International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Performance Evaluation of Modified TCP Congestion Control Algorithm Under Different Data Rates Environment

Dharmendrasinh D. Zala[1], Ajay Kumar Vyas[2*]

Submitted: 19/01/2024 **Revised**: 28/02/2024 **Accepted**: 05/03/2024

Abstract : Wireless Ad Hoc Networks (WAHNs) provide a flexible and dynamic communication infrastructure, making them ideal for situations where traditional wired networks are impractical. However, WAHNs present unique challenges for the Transmission Control Protocol (TCP), primarily due to their dynamic topologies, varying link qualities, and limited bandwidth. Traditional TCP congestion control algorithms often underperform in WAHN environments, leading to suboptimal throughput, increased latency, and inefficient resource utilization.

To improve performance, two approaches have been proposed: a congestion management technique where the congestion window's size is modified at the time of retransmission, and the other technique is from active queue management technique where modification in the algorithm of Random Early Detection protocol is applied. Simulation-based comparative analysis of the original TCP Westwood protocol and modified TCP Westwood is being conducted, assessing key metrics parameters such as packet delivery throughput, ratio of packet dropping, goodput, and total latency.

The Modified TCP-W and modified RED protocols have demonstrated improvements in both throughput and goodput while maintaining stable performance in terms of total latency and packet drop correlation. The proposed congestion control enhancements will be thoroughly evaluated through extensive simulations using Network Simulator (NS 2) software experiments in diverse WAHN scenarios.

Keywords*: TCP-W, WAHN, utilization, Network Simulator (NS 2), demonstrated, Westwood*

1 Introduction

Wireless communication requires efficient congestion control for smooth data transmission. Congestion issues become increasingly important as wireless network demand and device connectivity rise. Combining Multiphase Nonlinear Random Early Detection (RED) [1-6] with Modified TCPW may reduce wireless link congestion [7][8]. This research examines how this integrated technique can improve congestion control, network efficiency, and wireless data flow.

There are different types and extensions of TCP that work with different networks and their needs. This lets network administrators and engineers choose the best choice for each use case and performance goal. Some types of TCP are TCP Reno, which is aggressive during congestion and often leads to global synchronization, TCP New Reno, which improves on Reno to make it faster to resend lost packets [9], and TCP Cubic, which keeps throughput high but may cause queue sizes to be high [10], TCP Vegas, TCP BIC, which strikes a balance between high throughput and low queue sizes [11], and TCP Westwood, is specially modified for the wireless communication and also widely used as it provides good performance and fairness [12].

The literature review shows that As TCP Westwood is multilayer-designed, more equitable, flexible, and less prone to packet loss, it is a superior option for wireless connection networks[13].

Active Queue Management (AQM) is a set of techniques used to manage network congestion by monitoring and controlling the length of network queues. TCP Westwood, a congestion control algorithm, can work in conjunction with AQM mechanisms to improve network performance and fairness [14]. AQM helps to prevent or minimize network congestion, which is crucial for maintaining good performance in TCP connections. Popular AQM techniques include Random Early Detection (RED), which provides early congestion signaling, good for responsiveness, and fairness. ECN (Explicit Congestion Notification) allows TCP Westwood to adapt without packet loss and strong fairness. FQ-CoDel (Fair Queueing with Controlled Delay) [15] offers excellent fairness, low queuing delays, and minimal loss. PIE (Proportional Integral Controller Enhanced) [16] offers low latency, high throughput, and maintains good performance. DRR (Deficit Round Robin) provides good fairness among flows and simple and efficient scheduling. CoDel (Controlled Delay) reduces queuing delays and avoids buffer bloat, while SFQ (Stochastic Fair Queuing) minimizes flow

[1]*Research Scholar, Department of Electronics & Communication Gujarat Technological University (GTU), Ahmedabad-Gujarat, India.*
[2*]*Associate Professor, Department of Electronics & Communication, Adani Institute of Infrastructure Engineering, (GTU), Ahmedabad-Gujarat, India.*
**Corresponding Author: Ajay Kumar Vyas*
Email: ajay_ap7@yahoo.com

starvation. According to the literature, TCP Westwood with RED AQM improves timely congestion signaling, fairness, packet loss, and network efficiency [17]. This combination is especially useful in wireless and high-latency networks where performance and fairness are critical.

Any particular algorithm can identify and solve the problem of loss of packets as packet loss can be because of bit error or it can be because of congestion. It's also possible to make congestion control better by splitting the chance of dropped packets into more than one stage in the queue system of the current transmission control protocol. Therefore, the proposed method to change the algorithm in the Westwood protocol and the random early discovery queue mechanism is important for creating a better transmission control protocol algorithm. The goal of the work is to study and analyse various TCP protocol variations and algorithms for mobility, active queue mechanism approaches, routing protocols, and congestion control technique. Additionally, the objective is to look into how the original protocol can be changed to address issues with wireless environment detection. A dynamic, large-scale environment will be used to test the improved algorithm and analyse its impact on network performance [18].

2 Proposed Methodologies

This paper suggests two changes to the original TCP algorithm. If the TCP Westwood protocol is modified, It is achievable to determine the reason behind packet loss, which may be a bit of error or congestion, and then make a decision for the subsequent phase of transmission mode based on the results. The Random Early Detection (RED) technique could be modified to calculate the probability of packet loss in response to varying load conditions.

2.1 Modified TCP Westwood to identify the reason for packet loss

TCP Westwood, which is widely used and an essential component of the Linux operating system, is unable to determine the reason for the loss of packet, Whether it is due to traffic congestion or bit mistakes in wireless connections. Many older versions of TCP have used different ways to recover lost packets. As an example, the congestion window (cwnd) was limited to 1 in the classic TCP protocol due to timeouts. Similarly, TCP Tahoe set cwnd to 1 in addition to using a rapid recovery technique for segment repetition. TCP Reno utilized a system that was activated by three DUPACKs, which triggered the fast recovery mode upon receiving three DUPACKs but failed due to repeated segment losses [19]. In TCP New Reno, multiple losses were fixed, but bandwidth delay problems were not addressed properly. Selective Acknowledgments (SACK) were not used to tell the difference between different types of loss. In comparison to the other variations, TCP Westwood is significantly superior in terms of performance, as it increases throughput on wireless lines. It does, however, lower the congestion window when there is a timeout or when there are three DUPACKs, though it doesn't have to do this every time because of things like bit errors or random losses. The cwnd value is always cut in half or lowered to 1 by TCP Westwood. On wireless links, TCP Westwood is unable to distinguish between problems due to bit errors as well as losses due to congestion. When the router buffer size is lower than 10% of the product of bandwidth delay, TCP Westwood could not perform well using the queue management approach. The biggest issue with TCP Westwood is that it frequently decreases CWND when it encounters losses [20]. Thus, using the flag (F) in the TCP Header, we presented a novel method in our study to identify the cause of loss. We use the one-bit flag from the header of TCP [23] to show whether there is bit traffic or an incorrect bit. For congestion, we set F = 0 and for bit error, we set F = 1. Our algorithm works by keeping an eye on the sequence numbers at the receiving end. If a section is lost during transmission, the receiving end will notice out-of-order sequence numbers. As soon as the receiver gets these, it starts keeping track of the times for the next parts and keeps sending out Duplicate Acknowledgements (DUPACKs). If the next segments have delays, which means there is congestion in the medium, the receiver marks the loss as being caused by congestion by setting FLAG = 0 in the third DUPACK. In the other case, if segments follow each other without any delays, which means there is no congestion, the receiver sets FLAG = 1 in the third DUPACK. This method focuses on the third DUPACK because it is the only thing that shows when a TCP Westwood (TCPW) packet has been lost. In TCP Westwood, when a sender gets three DUPACKs, they instantly cut the congestion window (cwnd) in half without checking to see if the loss was caused by congestion or a bit error. This slows down real-time communication. Therefore, our study aims to enhance TCPW's capabilities by changing its protocol.

When our suggested approach identifies three DUPACKs, it does not immediately cut the sender's congestion window in half. It instead looks at the FLAG bit condition of the third duplex. If F = 0 in the third DUPACK, the Sender concludes the packet loss was caused by crowding and reduces cwnd by half. This initiates TCP's fast retransmission phase. If F = 1 in the third DUPACK, however, the Sender believes that the packet loss was due to a bit of mistake and does not decrease the size of cwnd. Instead, it goes into the TCP congestion avoidance phase. The goal of the discussed method is to boost performance, and our findings clearly show that TCPW and the changed TCPW protocol are not performing at the same level.

The suggested algorithm for Modified TCP Westwood (MTCPW) is shown in **Figure 1**. It is very complicated. Compared to the standard Linux-oriented TCP Westwood (TCPW), this new method has better performance and higher throughput. In the MTCPW approach, we use symbols such as cwnd (window of congestion), F (header of TCP flag), Seq (value of sequence), Rx_ Seg (segment of receiving), and DUP-ACK (Duplicate Acknowledgement). Here this improved approach maintains all of the fundamental features of the original TCPW. Slow start, avoiding congestion, quick retransmission, and quick recovery phases are a few of these. [21].

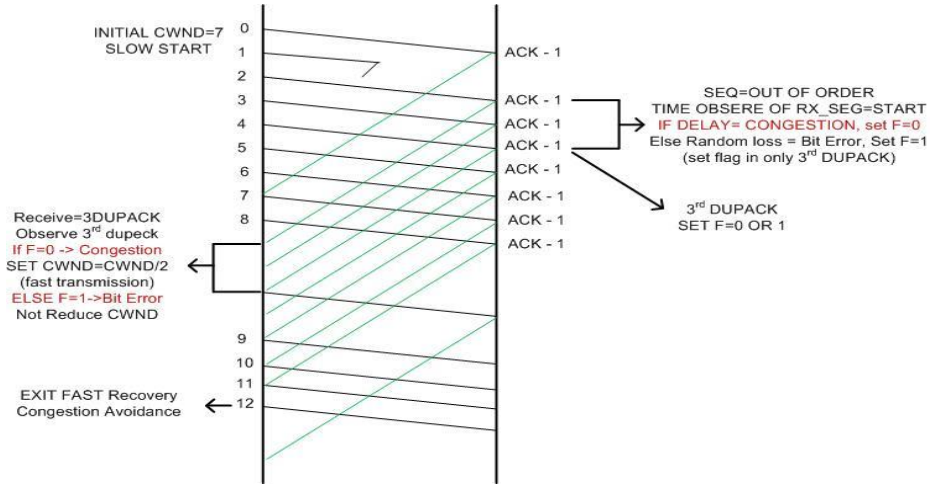


Fig 1. Modified TCP Westwood algorithm for congestion control

Fig 1 shows the starting state, where the client sets a cwnd of 7. At the t1 time, the client will transmit the frame-0, which will be stored by the server in its memory and then send ACK-1. Since the cwnd is set to 7, the client keeps sending frames without waiting for an acknowledgment. But frames don't come in order because frame 1 gets lost, which could be because of too much traffic or a bit mistake. As a result, the server gets frame 2 before frame 1, which makes it check for later frame times and send DUPACKs for frame 1. If there is a pause in the frames that follow, the server assumes that there is congestion and sets FLAG = 0 in the third DUPACK. When the client gets 3 DUPACKs in a row, it doesn't instantly lower the cwnd. Instead, it checks the TCP header's flag status in the third DUP-ACK. When Flag = 0, which means that the frame drop was caused by traffic congestion, the client halves cwnd and starts the quick retransmit part of the TCP algorithm. When FLAG = 1, which means a bit of mistake caused a frame loss, the client stays in the cwnd state and TCP moves to the congestion avoidance phase. When modified TCPW and NS-2 simulation data are used, the throughput is better than with traditional TCP Westwood.

2.2 Modified Random Early Detection for better performance under different load conditions

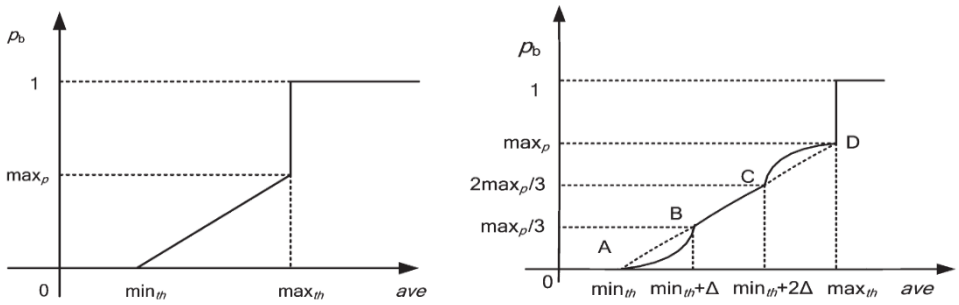
Active Queue Management (AQM) system RED was created by using algorithmic methods and is a queue-based system that replaces the Drop Tail method in TCP/IP networks. The goal is to find early signs of congestion, make sure that all flows are treated equally, manage queue lengths, stop packet drop correlation and global synchronization, reduce packet loss as much as possible, and improve network performance by improving link capacity. RED also tries to keep packet loss to a minimum by managing the chance of dropping packets carefully and making sure that available link capacity is used efficiently [22].

RED was created with three goals in mind: to improve network speed, reduce problems caused by congestion, and encourage equitable resource distribution among various data flows. The RED scheme applies a calculated chance to dropping packets and uses the average queue length (ave) to let traffic sources know about possible network congestion before it happens. EWMA, which stands for the exponentially weighted moving average, is the method that is used to calculate this average length of queue [20]. As per the fundamental terms, the EWMA functions as a low-pass filter, smoothing out changes in the immediate queue length [23]. This allows it to provide a measurement that is more constant and reliable. The amount of smoothing that is applied is determined by a weighting factor that is denoted by the symbol wq.

Moreover, it is important to remember that the average length of the queue [24] can be represented as

$$ave = wq\ q + (1 - wq)\ ave.$$---------- Equation 1

The current queue length is denoted by the letter 'q' in the equation that was given, while the weighting factor is denoted by the letter 'wq', which is a number that lies within the range of [0, 1]



(a) (b)

Fig 2. (a) Probability curve of dropping in RED and **(b)** Probability curve of dropping in Modified RED

Three parameters comprise the Random Early Detection (RED) mechanism: the highest threshold (maxth), the lowest threshold (minth), and the highest dropping probability (Pmax) at maxth. Weighted exponentially moving average (EWMA) is represented by "wq." As long as the average period of the queue is less as compared to the minimum length, RED does not drop any frames. On the opposite side, RED begins to drop receiving packets with a chance proportionate to the average period of the length if the average period of queue length exceeds the minth but remains below the maxth. Whenever the average duration of queue length exceeds maxth, RED discards the entire incoming packets. (a) shows the packet failure probability curve for RED. **Fig 2** (b) shows the expected packet-dropping function T-RED packet-dropping probability function, which was derived from the earlier analysis. It creates a connection between the average duration of queue length and the possibility that a packet would be dropped. Since the average queue length is an indication of packet traffic, it is possible to estimate the likelihood of a packet dropping by constantly monitoring this parameter. The relationship between these variables sheds light on the network's dynamic behavior and enables a more sophisticated understanding of how the system reacts to different traffic and congestion levels. The equation for

$$P_b = \begin{cases} 0 & \text{, ave } \epsilon \text{ [0, Min]} \\ Max\_p\left(\frac{Ave-m}{max-m}\right. & \text{, ave } \epsilon \text{ [Min, max]} \\ 1 & \text{, ave } \epsilon \text{ [max, } +\infty\text{]} \end{cases} \quad ---$$

--- Equation 2

To create an effective Active Queue Management (AQM) system, forced drops, and link underutilization must be avoided. Furthermore, it should be able to withstand a wide variety of loads, as noted in previous research [25][26]. The relationship between the initial packet discard probability in RED and the average queue size is nonlinear, which adds to the difficulty of designing an effective AQM system. However, as the average queue length approaches the lowest threshold, indicating a reasonably uncongested network condition, the packet drop probability should fall below that of RED. This slight adjustment is crucial for fine-tuning the AQM's reactivity to changing network conditions and ensuring that the system reacts effectively to less severe congestion scenarios. By modifying the packet drop probability in this way, the AQM architecture tries to achieve a compromise between network efficiency and congestion control, optimizing performance over a range of operating situations. This will make the best use of the network and could increase throughput. Also, the rate at which the packet drop probability changes should slow down as the average length of the queue goes down. This is to make sure that during this phase, all packet drop probabilities stay lower than those of RED. For the most part, during this phase, the average queue duration should cause the slope of the packet discard probability curve to rise.

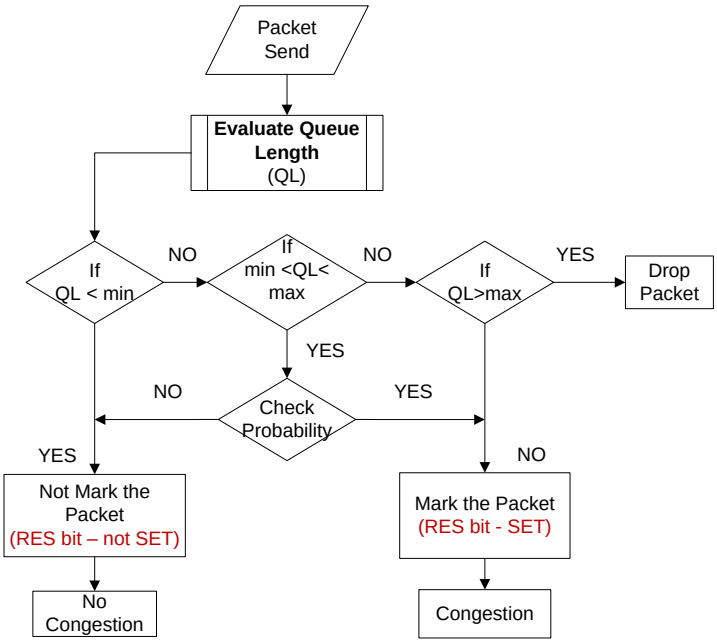


Fig 3 Proposed Modified Random Early Detection

However, when the average queue duration approaches the highest threshold, indicating a severely congested network, the likelihood of packet drops must exceed that of RED. This change is required to reflect the increased congestion and to ensure a more proactive and efficient response to increased network traffic. In such cases, a larger probability of packet drops is justified as a strategic measures to reduce congestion and maintain more consistent and manageable network performance. Allowing a larger risk of packet drops in certain conditions allows the system to better react to and manage the increased demand, reducing further degradation of network performance during peak activity [27]. This keeps the queue from getting too long and cuts down on the average wait time, which changes based on the overall duration of the queue [28]. Also, as the overall duration of the queue goes down, the rate of change in the chance of a packet being dropped should speed up in this phase to better deal with severe congestion [29]. At this point, the slope of the curve showing the chance of a packet being dropped decreases as the overall duration of the queue grows.

The packet discard probability function is best measured using the power form of the steady-state queue length distribution. Typically, an exponent range of 2 to 3 is regarded as suitable for this function. However, it is understood that an exponent value of 3 acts as an upper limit due to the possibility of oscillations. As a result, to reduce the risk of oscillations, this paper uses a packet-dropping probability exponent function set to 3. This selection seeks to achieve a balance between the need for precise computations and the avoidance of instability problems associated with greater exponents. Furthermore, to investigate the effect of various traffic loads on RED's performance, this study divides the average length of queue (ave) into the limits of (min_th, max_th), as shown in **Fig 2**. The segmentation process divides the range into three same divisions with gap sizes of $\Delta = (Max\_th - Min\_th)/3$. Based on these split portions, subsequent discussions will seek to enhance ways of managing varied traffic levels.

$$Ave \in (Min\_th, \Delta + Min\_th) \ldots\ldots\ldots\ldots\ldots\ldots Equation\ 3$$

It is possible that the link bandwidth is not fully utilized during times of low load and mild congestion within this range, which results in minimal average delay. To improve link utilization, it is suggested to use a dropping probability lower than that of RED.

$$Ave \in [\Delta + Min\_th, 2\Delta + Min\_th) \ldots\ldots\ldots\ldots\ldots\ldots\ldots Equation\ 4$$

The link bandwidth is fully used when there is limited load and current congestion in this range, which results in a moderate average delay. In this case, it makes sense to keep the fundamental RED uniform packet-dropping possibility, which is as follows:

$$Ave \in [2\Delta + Min\_th, max\_th) \ldots\ldots\ldots\ldots\ldots\ldots Equation\ 5$$

The link bandwidth is fully used when there is a lot of traffic in this range and a lot of people using it at the same time. This causes a big average delay. Because of this, it is best to use a packet-dropping probability higher than RED to lower the average delay by limiting the transmit speed at the sources.

3 Simulation Setup and Results

Here simulation algorithm is designed and implemented in the network simulator software (NS). Using software NS with version 2.35 setup the modified protocol outcome is compared with the original protocol. Here analysis was carried out for different numbers of nodes from 25 to 100 for the AODV protocol. Here results were carried out for the different data rates from 1 Mbps to 5 Mbps, different node speeds, and packet size as shown in **Fig 4**.

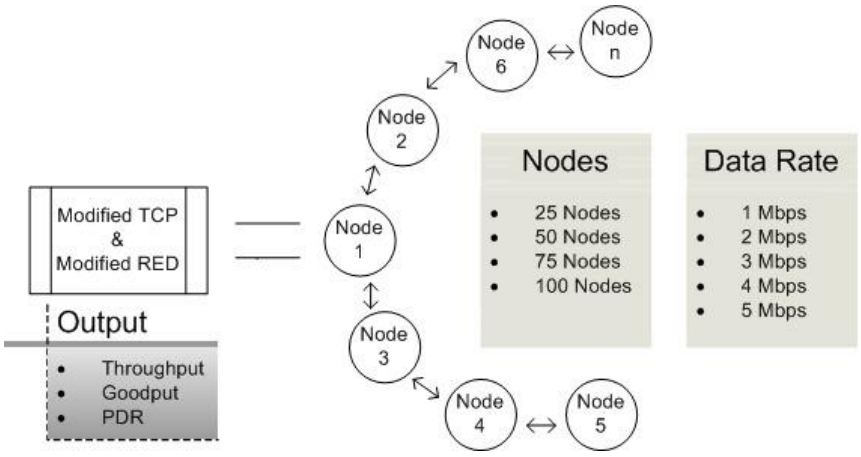


Fig 4 Proposed scheme's topology diagram.

A variety of parameters were tested in the simulations, such as the number of mobile nodes (25, 50, 75, 100), the size of the queues (512, 2048), the minimum and maximum thresholds for the queues (five

and fifteen, respectively), the data rates (ranging from 1 Mbps to five Mbps), and the length of the queues (fifty). The NS-2.35 simulator was employed to replicate the wireless ad-hoc network environment [30]. The AODV routing protocol [19][20] was consistently used for packet routing in all situations, with a designated area size of 500m x 500m.

Table 1. Throughput comparison of Original TCPW and RED with Modified TCPW and RED

Through put	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED
Bitrate / Nodes	1 Mbit/s		2 Mbit/s		3 Mbit/s		4 Mbit/s		5 Mbit/s	
25	325.58	323.47	532.7	541.15	671.7	672.59	768.9	770.73	885.27	879.97
50	560.93	560.93	838.28	842.6	1079.56	1085.67	1232.43	1230.66	1339.34	1354.18
75	542.46	545.17	955.51	921.09	1157.43	1157.43	1321.22	1275.57	1400.27	1446.15
100	560.75	560.75	890.77	894.35	1119.96	1120.23	1258.5	1261.87	1369.8	1369.8

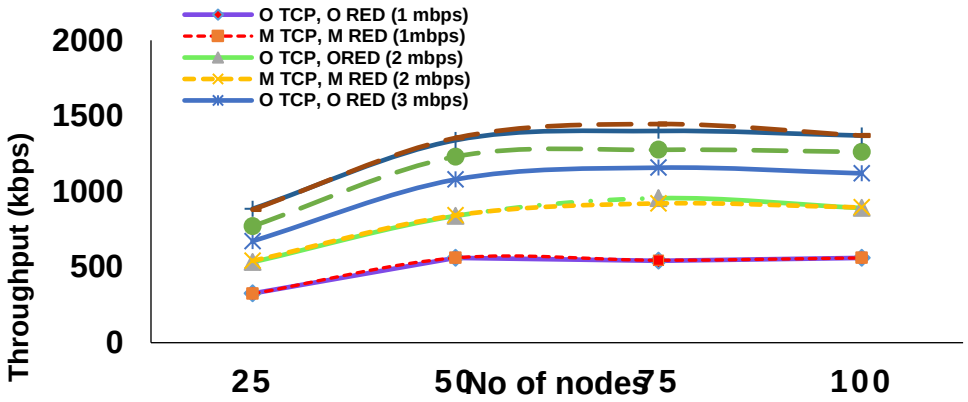


Fig 5. Throughput comparison of original TCPW and RED with modified TCPW and RED

Table and **Fig 5.** Throughput comparison of original TCPW and RED with modified TCPW and RED represents the throughput (in kbps) analysis for the original and modified TCPW plus the modified RED for varying no. of nodes (25, 50, 75, and 100) and data rates (1Mbps to 5Mbps).

Table 2. Comparison of Packet Delivery Ratio for Original TCPW and RED V/S Modified TCPW and RED

PDR (%)	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED	O-TCP & O-RED	M-TCP & M-RED
Bitrate / Nodes	1 Mbit/s		2 Mbit/s		3 Mbit/s		4 Mbit/s		5 Mbit/s	
25	0.9743	0.963	0.973	0.964	0.9689	0.9645	0.9752	0.9603	0.9726	0.9664
50	0.9988	0.9988	0.9957	0.9951	0.9937	0.9972	0.9951	0.9932	0.9943	0.9931
75	0.9901	0.998	0.9924	0.9918	0.9987	0.9987	0.9958	0.993	0.9908	0.989
100	0.9982	0.9982	0.9944	0.9943	0.9976	0.9978	0.995	0.9961	0.9968	0.9968

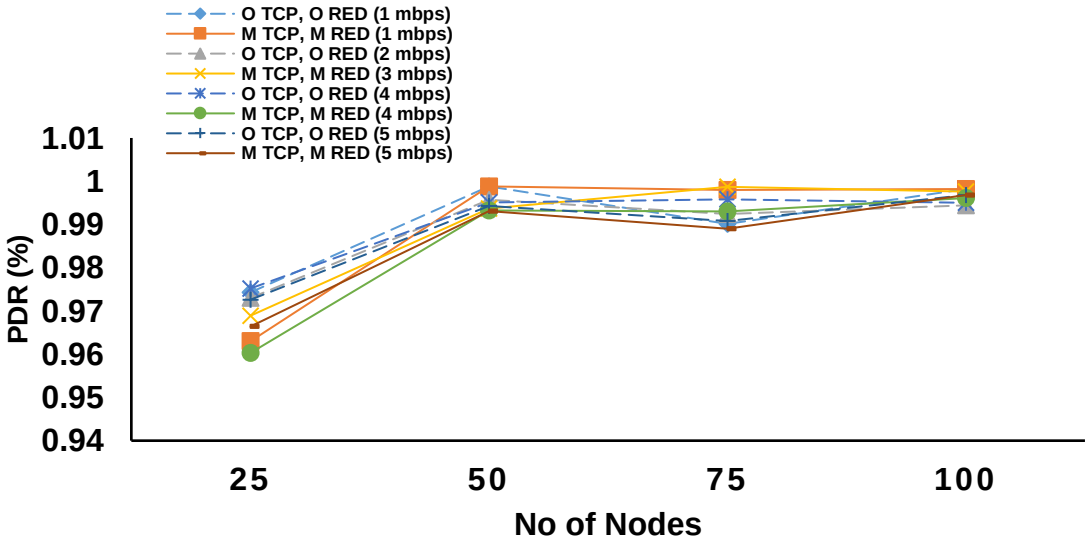


Fig 6 Packet Delivery Ratio comparison of original TCPW and RED with modified TCPW and RED

Table 2 and **Fig 6** represent the Packet Drop Ratio (PDR in percentage %) analysis for the original and modified TCPW plus the modified RED for varying no. of nodes (25, 50, 75, 100) and data rates (1Mbps to 5Mbps).

Table 3. Goodput comparison for original TCPW and RED with Modified TCPW and RED

Goodput	O TCP & O RED	M TCP & M RED	O TCP & O RED	M TCP & M RED	O TCP & O RED	M TCP & M RED	O TCP & O RED	M TCP & M RED	O TCP & O RED	M TCP & M RED
Bitrate	1 Mbit/s		2 Mbit/s		3 Mbit/s		4 Mbit/s		5 Mbit/s	
Nodes										
25	314.52	312.49	514.61	522.77	648.89	649.75	742.79	744.56	855.2	850.08
50	541.88	541.88	809.8	813.97	1042.9	1048.8	1190.57	1188.86	1293.85	1308.19
75	524.04	526.65	923.05	889.81	1118.12	1118.12	1276.5	1232.25	1352.71	1397.04
100	541.7	541.7	860.52	863.98	1081.92	1082.19	1215.76	1219.01	1323.27	1323.27

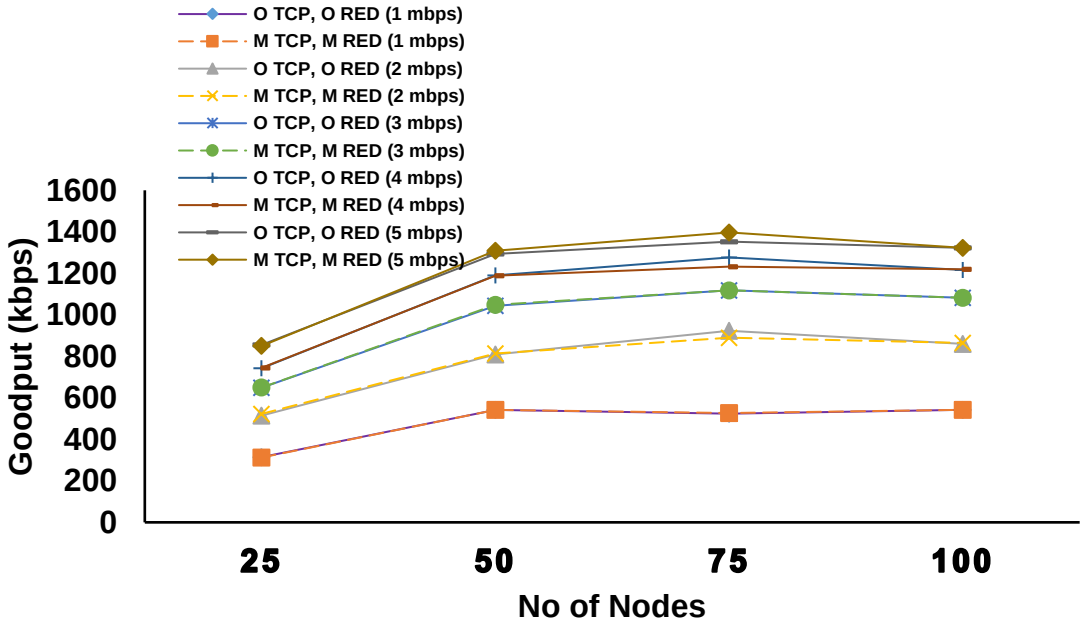


Fig 7. Goodput comparison for original TCPW and RED with Modified TCPW and RED

Table 3. Goodput comparison for original TCPW and RED with Modified TCPW and RED

Table 3 and **Fig 7** represent the Goodput (in kbps) analysis for the original and modified TCPW plus the modified RED for varying no. of nodes (25, 50, 75, and 100) and data rates (1Mbps to 5Mbps).

The results confirm that the suggested method enhances throughput and goodput while reducing the Packet Drop Ratio at greater data rates, in both M-TCPW with M-RED scenarios, when compared to the original TCP and RED protocols. Nevertheless, when the quantity of nodes grows, the impact of congestion becomes more noticeable, resulting in a minor reduction in throughput. The node architecture consisting of 50 nodes exhibits the maximum throughput in both processes.

The results demonstrate the efficacy of M-TCPW and M-RED in effectively managing congestion and optimizing

throughput and goodput, while also reducing the Packet Drop Ratio in the network configuration provided.

4 Conclusion

The results confirm that the suggested method improves throughput and goodput while lowering the Packet Drop Ratio at higher data rates, in both M-TCPW and M-RED scenarios, compared to the original TCP and RED protocols. However, as the total number of available nodes in the network increases, congestion has a bigger effect, causing a small drop in throughput. The node architecture with 50 nodes has the highest throughput in both processes.

References

[1] M. Patel, N. Tanna, P. Patel, and R. Banerjee, "TCP over Wireless Networks: Issues, Challenges, and Survey of Solutions," *Univ. Texas, Dallas*, 2001.

[2] R. Doshi, K. K. Hiran, B. P. Doppala, and A. K. Vyas, "Deep belief network-based image processing for local directional segmentation in brain tumor detection," *J. Electron. Imaging*, vol. 32, no. 6, p. 62502, 2023.

[3] H. Abdel-Jaber, "An exponential active queue management method based on random early detection," *J. Comput. Networks Commun.*, vol. 2020, pp. 1–11, 2020.

[4] M. Gerla, M. Y. Sanadidi, R. Wang, A. Zanella, C. Casetti, and S. Mascolo, "TCP Westwood: Congestion window control using bandwidth estimation," *Conf. Rec. / IEEE Glob. Telecommun. Conf.*, vol. 3, pp. 1698–1702, 2001, doi: 10.1109/glocom.2001.965869.

[5] H. Khelifi *et al.*, "Named Data Networking in Vehicular Ad Hoc Networks: State-of-the-Art and Challenges," *IEEE Commun. Surv. Tutorials*, vol. 22, no. 1, pp. 320–351, 2020, doi: 10.1109/COMST.2019.2894816.

[6] S. Hassan and A. Rufai, "Modified dropping-random early detection (MD-RED): a modified algorithm for controlling network congestion," *Int. J. Inf. Technol.*, vol. 15, no. 3, pp. 1499–1508, 2023.

[7] C. Casetti, M. Gerla, S. Mascolo, M. Y. Sanadidi, and R. Wang, "TCP Westwood: End-to-End Congestion Control for Wired/Wireless Networks," *Wirel. Networks*, vol. 8, no. 5, pp. 467–479, 2002, doi: 10.1023/A:1016590112381.

[8] Y. Patidar, M. Jain, and A. K. Vyas, "Optimal Stable Cluster Head Selection Method for Maximal Throughput and Lifetime of Homogeneous Wireless Sensor Network," *SN Comput. Sci.*, vol. 5, no. 2, p. 218, 2024, doi: 10.1007/s42979-023-02459-9.

[9] M. N. I. Khan, R. Ahmed, and M. T. Aziz, "A survey of TCP Reno, new Reno and sack over mobile ad-hoc network," *arXiv Prepr. arXiv1205.3125*, 2012.

[10] M. Ahmad, A. Bin Ngadi, A. Nawaz, U. Ahmad, T. Mustafa, and A. Raza, "A survey on TCP CUBIC variant regarding performance," in *2012 15th International Multitopic Conference (INMIC)*, 2012, pp. 409–412.

[11] W. Hua and G. Jian, "Analysis of TCP BIC congestion control implementation," in *2012 International Conference on Computer Science and Service System*, 2012, pp. 781–784.

[12] S. Hagag and A. El-Sayed, "Enhanced TCP Westwood Congestion Avoidance Mechanism (TCP WestwoodNew)," *Int. J. Comput. Appl.*, vol. 45, no. 5, pp. 21–29, 2012, [Online]. Available: https://research.ijcaonline.org/volume45/number5/pxc3879071.pdf.

[13] D. Zala and A. Vyas, "Development of RL-Based Adaptive Congestion Control Algorithm for Wireless Networks," in *2023 International Conference on Ambient Intelligence, Knowledge Informatics and Industrial Electronics (AIKIIE)*, 2023, pp. 1–8.

[14] L. Lenzini, E. Mingozzi, and G. Stea, "Tradeoffs between low complexity, low latency, and fairness with deficit round-robin schedulers," *IEEE/ACM Trans. Netw.*, vol. 12, no. 4, pp. 681–693, 2004.

[15] M. Bachl, "Fair Queuing Aware Congestion Control," *arXiv Prepr. arXiv2206.10561*, 2022.

[16] G. White and R. Pan, "Active Queue Management (AQM) Based on Proportional Integral Controller Enhanced (PIE) for Data-Over-Cable Service Interface Specifications (DOCSIS) Cable Modems," 2017.

[17] N. Khademi, G. Armitage, M. Welzl, S. Zander, G. Fairhurst, and D. Ros, "Alternative backoff: Achieving low latency and high throughput with ECN and AQM," in *2017 IFIP Networking Conference (IFIP Networking) and Workshops*, 2017, pp. 1–9.

[18] A. K. Vyas, R. K. Gangwar, and S. Kumar, "Elliptical air hole PCF-based low-cost sensor for refractive index and temperature detection: Design and analysis," *Opt. Fiber Technol.*, vol. 73, p. 103060, 2022.

[19] D. G. Raimagia and C. N. Chanda, "A novel approach to enhance performance of Linux-TCP Westwood on wireless link," in *2013 Nirma University International Conference on Engineering (NUiCONE)*, 2013, pp. 1–6, doi: 10.1109/NUiCONE.2013.6780126.

[20] S. B. Zhang, L. Gang, and K. Jun, "Study of congestion control algorithm based on neural network supervised control," *Appl. Res. Comput.*, vol. 27, no. 2, pp. 657–660, 2010.

[21] V. Jacobson, "Congestion avoidance and control," *Symp. Proc. Commun. Archit. Protoc. SIGCOMM*

1988, no. 1, pp. 314–329, 1988, doi: 10.1145/52324.52356.

[22] D. D. Zala and A. K. Vyas, "Comparative Analysis of RED Queue Variants for Data Traffic Reduction Over Wireless Network," in *Recent Advances in Communication Infrastructure*, 2020, pp. 31–39.

[23] W. Wu, Y. Ren, and X. Shan, "Stability analysis on active queue management algorithms in routers," in *MASCOTS 2001, Proceedings Ninth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 2001, pp. 125–132.

[24] C. Pan, S. Zhang, C. Zhao, H. Shi, Z. Kong, and X. Cui, "A novel active queue management algorithm based on average queue length change rate," *IEEE Access*, vol. 10, pp. 75558–75570, 2022.

[25] S. Ryu, C. Rump, and C. Qiao, "Advances in Active Queue Management (AQM) Based TCP Congestion Control," *Telecommun. Syst.*, vol. 25, no. 3, pp. 317–351, 2004, doi: 10.1023/B:TELS.0000014788.49773.70.

[26] E. Plasser, T. Ziegler, and P. Reichl, "On the Non-linearity of the RED Drop Function," *Peter Reichl*, p. 209, 2002.

[27] K. K. Chandulal, "A Survey On Red Queue Mechanism For Reduce Congestion In Wireless Network," *Int. Res. J. Eng. Technol.*, vol. 5, no. 1, pp. 99–103, 2018.

[28] M. Kwon and S. Fahmy, "Comparison of load-based and queue-based active queue management algorithms," in *Quality of Service over Next-Generation Internet*, 2002, vol. 4866, pp. 35–46.

[29] K. Zhou, K. L. Yeung, and V. O. K. Li, "Nonlinear RED: A simple yet efficient active queue management scheme," *Comput. Networks*, vol. 50, no. 18, pp. 3784–3794, 2006.

[30] J. Pan and R. Jain, "A survey of network simulation tools: Current status and future developments," *Email jp10@ cse. wustl. edu*, vol. 2, no. 4, p. 45, 2008.