sources and destinations across several paths to overcome these obstacles. The MPT library was developed to maximize data transmission by load balancing across several paths, which frequently leads to throughput capacities that are almost the total throughputs of the individual paths combined [7]. Congestion algorithm integration in multipath networks holds great potential for building robust and effective network infrastructures.

This paper aims to handle the issue of transient periods forced by delays when measuring the tunnel throughput of the MPT network. To resolve this, various congestion control algorithms are used that have demonstrated their efficacy in ensuring a consistent data flow through the MPT-GRE tunnel

right from the onset of the measurements, thereby improving the tunnel throughput of the MPT network.

The subsequent sections of this paper are organized as follows. Section II provides an overview of MPT-GRE technology. Section III elaborates on the TCP congestion control algorithms. Section IV explains the experimental environments containing the hardware environment and software configurations. Section V presents the MPT-GRE measurements using diverse congestion control algorithms and their results. Finally, the conclusion and the directions of the future research plans are presented.

II. Overview of the MPT-GRE Technology

The MPT-GRE multipath network is an innovative multipath technology rooted in the GRE-in-UDP tunnel specification framework. The conceptual framework of MPT-GRE emphasizes its architectural design towards network layer multipath communication technology with benefit from the GRE-in-UDP encapsulation, which was defined in IETF RFC 8086 [8]. A critical difference between MPT and MPTCP is that MPT operates at the network layer, while MPTCP functions at the transport layer. The research group conducted the MPT software development efforts within the Faculty of Informatics at the University of Debrecen.

The architecture of the MPT-GRE software library, depicted in Fig 1, complies with the specifications outlined in the IETF RFC 8086, adapted to operate within a multiple path technique. In contrast to conventional TCP/IP protocols, MPT introduces a novel logical (tunnel) layer epitomized through a tunnel path. Although the functioning above the GRE-in-UDP layer remains similar, incoming data from the application layer is routed not to a physical path but to a tunnel path. Subsequently, the MPT software produces traffic mapping from the logical interface to the physical interface. Within the MPT environment, multiple paths are utilized through the capability of the MPT library to dynamically distribute packets

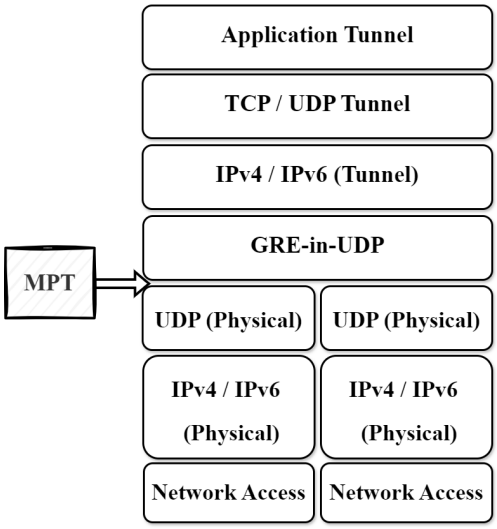

Fig. 1. The Conceptual Architecture of MPT-GRE [2].

incoming from the logical path (tunnel interface) across the available physical paths. Therefore, this enables multipath communication for applications by redistributing incoming packets across paths, wherein the application program uses only the tunnel interface for communication. IP tunnel versions and IP path versions operate independently within the MPT library; therefore, MPT-GRE can also facilitate the IPv6 transition. A detailed description of MPT's operational principles is available in the prior papers [9], [10], [11].

III. TCP Congestion Control Algorithms

Congestion control algorithms in computer networks are essential mechanisms designed to manage and control congestion when the demand on network resources exceeds their specified bandwidth capacity or when one of the incoming packet flows exploits the bandwidth, leading to the deterioration of network throughput due to increased latency or delay, and thus the probability of packet loss [12]. These algorithms aim to manage data flow using various strategies, some of which rely on monitoring incoming packets from source to destination, eventually leading to improved network performance and increased throughput aggregation and ensuring fair allocation of resources resulting in a robust network. Various TCP congestion control algorithms, such as CUBIC, HSTCP, TCP-Illinois, Reno, Vegas, Veno, BIC, Westwood, Hybla, Scalable, and BBR [13], are used in this paper. The critical function of these algorithms is to adjust the transmission rate based on the network conditions and structure. Some use slowly starting packets to avoid congestion and fast retransmission. The delayed starting mechanism progressively increases the transmission speed until congestion happens. Some authors suggested a practical solution called C-MPBBR, a BBR-based CCA designed for real-world scenarios involving multiple paths [14].

Moreover, congestion control preserves a stable transmission speed of packets to avoid additional congestion by precisely monitoring incoming packets [15]. The fast retransmission feature immediately retransmits lost packets when packet loss is detected, reducing the influence of network congestion, and increasing channel throughput aggregation [4]. Some congestion detection and control algorithms, such as TCP Vegas and CUBIC, rely on monitoring estimated round-trip times (RTT) to detect congestion and increase transmission rates if it occurs. This approach effectively reduces the damage of multiple packet losses within a single transmission window, thus improving network throughput aggregation [16]. As a result, congestion control algorithms play a crucial role in maintaining network stability, efficiency, and robustness [17]. As networks expand, ongoing research and development in congestion control remains critical to handle emerging challenges and provide resilient network performance in many scenarios.

IV. Experimental Test Environment

A. Hardware Environment Details

As shown in Fig 2, a test system was built to perform measurements and analyses. Here are the specifications for the two Dell PowerEdge R620 servers that were used:

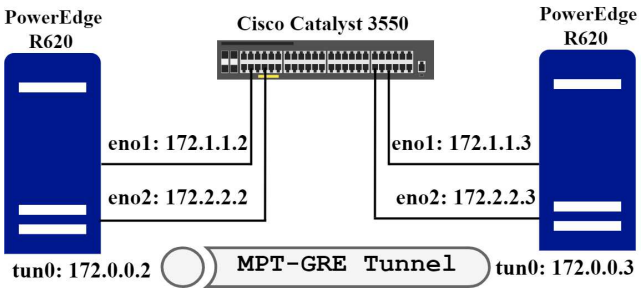

Fig. 2. The MPT-GRE Multipath Network Topology.

- 4x8 GB 1333MHz DDR3 SDRAM.
- Dual-core Intel Xeon E5-2620 processors clocked at 2.00 GHz.
- Network Interface Card (NIC): Intel quad-port Gigabit Ethernet controller, of which two ports were used for the examination.

The two interfaces used were connected using a Cisco Catalyst 3550 switch with a transmission limit of 100Mbps. Two separate physical paths were established between the interfaces (eno1, eno2), where the IP address setting of the experimental network may also be observed in the figure, and the MPT-GRE software enabled the tunnel (tun0). Both servers operated on Ubuntu 22.04.4 LTS (Jammy Jellyfish) / Linux operating system. This experimental design aims to assess the efficacy of the MPT-GRE tunnel throughput library and to supervise and validate the effectiveness of several congestion control algorithms.

B. MPT configuration environment

The MPT software setup requires the use of a 64-bit version, especially the version mpt-gre-lib64-2019.tar.gz, which may be found in [16]. Two files located in the MPT installation directory required modifications to the configuration. The general interface information, such as local command UDP port number, interface number, acceptance of remote requests, and specifics regarding tunnel interfaces, including name, maximum transfer unit, and IPv4 address with prefix length, were configured, and described in the first file `conf/interface.conf`. Similar configuration methods were applied to the interface of the second server. Further, the tunnel files were categorized into distinct connection files that included all relevant IP addresses and constant-length data for each MPT-GRE software, thereby establishing the logical path (tunnel).

Moreover, in the second file, connections were organized and arranged, featuring IPv4 over IPv4 connection specifications outlined in the `conf/connections/IPv4.conf` file. This file contained connection information such as sending and receiving permissions, IP versions, local and remote IP addresses, ports, and path weights. Our prior paper [9] provided detailed descriptions of MPT configuration file settings; these configuration files have been publicly available on GitHub [18].

V. Experiments with TCP Congestion Control Algorithms

The primary objective of this study is to confirm the influence of congestion control algorithms on MPT-GRE multipath networks and prove the optimal throughput aggregation value for the tunnel. Moreover, a comprehensive evaluation of tunnel throughput is critical in enhancing our understanding of MPT-GRE operations. A range of scenarios was designed to conduct the experiments with precision and effectiveness and to facilitate the experimental process throughout the 100-second duration; a Python script was employed, ensuring automation and accuracy. This scripting not only facilitated frequent execution but also automated the saving of the tunnel throughput results, thus enhancing the consistency and repeatability of our results.

Furthermore, a comprehensive set of TCP congestion control algorithms, including CUBIC, HSTCP, TCP-Illinois, Reno, Vegas, Veno, BIC, Westwood, Hybla, Scalable, and BBR, were combined into the Python code. These algorithms were employed with iperf3 [19] within the program code. By analyzing the performance of congestion control algorithms in the MPT-GRE multipath network, particularly when considering two paths with symmetric speeds and varying delay times and packet loss, the study involves a multipath network with two paths, each having a speed of 100Mbps. Two scenarios were applied: the impact of delay and packet loss were investigated with and without congestion control algorithms.

A. Impact of Delay with and without Congestion Control Algorithms

The first scenario included adding various delay times on the first physical path using the traffic control **`tc`** command:

```
tc qdisc add dev eno2 root netem delay xms
```

The values of variable x have been set to 10ms, 20ms, 30ms, 40ms, 50ms, and 100ms to represent the delay periods. The primary metric used for evaluating the performance of the algorithms is tunnel throughput. Tunnel throughput measures the amount of data successfully transmitted through the network tunnel over a given period. Higher tunnel throughput indicates better utilization of network resources and improved data transmission efficiency. The tunnel throughput of MPT-GRE was measured with and without congestion control algorithms in both scenarios. The study evaluated the performance of several congestion control algorithms (CUBIC, HSTCP, TCP-Illinois, Reno, Vegas, Veno, BIC, Westwood, Hybla, Scalable, and BBR). Each algorithm employs distinct congestion avoidance and control strategies, including window-based approaches, rate-based approaches, or hybrid mechanisms. Our results upon measuring the throughput aggregation of the MPT-GRE tunnel disclosed a transient period at the beginning of the measurement, later stabilizing over time.

Fig 3 illustrates sample results from the initial 21 seconds, depicting the transient behavior observed during measurements conducted without the application of any congestion control algorithm. Analysis of these results determined that the impact

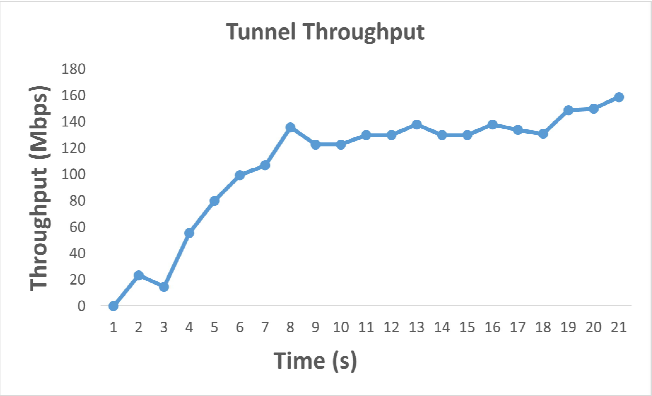

Fig. 3. The MPT-GRE tunnel throughput transient period (delay: 50ms).

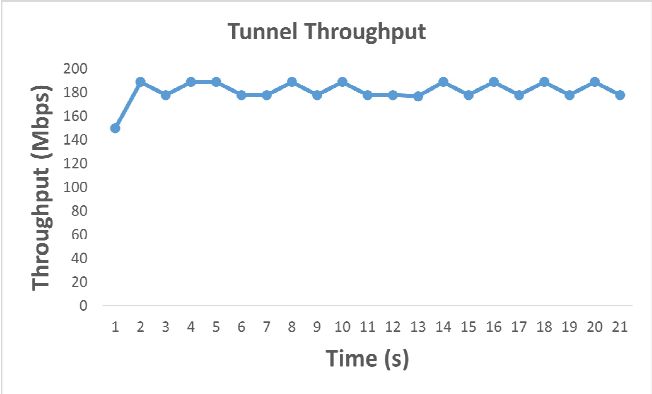

Fig. 4 The MPT-GRE tunnel throughput transient period (delay: 50ms) with the HSTCP congestion control algorithm.

of delay on tunnel throughput was observed, leading to performance challenges within the MPT connection. Fig 4 shows the sample results from the initial 21 seconds, how using some of the congestion control algorithms, such as the HSTCP algorithm, resolved transient period issues in MPT-GRE multipath networks.

Table I demonstrates the influence of delay on MPT-GRE tunnel throughput when we added delay to the first path. Analyzing the results, we find that as the delay increases, it impacts the MPT tunnel throughput, increasing the duration of the transient period. On the other hand, it elaborates on the impact of the congestion control algorithms on the tunnel throughput of the MPT-GRE software and how to enhance the tunnel throughput with some congestion control algorithms; the findings indicate that specific congestion control algorithms, like HSTCP, BIC, and Scalable TCP, could improve tunnel throughput even with added delay. These algorithms effectively manage network resources under different conditions, reduce the negative impact of delay and enhance overall performance.

B. Impact of Packet loss with and without Congestion Control Algorithms

The second scenario included adding packet loss of different percentages to the first path too, using the traffic control `tc` command:

```
tc qdisc add dev eno1 root netem loss x%
```

The values of variable x have been set to 1% to 5% represent the packet loss percentage. Fig 5 illustrates the impact of packet loss on the tunnel throughput of the MPT-GRE. Analyzing the results, we find that as packet loss increases, it significantly influences the MPT tunnel throughput, increasing the duration of the transient period. It also elaborates on the influence of congestion control algorithms on MPT-GRE tunnel throughput aggregation with packet loss and how some control algorithms improved the tunnel throughput of MPT. Furthermore, this figure focuses on the effectiveness of different congestion control algorithms in reducing these packet loss effects on the MPT-GRE tunnel, focusing on how BBR (bottleneck bandwidth and round-trip propagation time) could enhance tunnel throughput under Packet loss conditions.

VI. Conclusion and Future Research Directions

We examined the impact of various TCP congestion control mechanisms in a multipath MPT-GRE network setting. Through systematic experimentation and analysis, we assessed the efficiency of different TCP congestion control algorithms

TABLE I. The Effectiveness of the Congestion Control Algorithms on the MPT-GRE Tunnel Throughput with Various Delay Values.

Congestion Control Algorithms	Delay= 10 ms	Delay= 20 ms	Delay= 30 ms	Delay= 40 ms	Delay= 50 ms	Delay= 100 ms
Uncontrolled	180	178	179	178	169	165
CUBIC	174	178	180	179	166	170
HSTCP	180	180	183	183	182	182
TCP-Illinois	180	175	183	180	182	171
Reno	176	182	183	119	124	56.3
Vegas	5.65	3.13	2.11	1.64	1.15	0.609
Veno	180	152	106	158	181	112
BIC	180	180	179	182	182	180
Westwood	174	134	109	130	167	45
Hybla	178	166	142	115	182	96.2
Scalable	182	182	179	182	182	179
BBR	28.6	14.5	21.2	22.2	4.96	10.6

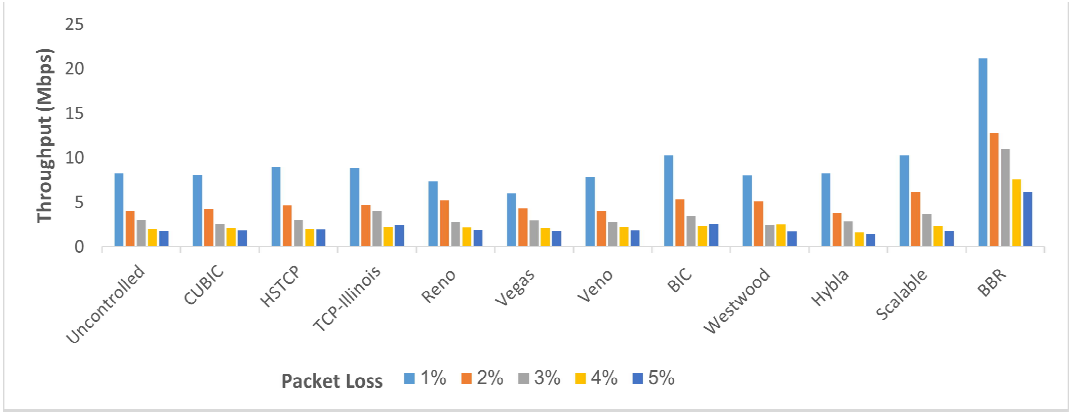

Fig. 5. The effectiveness of the congestion control algorithms on the MPT-GRE tunnel throughput with various packet loss rates.

in improving the tunnel throughput of the MPT-GRE networks and fixing transient period issues. The results showed that the performance of the MPT-GRE networks was much enhanced after using TCP congestion management techniques. Our results showed that several TCP congestion control algorithms, including Scalable, HSTCP, and BIC reduced the negative impact of (up to 100ms) delay measures. As for the negative impact of the significant amount of packet loss (up to 5%), BBR was the only one that could efficiently reduce it. As a result, the tunnel throughput of MPT-GRE was improved in both scenarios, but significant improvement was facilitated by a non-overlapping set of congestion control algorithms. With other words, none of the tested TCP congestion control algorithms could efficiently reduce the negative impact of both the delay and the packet loss, therefore, we conclude that there is room for developing a novel congestion control algorithm to be used with MPT-GRE multipath networks.

One of our future research plans is to design a customized algorithm for the MPT-GRE software that benefits the basic ideas of those TCP congestion control algorithms that have proven effective under certain conditions.

REFERENCES

[1] Z. A. Polgar, "Reliable Data Transport Protocol with FEC Mechanism for Erasure Channels," in *2021 44th International Conference on Telecommunications and Signal Processing (TSP)*, 2021, pp. 114–119, doi: 10.1109/TSP52935.2021.9522618.

[2] B. Almási, G. Lencse, and S. Szilágyi, "Investigating the multipath extension of the GRE in UDP technology," *Comput. Commun.*, vol. 103, pp. 29–38, 2017, doi: 10.1016/j.comcom.2017.02.002.

[3] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, "TCP extensions for multipath operation with multiple addresses," 2013.

[4] M. Morawski and P. Ignaciuk, "Influence of Congestion Control Algorithms on Head-of-Line Blocking in MPTCP-based Communication," *27th Telecommun. Forum, TELFOR 2019*, pp. 48–51, 2019, doi: 10.1109/TELFOR48224.2019.8971059.

[5] J. Wang, J. Wen, J. Zhang, and Y. Han, "TCP-FIT: An improved TCP congestion control algorithm and its performance," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 2894–2902, doi: 10.1109/INFCOM.2011.5935128.

[6] Y. Thomas, G. Xylomenos, and G. C. Polyzos, "Multipath congestion control with network assistance," *Comput. Commun.*, vol. 153, pp. 264–278, 2020, doi: https://doi.org/10.1016/j.comcom.2020.01.071.

[7] S. Szilágyi, I. Bordán, L. Harangi, and B. Kiss, "MPT-GRE: A Novel Multipath Communication Technology for the Cloud," *9th IEEE Int. Conf. Cogn. Infocommunications, CogInfoCom 2018 - Proc.*, pp. 81–86, 2018, doi: 10.1109/CogInfoCom.2018.8639941.

[8] L. Yong, E. Crabbe, X. Xu, and T. Herbert, "GRE-in-UDP encapsulation," 2017.

[9] N. Al-Imareen and G. Lencse, "On the Impact of Packet Reordering in MPT-GRE Multipath Networks," in *2023 46th International Conference on Telecommunications and Signal Processing (TSP)*, Jul. 2023, pp. 82–86, doi: 10.1109/TSP59544.2023.10197737.

[10] N. Al-Imareen and G. Lencse, "Effect of Path QoS on Throughput Aggregation Capability of the MPT Network Layer Multipath Communication Library," *Infocommunications J.*, vol. 15, no. 2, pp. 14–20, 2023, doi: 10.36244/ICJ.2023.2.3.

[11] N. Jabbar and G. Lencse, "An Overview of the Multipath Technologies, their Importance and Types," *Proc. Tech. Univ. Sofia*, vol. 72, no. 1, pp. 21–28, Apr. 2022, doi: 10.47978/TUS.2022.72.01.004.

[12] S. Szilágyi and I. Bordán, "The effects of different congestion control algorithms over multipath fast ethernet IPv4/IPv6 environments," *CEUR Workshop Proc.*, vol. 2650, pp. 341–349, 2020, [Online]. Available: https://api.semanticscholar.org/CorpusID:221662730.

[13] R. Al-Saadi, G. Armitage, J. But, and P. Branch, "A Survey of Delay-Based and Hybrid TCP Congestion Control Algorithms," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 4, pp. 3609–3638, 2019, doi: 10.1109/COMST.2019.2904994.

[14] I. Mahmud, T. Lubna, Y.-J. Song, and Y.-Z. Cho, "Coupled Multipath BBR (C-MPBBR): A Efficient Congestion Control Algorithm for Multipath TCP," *IEEE Access*, vol. 8, pp. 165497–165511, 2020, doi: 10.1109/ACCESS.2020.3022720.

[15] F. Paganini, Z. Wang, J. C. Doyle, and S. H. Low, "Congestion control for high performance, stability, and fairness in general networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 43–56, 2005, doi: 10.1109/TNET.2004.842216.

[16] H. Jamal and K. Sultan, "Performance analysis of TCP congestion control algorithms," *Int. J. Comput. Commun.*, vol. 2, no. 1, pp. 30–38, 2008, [Online]. Available: http://w.naun.org/multimedia/UPress/cc/cc-27.pdf.

[17] T. A. N. Nguyen, S. Gangadhar, and J. P. G. Sterbenz, "Performance Evaluation of TCP Congestion Control Algorithms in Data Center Networks," in *Proceedings of the 11th International Conference on Future Internet Technologies*, Jun. 2016, pp. 21–28, doi: 10.1145/2935663.2935669.

[18] N. Al-Imareen and G. Lencse, "MPT connections files," 2023. https://github.com/NaseerAJabbar/MPT_Connections_files.

[19] K. P. Jon Dugan, Seth Elliott, Bruce A. Mah, Jeff Poskanzer, "Iperf3 documentation," 2018. https://iperf.fr/iperf-doc.php.