PCMJ

Progress in Computer and Mathematics Journal (PCMJ)
volume 2 [August, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

TRANSMISSION CONTROL PROTOCOL (TCP) CONGESTION ANALYSIS OF WIRED NETWORK USING NETWORK SIMULATOR 2

Intan Zulaikha Zulkifli
College of Computing, Informatics, and Mathematics
Universiti Teknologi MARA
Jasin, Melaka
intanzulaikha369@gmail.com

Mohd Hafifi Mohd Supir*
College of Computing, Informatics, and Mathematics
Universiti Teknologi MARA
Jasin, Melaka
hafifisupir@uitm.edu.my

Article Info

Abstract

The study investigates the performance of various Transmission Control Protocol (TCP) congestion control mechanisms—TCP Reno, TCP NewReno, and TCP Vegas—within wired network environments using the NS-2 network simulator. The primary research problem focuses on evaluating the efficiency, stability, and throughput of these TCP variants in response to network congestion, with the aim of identifying the most effective protocol for maintaining high performance under different network conditions. The methodology involves simulating a wired network environment in NS-2 and analyzing key performance metrics such as throughput, packet loss, and congestion window behavior for each TCP variant. The analysis reveals that TCP NewReno generally offers better performance compared to TCP Reno in scenarios involving multiple packet losses due to its improved fast recovery algorithm. TCP Vegas, on the other hand, demonstrates superior throughput and lower delays by proactively avoiding congestion through RTT-based adjustments, making it the most efficient protocol in stable network conditions. However, TCP NewReno proves more reliable than TCP Reno in environments with higher packet loss. The findings suggest that TCP Vegas is well-suited for networks requiring consistent performance and minimal delay, while TCP NewReno is preferable in networks where packet loss is a more frequent issue. The study concludes that TCP Vegas, with its congestion avoidance strategy, provides the best overall performance, though TCP NewReno remains a strong contender in more volatile network conditions, thereby offering a nuanced understanding of TCP protocol efficiency in wired networks.

Received: August 2024
Accepted: March 2025
Available Online: August 2025

Keywords: TCP, Congestion Control, Wired Networks, Network Simulator 2 (NS-2), TCP NewReno, TCP Reno, TCP Vegas

INTRODUCTION

Transmission Control Protocol (TCP) plays a crucial role in ensuring reliable data transmission across networks, particularly in wired networks where maintaining a consistent flow of data is vital for effective communication (Grazia, 2019). Over the years, various TCP variants have been developed to enhance performance in diverse network conditions, each employing different strategies to manage congestion. Among these, TCP Reno, TCP NewReno, and TCP Vegas are widely recognized for their unique approaches to congestion control. Understanding the performance implications of these protocols in wired networks is essential for optimizing network efficiency and ensuring robust data transmission.

The evolution of TCP has been driven by the need to address specific challenges such as packet loss, network congestion, and the efficient utilization of available bandwidth (Aung & Ohsaki, 2023). TCP Reno introduced significant improvements over its predecessor, TCP Tahoe, by incorporating fast retransmit and fast recovery mechanisms, which reduce the frequency of complete slow starts. TCP NewReno further refines these mechanisms by improving the handling of multiple packet losses within a single window of data. On the other hand, TCP Vegas adopts a fundamentally different approach by focusing on congestion avoidance through proactive adjustments based on round-trip time (RTT) estimations. This diverse range of strategies highlights the importance of selecting the appropriate TCP variant for specific network conditions.

Despite the advancements made by these TCP variants, there remains a need for a comprehensive analysis of their performance in wired networks, where the nature of congestion and data flow can differ significantly from wireless or hybrid environments. Previous studies have often focused on specific aspects of these protocols, such as their response to packet loss or their throughput under varying conditions. However, there is a gap in the literature when it comes to directly comparing the performance of TCP Reno, TCP NewReno, and TCP Vegas within a controlled, wired network environment. This comparison is critical for network administrators and engineers who must choose the most suitable protocol to optimize network performance.

This study aims to address this gap by conducting a detailed performance analysis of TCP Reno, TCP NewReno, and TCP Vegas using the NS-2 network simulator. By simulating a wired network environment, the study will evaluate these protocols based on key performance metrics such as throughput, packet loss, and congestion window behavior. The results of this analysis will provide valuable insights into the strengths and weaknesses of each protocol, offering guidance for selecting the most effective TCP variant for specific wired network scenarios.

LITERATURE REVIEW

The Transmission Control Protocol (TCP) has undergone numerous modifications since its inception, leading to the development of various versions tailored to address specific challenges in network communication. TCP Reno, TCP NewReno, and TCP Vegas are among the most widely studied variants, each introducing unique mechanisms to manage congestion, maintain throughput, and reduce packet loss in wired network environments.

TCP Reno, an evolution of TCP Tahoe, introduced the fast retransmit and fast recovery mechanisms, which significantly improved network performance by reducing the need for a complete retransmission timeout upon detecting packet loss. Jacobson (1988) initially proposed TCP Reno, focusing on minimizing network congestion by responding quickly to packet loss events. While TCP Reno's approach significantly enhanced network efficiency, it was later observed that its performance diminished in scenarios involving multiple packet losses within a single transmission window, leading to the development of TCP NewReno.

TCP NewReno builds upon the foundations laid by TCP Reno by refining the fast recovery mechanism, enabling it to handle multiple packet losses more effectively without resorting to a full slow start. Floyd and Henderson (1999) provided a detailed analysis of TCP NewReno, highlighting its ability to maintain higher throughput in high-loss environments by selectively acknowledging successfully received packets. This selective acknowledgment (SACK) strategy allows TCP NewReno to recover from multiple losses more gracefully, making it more robust than its predecessor under adverse network conditions.

On the other hand, TCP Vegas takes a proactive approach to congestion management, focusing on congestion avoidance rather than congestion detection and recovery. TCP Vegas, as proposed by Brakmo and Peterson (1995), uses round-trip time (RTT) measurements to predict potential congestion before it leads to packet loss. By adjusting the sending rate based on these predictions, TCP Vegas can maintain a more stable congestion window, resulting in smoother throughput and lower latency compared to TCP Reno and TCP NewReno. Research by Ahn et al. (1999) has shown that TCP Vegas generally outperforms other TCP variants in environments where network conditions are stable, due to its emphasis on congestion avoidance.

However, despite the advantages of TCP Vegas in stable network conditions, its performance can be inconsistent in environments with highly variable traffic. This variability can cause TCP Vegas to underutilize available bandwidth, as its conservative approach may not fully exploit the network's capacity. Consequently, while TCP Vegas provides superior performance in specific scenarios, TCP NewReno's balance between throughput and responsiveness to congestion makes it a more versatile choice in diverse network conditions.

Comparison Between TCP Congestion Control Variants

Table 1 shows the comparison between three TCP congestion control variants. The comparative analysis of these TCP variants is important to gain deeper insights into their behavior under various network conditions.

Table 1 The Comparison Between TCP Congestion Control Variants

TCP Variants	Objective Situation	Congestion Window Control	Lost Detection	Slow Start	Complexity
Vegas	High packet loss, wireless networks	Additive increase, multiplicative decrease	Delay-based	Slow, adapts to delay	High, requires delay measurement
NewReno	General purpose	Larger window during loss	Loss-based	Avoids premature restarts	Moderate
Reno	High-speed, long-distance networks	Additive increase, multiplicative decrease	Loss-based	Standard slow start	Moderate

Comparison Between TCP Congestion Control Techniques

Table 2 shows the comparison between four TCP congestion control techniques. It is essential to make comparative analysis to understand the trade-offs.

Table 2 The Comparison Between TCP Congestion Control Techniques

TCP Techniques	Focus	Action	Advantages	Disadvantages
Slow Start	Window size growth	Increase window size slowly	Prevent network overload	Slow initial data transfer
Congestion Avoidance	Window size adjustment	Increase window size slightly after successful transmissions	Minimizes delays	Can be sensitive to network changes
Fast Retransmission	Retransmission speed	Resend lost packets immediately	Minimizes delay	May worsen congestion if overused
Fast Recovery	Window size control after retransmission	Adapt window size based on network response	Maintains smooth data flow after recovery	Requires careful window size adjustments

METHODOLOGY

The wired network topology was designed using NS-2, which is widely recognized for its ability to simulate various network protocols accurately. The network comprises multiple nodes connected through a series of routers, with data traffic generated between source and destination nodes. Each TCP variant (Reno, NewReno, Vegas) was implemented in the simulation environment, and identical network scenarios were executed for each protocol to ensure a fair comparison. Figure 1 illustrates the network topologies used in this project. Mesh topology consists of interconnected nodes allowing multiple paths for data transmission.

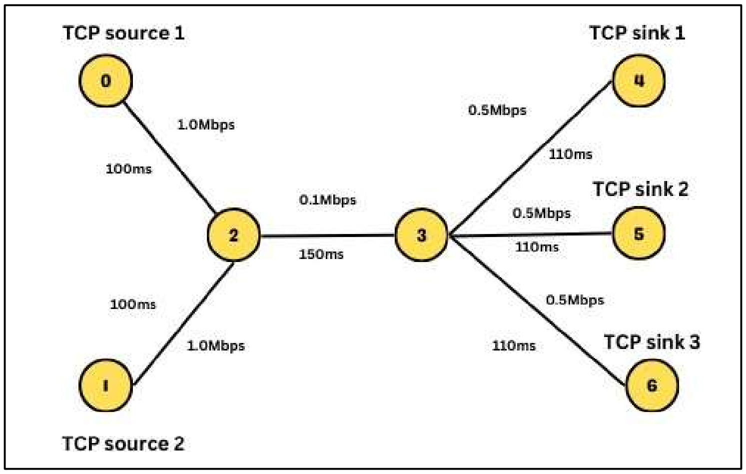

Figure 1 Mesh Topology

Table 3 shows the simulation parameters to conduct this simulation. All other scenarios use the same simulation parameters except for the changes of the value that are based on the scenario specifications.

Table 4.1 Simulation Parameters

Parameters	Value
Channel Type	Wired
Number of nodes	7
Packet Size	1500
Simulation Time	130 s
Link Bandwidth	0.1 - 1.0 Mbps
Link Delay	100 – 150 ms
Queue Type	Droptail
Application	FTP

RESULT AND DISCUSSION

This part presents the findings and analysis of the project, focusing on the impact of TCP congestion control mechanisms in various network scenarios. The results and analysis are based on five distinct scenarios designed to thoroughly evaluate the performance of TCP in different network conditions.

Scenario 1: Network Topology Impact on Network Congestion Performance

In this scenario, three distinct network topologies which are Linear, Star, and Mesh topology were configured to investigate the performance of TCP congestion control mechanisms under varying network structures. The Figures illustrate the graph of behavior of three different TCP congestion control algorithms.

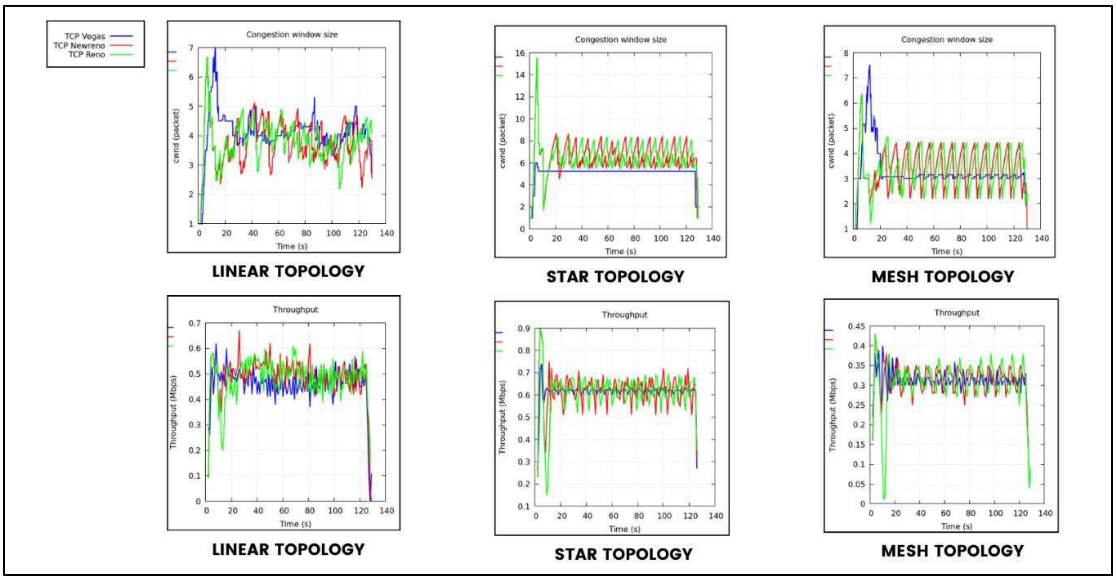

Figure 2 Cwnd Size and Throughput Over Time on Different Network Topology

Based on Figure 2, the cwnd size for TCP Vegas stabilizes quickly and shows less instability compared to TCP Reno and TCP NewReno. The cwnd for TCP NewReno fluctuates significantly but recovers faster from packet losses compared to TCP Reno. The throughput data for TCP Reno shows fluctuations with some noticeable rises and drops. For networks requiring stable and consistent throughput, TCP Vegas is recommended due to its proactive congestion avoidance.

PCMJ
Progress in Computer and Mathematics Journal (PCMJ)
volume 2 [August, 2025]
e-ISSN: 3030-6728
Website: fskmjebat.uitm.edu.my/pcmj

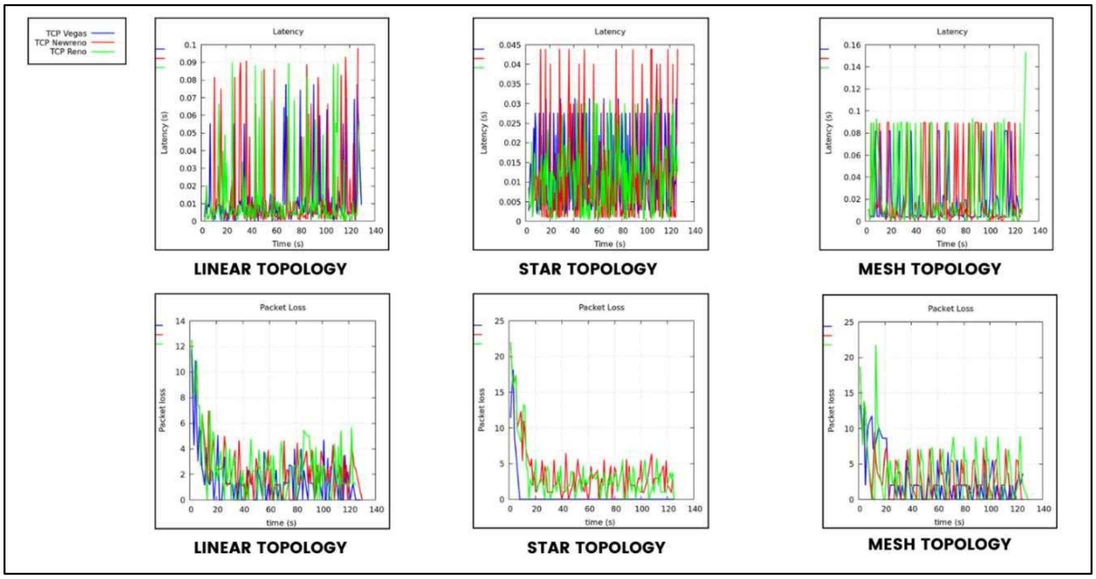

Figure 3 Latency and Packet Loss Over Time on Different Network Topology

Based on Figure 3, the latency data for TCP Reno shows significant fluctuations with a frequent rise reaching up. TCP NewReno exhibits similar characteristics to Reno but with slightly lower latency. TCP Vegas's latency remains low and stable for the most time, with fewer and less rise compared to Reno and NewReno. TCP Vegas demonstrates a different behavior due to its proactive congestion avoidance strategy. The packet loss decreases after the initial phase and remains low with fewer and less severe rises compared to Reno and NewReno.

Scenario 2: Comparing Single TCP Flow to Multiple TCP Flows

This scenario investigates the performance of TCP congestion control mechanisms by comparing a single TCP flow to multiple TCP flows. The aim is to understand how different TCP variants manage varying traffic loads and the resulting congestion. The Figure illustrates the graph of behavior of three different TCP congestion control algorithms.

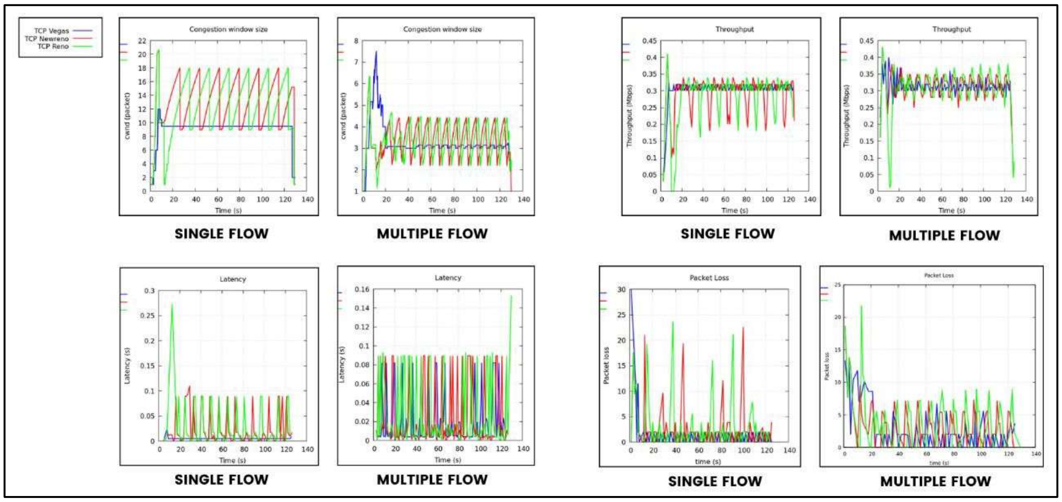

Figure 4 Cwnd, Throughput, Latency, and Packet Loss Over Time on Different TCP Flow

Based on Figure 4, it shows that cwnd size is higher on single flow TCP. In the multiple flow scenario, TCP Vegas stabilizes at a cwnd size of around 3 packets, while TCP NewReno and TCP Reno fluctuate between 3 and 4 packets. Throughput is higher on multiple TCP flows. In the multiple TCP flows scenario, TCP Vegas continues to exhibit stable performance, maintaining throughput around 0.35 Mbps

Scenario 3: Impact of Bandwidth Variation on TCP Performance

In this scenario, the adaptability of TCP protocols to different bandwidth capacities was evaluated. By altering the bandwidth from the lowest to the highest levels, the study aimed to observe how TCP congestion control mechanisms respond to changes in available bandwidth.

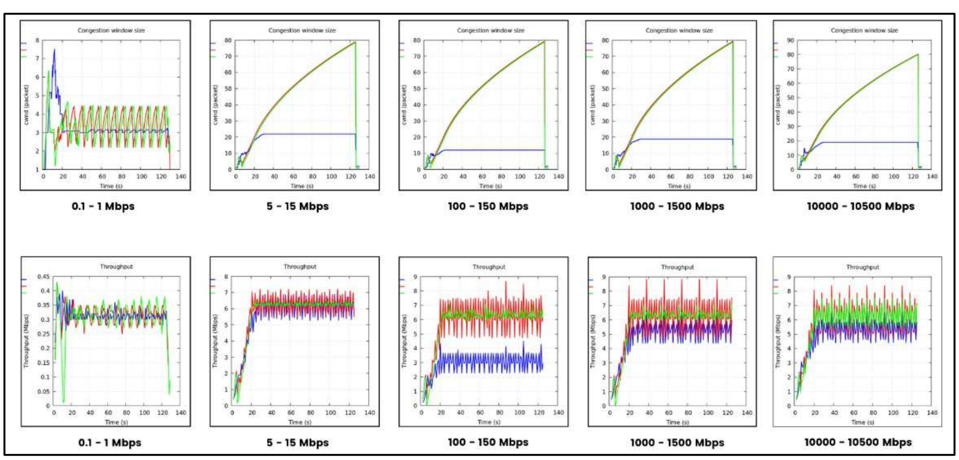

Figure 5 Cwnd and Throughput Over Time on Different Bandwidth

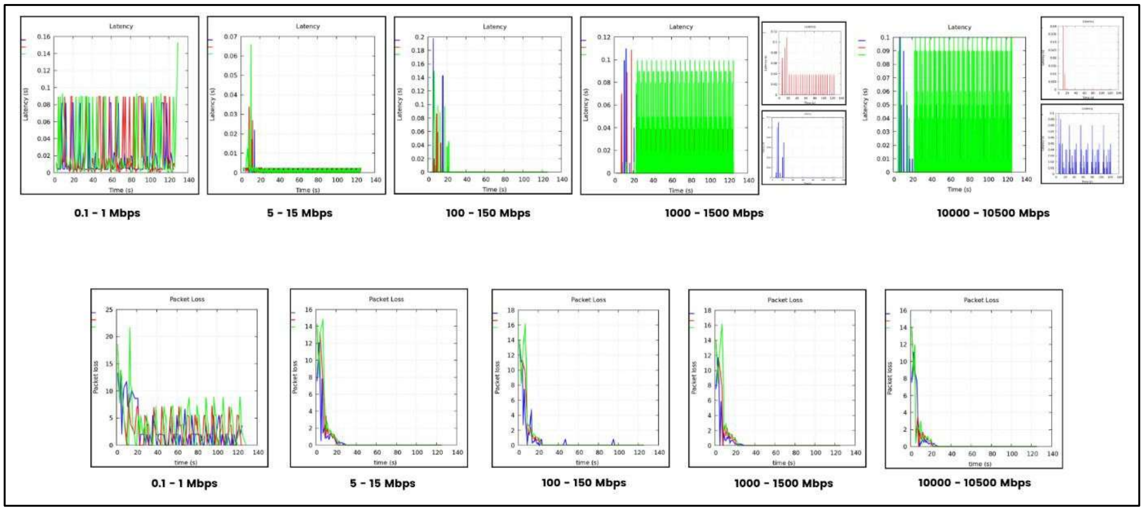

Figure 6 Latency and Packet Loss Over Time on Different Bandwidth

Based on Figure 5 and 6, TCP Reno and NewReno increase their cwnd sizes as bandwidth rises, with NewReno showing more stability and control. TCP Vegas maintains a conservative cwnd, prioritizing low latency and stability over maximizing bandwidth. TCP Vegas shows minimal drops and quick recovery at lower bandwidths, while TCP NewReno and Reno fluctuate more but stabilize around 0.30-0.35 Mbps. At mid and higher bandwidth, TCP Reno achieves relatively low packet loss, TCP Vegas consistently demonstrates the lowest packet loss. As the bandwidth increases, the packet loss decreases because more capacity for data transmission.

Scenario 4: Effects of Delay Variation on TCP Performance

This scenario explored the impact of different delays on TCP performance. By adjusting the delay from the lowest to the highest levels, the study aimed to understand how latency affects TCP congestion control mechanisms.

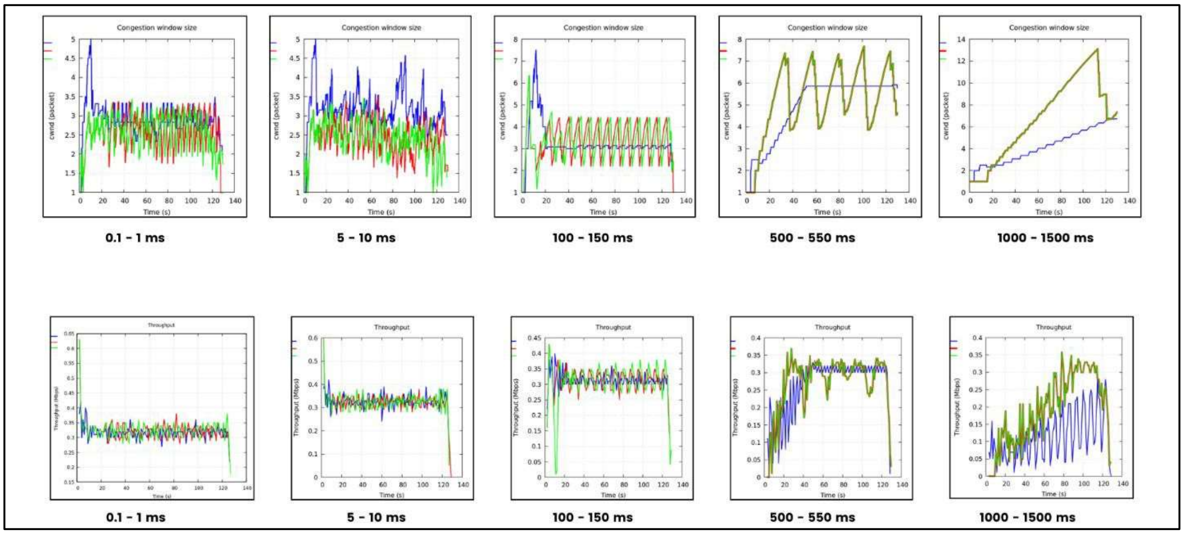

Figure 7 Cwnd and Throughput Over Time on Different Delay

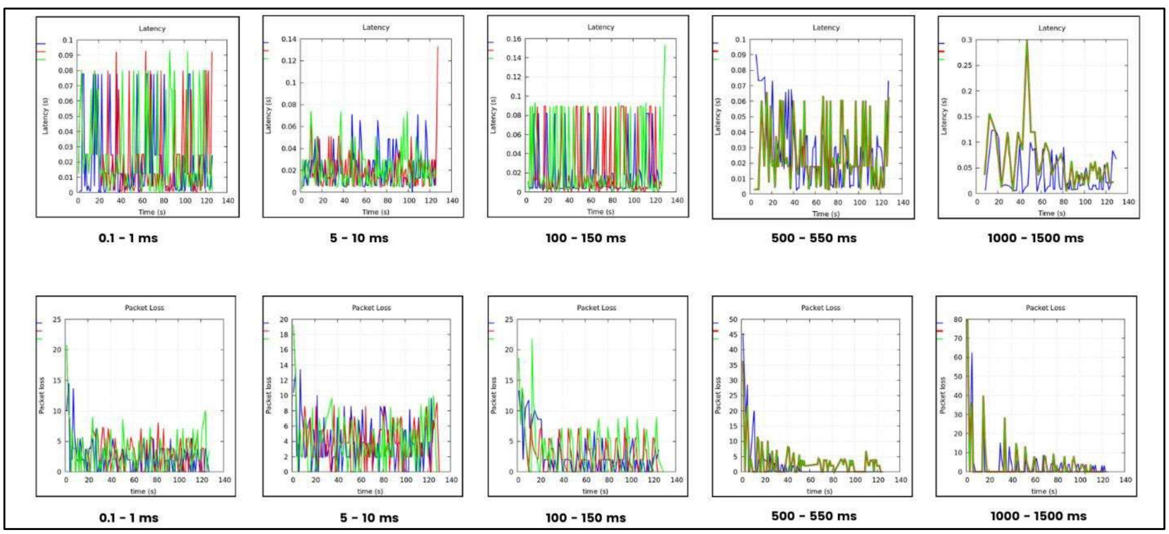

Figure 8 Latency and Packet Loss Over Time on Different Delay

Based on Figure 7 and 8, TCP Vegas, which employs a congestion avoidance strategy, demonstrates slightly higher cwnd than Reno and NewReno at low delay. As delays increase to moderate levels (100 - 150 ms), throughput fluctuations rise for all variants, but TCP Vegas remains more stable. Under high delay conditions (500 - 1500 ms), TCP Vegas maintains lower latency compared to TCP NewReno and TCP Reno. TCP Vegas manages to maintain relatively lower packet loss due to its delay-based control mechanisms.

Scenario 5: Impact of DropTail and Fair Queueing (FQ) Queue Management on Network Congestion

The final scenario examined how different queue management techniques, specifically DropTail and Fair Queueing (FQ), affect network congestion and overall performance. The study aimed to determine how these techniques influence congestion control, packet loss, and network efficiency

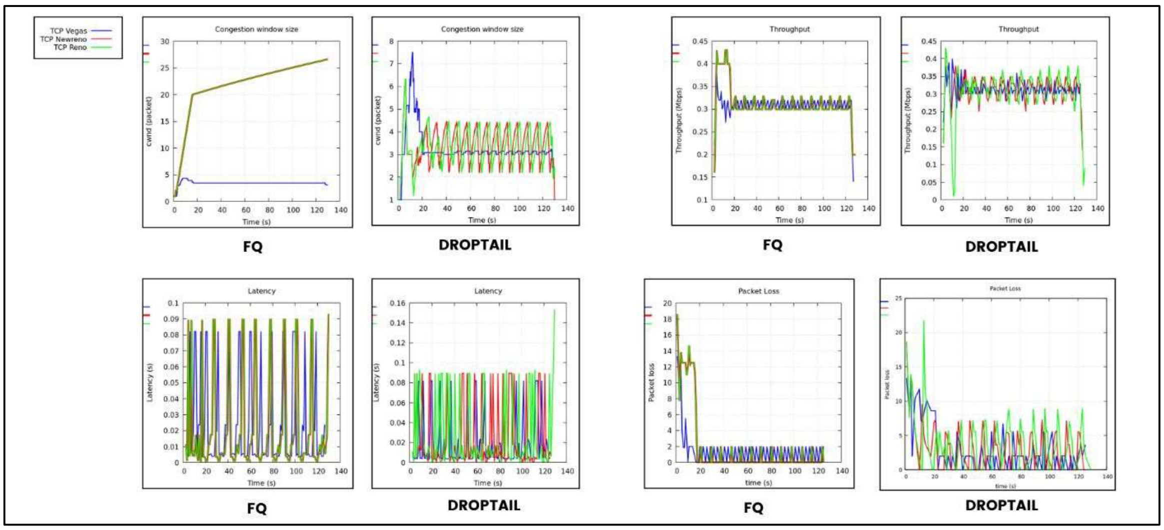

Figure 9 Cwnd, Throughput, Latency, and Packet Loss Over Time on Different Queue

Based on Figure 9, Fair Queuing (FQ) queue management provides a more consistent environment for TCP congestion control, enabling TCP NewReno and TCP Reno to reach higher and steadily increasing congestion window sizes, while TCP Vegas maintains a stable lower congestion window due to its proactive approach to congestion avoidance. TCP Vegas maintaining consistently lower latency rise while TCP NewReno and TCP Reno show slightly higher peaks around 0.08 to 0.09 seconds. In contrast, Droptail leads to a higher latency rise, with TCP Reno reaching peaks up to 0.14 seconds. FQ results in stable and low packet loss for all TCP variants, with an initial rise that quickly stabilizes to near zero, showing minimal and infrequent rise, particularly for TCP Vegas.

CONCLUSION

The study concludes that the choice of TCP congestion control algorithm should be based on the specific network conditions. TCP NewReno is a robust choice for networks with moderate to high bandwidth and low to moderate latency, while TCP Vegas is more suitable for high-latency networks where proactive congestion control is required. The findings contribute to the field by providing insights into optimizing TCP performance in wired networks. Future research could focus on hybrid algorithms that combine the strengths of different TCP variants to enhance performance further.

REFERENCES

Hussain, S. Z., & Parween, S. (2021, December 17). Comparative Study of TCP Congestion Control Algorithm in IoT. 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). https://doi.org/10.1109/icac3n53548.2021.9725642

Zhang, J., Yao, Z., Tu, Y., & Chen, Y. (2020, October 23). A Survey of TCP Congestion Control Algorithm. 2020 IEEE 5th International Conference on Signal and Image Processing (ICSIP). https://doi.org/10.1109/icsip49896.2020.9339423

Paliwal, G., Sharma, K. P., Taterh, S., & Varshney, S. (2019, November). A New Effective TCP- CC Algorithm Performance Analysis (NS3). 2019 4th International Conference on Information Systems and Computer Networks (ISCON). https://doi.org/10.1109/iscon47742.2019.9036301

Grazia, C. A. (2019, October). IEEE 802.11n/ac Wireless Network Efficiency under different TCP Congestion Controls. 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). https://doi.org/10.1109/wimob.2019.8923418

Sutawijava, B., & Basuki, A. (2019, September). The Behavior and Performance of TCP Congestion Control Algorithms Under Multi-Path Routing. 2019 International Conference on Sustainable Information Engineering and Technology (SIET). https://doi.org/10.1109/siet48054.2019.8986012

Himeedah, J., Hamouda, A., Taher, M., & Sati, S. (2022, May 23). TCP Congestion Analysis of Wired and Wireless Links Using NS3. 2022 IEEE 2nd International Maghreb Meeting of the Conference on Sciences and Techniques of Automatic Control and Computer Engineering (MI-STA). https://doi.org/10.1109/mi-sta54861.2022.983774

Aung, H. N., & Ohsaki, H. (2023, June). On the Potential of Modern TCP Congestion Control Algorithms in Information-Centric Networking. 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC). https://doi.org/10.1109/compsac57700.2023.00167

Abadleh, A., Tareef, A., Btoush, A., Mahadeen, A., Al-Mjali, M. M., Alja'Afreh, S. S., & Alkasasbeh, A. A. (2022, June 21). Comparative Analysis of TCP Congestion Control Methods. 2022 13th International Conference on Information and Communication Systems (ICICS). https://doi.org/10.1109/icics55353.2022.9811217

Biswal, S. P., Gupta, S., Kumar, A., & Patel, S. (2022, December 16). Comparative Analysis of Compound TCP with Various End-to-End Congestion Control Mechanisms. 2022 4th International Conference on Advances in Computing, Communication Control and Networking (ICAC3N). https://doi.org/10.1109/icac3n56670.2022.10074083

Patel, S., Shukla, Y., Kumar, N., Sharma, T., & Singh, K. (2020, March). A Comparative Performance Analysis of TCP Congestion Control Algorithms: Newreno, Westwood, Veno, BIC, and Cubic. 2020 6th International Conference on Signal Processing and Communication (ICSC). https://doi.org/10.1109/icsc48311.2020.9182733

Dzivhani, M., & Ouahada, K. (2019, September). Performance Evaluation of TCP Congestion Control Algorithms for Wired Networks using NS-3 Simulator. 2019 IEEE AFRICON. https://doi.org/10.1109/africon46755.2019.9133880

Jin, X., Yang, Y., Ma, J., & Li, Z. (2019, June). Congestion Control of Wireless Sensor Networks based on L1/2 Regularization. 2019 Chinese Control and Decision Conference (CCDC). https://doi.org/10.1109/ccdc.2019.8832586

Han, K., Lee, J. Y., & Kim, B. C. (2019, January). Machine-Learning based Loss Discrimination Algorithm for Wireless TCP Congestion Control. 2019 International Conference on Electronics, Information, and Communication (ICEIC). https://doi.org/10.23919/elinfocom.2019.8706382

Xie, H., Jing, Y., Chen, J., & Liu, Y. (2022, August 15). Adaptive fuzzy fixed-time congestion control for TCP/AWM networks. 2022 34th Chinese Control and Decision Conference (CCDC). https://doi.org/10.1109/ccdc55256.2022.10033539

Shrivastava, P., Meeradevi, & Mundada, M. R. (2022, December 21). Survey on Congestion Control Approaches in 4G/5G Cellular Networks. 2022 4th International Conference on Circuits, Control, Communication and Computing (I4C). https://doi.org/10.1109/i4c57141.2022.10057675

Gao, Q., Li, J., Liu, Y., & Xing, Y. (2023). Optimal Control Method for Congestion Control and Delay Reduction in Deterministic Networks. IEEE Access, 11, 82912–82923. https://doi.org/10.1109/access.2023.3301964

Zhao, L., Qu, S., Huang, X., & Luo, J. (2019, June). Congestion Control of Wireless Sensor Networks using Discrete Sliding Mode Control. 2019 Chinese Control and Decision Conference (CCDC). https://doi.org/10.1109/ccdc.2019.8833135

Sone, K., Ito, Y., & Hasegawa, A. (2019, October). Proposal of New TCP Congestion Control to Suppress QoS Fluctuation for Web Services. 2019 International Conference on Information and Communication Technology Convergence (ICTC). https://doi.org/10.1109/ictc46691.2019.8939775

Qi, X., Jing, Y., & Dimirovski, G. M. (2020, August). Finite-time Congestion Control for a Class of TCP Network Systems. 2020 Chinese Control and Decision Conference (CCDC). https://doi.org/10.1109/ccdc49329.2020.9164545

Hu, J., Zhao, L., Feng, Y., Liu, Y., & Gao, M. (2022, September). Research, Implementation and Practice of Congestion Control Mechanism in LTE-V2X. 2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall). https://doi.org/10.1109/vtc2022-fall57202.2022.10012979

Cheng, L., Siyuan, C., Zhiqiang, Y., & Dachang, L. (2019, September). Wireless Sensor Networks Congestion Control Protocol Based on Intelligent Power Allocation. 2019 IEEE 2nd International Conference on Information Communication and Signal Processing (ICICSP). https://doi.org/10.1109/icicsp48821.2019.8958485

Do, H., Gregory, M. A., & Li, S. (2019, November). SDN-based Wireless Access Networks Utilising BBR TCP Congestion Control. 2019 29th International Telecommunication Networks and Applications Conference (ITNAC). https://doi.org/10.1109/itnac46935.2019.9078015

Wei, W., Xue, K., Han, J., Wei, D. S. L., & Hong, P. (2020, April). Shared Bottleneck-Based Congestion Control and Packet Scheduling for Multipath TCP. IEEE/ACM Transactions on Networking, 28(2), 653–666. https://doi.org/10.1109/tnet.2020.2970032

Deguchi, Y., Kobayashi, A., Tarutani, Y., Fukushima, Y., & Yokohira, T. (2022, October 19). Throughput Fairness in Congestion Control of Multipath TCP. 2022 13th International Conference on Information and Communication Technology Convergence (ICTC). https://doi.org/10.1109/ictc55196.2022.9952427

Imaduddin, M. F., Putrada, A. G., & Karimah, S. A. (2021, August). Multipath TCP Scheduling Performance Analysis and Congestion Control on Video Streaming on the MPTCP Network. 2021 International Conference on Software Engineering & Computer Systems and 4th International Conference on Computational Science and Information Management (ICSECS-ICOCSIM). https://doi.org/10.1109/icsecs52883.2021.00109

Sun, G., Li, C., Ma, Y., Li, S., & Qiu, J. (2023, March). End-to-End TCP Congestion Control as a Classification Problem. IEEE Transactions on Reliability, 72(1), 384–394. https://doi.org/10.1109/tr.2022.3172335

Kieu, M. V., Nguyen, D. T., & Nguyen, T. T. (2020, October). A Way to Estimate TCP Throughput under Low-Rate DDoS Attacks: One TCP Flow. 2020 RIVF International Conference on Computing and Communication Technologies (RIVF). https://doi.org/10.1109/rivf48685.2020.9140777