Journal of Computing and Communication Vol.3 , No.1, PP. 33-43 , 2024

Enhancing the TCP Newreno Fast RecoveryAlgorithm on 5G Networks

Saleh M. Abdullah *[a], Mohamed S. Farag [a], Hatem Abdul-Kader[b], S. E. Abo-Youssef [a]

[A] *Department of Mathematics and Computer Science, Faculty of Science, Al-Azhar University, Egypt*

[b] *Department of Information Systems, Faculty of Computers and Information, Menoufia University, Egypt*

*Corresponding Author: Saleh M. Abdullah [***alraeesaleh15@gmail.com***]

ARTICLE DATA

Article history:
Article history:
Received 21 September 2023
Revised 09 December 2023
Accepted 09 December 2023
Available online

Keywords: TCP, Fast Recovery, 5G networks, Newreno.

.

ABSTRACT

Fifth-generation (5G) networks provide high-speed data transmission service. Therefore, relying on effective congestion control methods is one of the secrets to the success of the network's performance. One of the key functions of Transmission control protocol (TCP) is congestion control, which attempts to limit the TCP connection to its fair share of network bandwidth. TCP should have the ability to adapt to network conditions and channel fluctuations to get highly reliable and efficient data communications. This paper presents an adjusted fast recovery algorithm to enhance the performance of TCP-Newreno. The main idea behind this modification is to adjust the congestion window (cwnd) based on evaluating the round-trip time (RTT) changes to adapt to the dynamic channel conditions rather than reducing it to half of its value when network congestion is detected. In 5G networks, channel conditions tend to change frequently. This adjustment increases network throughput, reduces packet loss, and decreases delay.

1. Introduction

5G networks represent the latest generation of communications, offering quicker and more reliable connections on smartphones and other devices than ever before. This advancement enables a smarter and more interconnected world [1].

The congestion control (CC) mechanism used by TCP significantly impacts how well it performs. Among the widely used TCP versions on the internet, TCP-Newreno stands out for its ability to recover from multiple losses within a single loss window. This reduces the need for frequent retransmission timeouts, enhancing overall efficiency [2]. Therefore, if outages occur on a specific time scale, they may not work well in our environment. These observations have motivated us to design novel congestion control procedures to better adapt to the new radio environment. RTT is an important metric when it comes to TCP performance. It calculates the time it takes for a packet to travel from sender to recipient and back. In wireless networks, the RTT can be significantly higher than in wired networks due to factors such as signal propagation delays and network congestion. As a result, TCP may interpret these delays as a sign of network congestion and reduce its transmission rate, even when there is no actual congestion. This can lead to degraded performance and longer download times [3]. In the fast recovery algorithm of TCP congestion control:

The cwnd grows more slowly than it does during the slow start phase. This slower growth rate is due to TCP attempting to recover from packet loss, so it is being more cautious in its transmission rate.

TCP uses a "fast retransmit" mechanism to quickly retransmit lost packets without waiting for a retransmission timeout. While this can help recover more quickly from packet loss, it can also result in more duplicate packets being transmitted, further slowing down the flow convergence rate and increasing the time duration to complete the data flow [4]. It is important to carefully design and tune TCP congestion control algorithms to optimize performance in different network conditions [5].

In this study, we propose to develop an efficient CC mechanism to enhance TCP-Newreno's congestion control during fast recovery phases and implement it over a 5G network.

The rest of this paper is organized as follows: Section 2 describes the Background of 5G and TCP. Section 3 describes related works. In Section 4 we illustrate the steps of development TCP-E-newreno in the fast recovery algorithm. Section 5 evaluates performance and discusses the results. Finally, the conclusion ends the paper.

2. Background

2.1 Overview of 5G Networks

Several countries and regions worldwide have initiated research on the requirements and technologies of 5G communication. Research into 5G communication technology began as early as 2012 in some countries and regions worldwide. The European Commission launched a research initiative called the 5G Infrastructure Public-Private Partnership (5G PPP), which aimed to develop and deploy a 5G network infrastructure in Europe by 2020. The 5G PPP brought together industry leaders, academic institutions, and government organizations to collaborate on research and development efforts and establish a roadmap for deploying 5G networks [6].
Creating workable solutions for various vertical industries, such as the automotive, healthcare, transportation, and utility sectors, is one of the key goals of 5G.
5G aims to address six key issues. These issues include:

1. Lower end-to-end delay: 5G networks seek to cut end-to-end delay significantly. This is crucial for real-time communication-dependent applications like remote surgery and driverless vehicles.
2. Increased data rates: 5G networks aim to provide faster data rates, enabling users to download and upload large files more quickly and efficiently.
3. Higher capacity: The design of 5G networks allows for far more linked devices, which is critical for the growth of the Internet of Things (IoT).
4. Extensive-scale device connectivity: 5G networks aim to provide seamless connectivity for various devices, including mobile devices, IoT devices, and sensors.
5. Decreased costs: 5G networks aim to reduce the cost of wireless access and deployment, making it more affordable for consumers and businesses.
6. Consistent Quality of Experience (QoE): 5G networks aim to provide a consistent experience for users, regardless of their location or the number of devices connected to the network [7].
   To meet the performance criteria and specifications of 5G, it is important to address issues such as congestion that can impact end-to-end transmission performance, particularly at the start of data transmission.
   Measuring the RTT can be useful in addressing congestion and optimizing the performance of 5G networks.

Measuring the RTT can help determine the TCP-Newreno transmission rate and window size for a given network condition. By measuring the time it takes for a packet to travel from the sender to the receiver and back again, the RTT can indicate the network latency and help determine the appropriate transmission rate and window size to avoid congestion.
:

2.2 LTE-5G Tight Integration

LTE-5G tight integration is an important development in wireless technology that enables seamless connectivity and improves the performance of wireless networks. By integrating LTE and 5G networks, operators can provide faster, more reliable, and more efficient wireless connectivity to users and enable new use cases and applications that were not likely with previous generations of communication.
One of the fundamental benefits of LTE-5G tight integration is that it allows for the dynamic allocation of resources between LTE and 5G networks based on network conditions and user demand. For example, during periods of high network congestion, resources can be allocated to the 5G network to ensure that users experience fast and reliable connectivity. Conversely, during periods of low demand, resources can be shifted back to the LTE network to optimize resource utilization [8].

2.3 Congestion control mechanisms of TCP

End-to-end techniques optimize the network's performance as a whole without modifying individual layers or routes. This approach is often preferred because it is more flexible and scalable and can be applied to various networks and devices. The TCP protocol is an example of an end-to-end optimization technique widely used in wired and wireless networks. TCP uses congestion control algorithms to optimize the transmission rate based on

network conditions, allowing it to adapt to changes in the network and maintain reliable performance. However, the Additive Increase Multiplicative Decrease (AIMD) mechanism used in TCP can result in the unnecessary reduction of the cwnd and a decrease in throughput. Some techniques and algorithms can be used to mitigate this issue and improve TCP performance in modern networks, including 5G networks [9].

The mechanisms of TCP as slow start, congestion avoidance, fast retransmit, and fast recovery, are used to regulate the data flow between the sender and receiver and prevent network congestion Fig. 1 [10].

However, these variants differ in their reactions to packet losses [11]. For example, some variants use different algorithms for computing the RTT and the retransmission timeout (RTO) values, while others use different congestion control mechanisms. TCP-Newreno is similar to the original TCP protocol but uses a modified fast recovery algorithm that allows for more aggressive retransmission of lost packets. This helps reduce packet loss and improve network performance.

TCP-Tahoe [12], on the other hand, detects packet loss based on the reception of three duplicate ACKs or a timeout. During the slow start phase, the window size is exponentially increased. In the congestion avoidance phase, the window size is expanded linearly. Upon receiving a timeout or three duplicate ACKs, TCP-Tahoe retransmits, reduces the cwnd to one, and re-enters the slow start phase. The following is a summary of the congestion control algorithm employed by TCP-Tahoe.

If $cwnd < ssthresh$

$cwnd = cwnd + 1$

else

$cwnd = cwnd + \frac{1}{cwnd}$

For Timeout:

$ssthresh = \frac{cwnd}{2}$

$cwnd = 1$

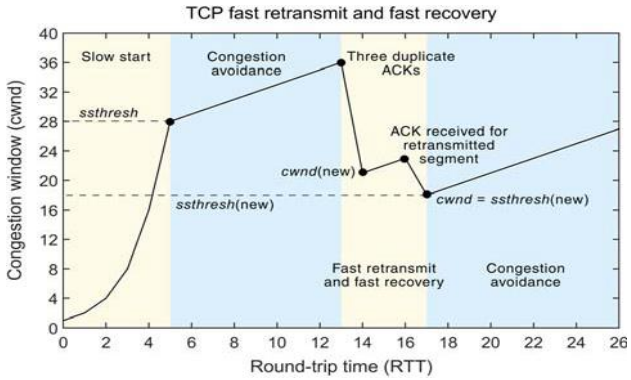


Figure 1. TCP fast retransmit and fast recovery algorithms

TCPVegas is a new implementation of TCP that utilizes the congestion avoidance mechanism to prevent packet loss by reducing its cwnd as soon as it detects impending congestion. In TCP-Vegas, cwnd is specified by the difference between expected and actual throughput [13]**.**

TCP-Newreno is a variant of Reno that incorporates specific changes to address an issue that leads to unnecessary timeouts due to multiple packet congestion [14]**.**

TCP-Westwood [15] uses aggressive end-to-end congestion control. Examining the frequency and pattern of ACKs sent back across the reverse links calculates the sender's side network bandwidth. It is noteworthy that it does not distinguish between various packet loss reasons.

3. Related Works

Landström et al. [16] combined a modified fast recovery algorithm and a variant of the appropriate byte counting (ABC) algorithm to enhance TCP bandwidth utilization when the acknowledgment (ACK) frequency is reduced to two or four per send window. However, these algorithms have limits because they only deal with a portion of the problems. Their approach led to significantly occupying router buffers without effectively mitigating traffic bursts. Additionally, neither of these solutions addressed the interference caused by delayed ACKs on round-trip timing. Thus, this solution did not tackle the interference problem on round-trip timing caused by delayed ACKs.

Saedi and El-Ocla [17] revisited the TCP Congestion Control Enhancement for Random Loss (CERL) mechanism, which is called new developed TCP variant CERL Plus (CERL+).

Its main idea is to use a dynamic threshold in terms of RTT. CERL+ is primarily a modification of the fast recovery mechanism of the Reno protocol, specifically focusing on cwnd expansion. As a result, there is no need for any additional overhead signaling in the CERL+ protocol. Unlike other algorithms, CERL+ does not introduce any extra overhead packets.

Raimagia et al. [18] present a novel method to enhance TCPW's performance by distinguishing between packet loss problems brought on by congestion or bit error. To differentiate between the problems of congestion loss and bit error loss, modified TCPW uses the flag indication of the TCP header.

Azzino et al. [19] proposed cross-layer-based TCP (X-TCP), a technique used to improve the performance of TCP in wireless networks. X-TCP uses a control loop to adjust the TCP cwnd based on the estimated RTT and the buffer occupancy level. By doing so, X-TCP can prevent the buffer from becoming filled with packets and ensure that packets are delivered promptly. This can improve network performance and reduce delays and latency in the network.

Xing, Xiaolin, et al.[20] Continue to use packet loss as a congestion control indicator rather than delay. However, we can choose between random loss or network congestion based on how other network metrics, such as RTT, are perceived and implement various coping mechanisms.

The proposed approach by Zong, Liang, et al. [21] improves Veno's fast recovery algorithm using multilevel differentiation for differentiating between distinct data losses. The upgraded TCP strategy suggested in this study performs better than the other three schemes in the simulation test while dealing with random and congestion data loss. Considered are networking scenarios where there is only random packet loss and instances where there are both random and congestion losses. The simulation results show that, in comparison to the other three TCP variations, our suggested technique performs better in the two networking scenarios.

4. Methodology

Due to the large bandwidth and low latency of 4G and 5G networks, TCP must respond fast to shifting network conditions, such as congestion. By enabling more data to be transferred in each RTT and consequently boosting throughput, increasing the TCP cwnd size can enhance TCP performance. To keep the network stable, congestion control mechanisms must be enhanced.

The following explains the proposed TCP-E-new reno during the Enhanced Fast Recovery Algorithm (EFRA).

The issue with the existing fast recovery algorithm is that, regardless of the network's state, the TCP sender cuts in half its cwnd as soon as congestion detection occurs.

The core of the enhanced fast recovery concept is to reduce the cwnd based on the network's condition to enable sending more packets to the destination. Using the change in RTT, the sender can estimate the network's congestion level and adjust the cwnd as necessary. Since every packet travels in a round from the sender to the receiver and back, the RTT will alter as the network changes.

As a result, while using the fast recovery algorithm, the sender can recognize changes in RTT and reduce the cwnd by a quantity corresponding to those changes.

The proposed mechanism consists of two steps:

1) Calculating the RTT Change

The sender keeps track of the last N RTT values while continuously monitoring the RTT. During the fast recovery algorithm of TCP, when the sender receives duplicate acknowledgments from the receiver, the sender uses the queued RTT information to calculate the average RTT. This information is used to adjust the cwnd:

$$RTTavrage = RTT1 + RTT2 + \dots + \frac{RTTn}{n} \quad (1)$$

Then, the TCP sender computes the difference in RTT ($diffRTT$) as the difference between the latest RTT ($RTTn$), which is typically performed just before congestion is detected by the sender. This is because the change in RTT can be used as an indicator of network congestion, which can then be used to adjust the sending rate of packets, manage congestion, and the average RTT as:

$$RTTdiff = RTTn - RTTavrage \quad (2)$$

The calculation of ($diffRTT$) just before congestion is detected is an important technique for managing network congestion and ensuring the reliable delivery of packets.

2) Change the cwnd

The sender computes the increasing sending rate factor by dividing the current cwnd size by the average RTT. This calculation determines how much the sender can increase its sending rate without causing further congestion in the network. We denote it with a β.

$$\beta = (cwnd/RTT_{avrage}) \quad (3)$$

TCP-E-newreno decreases the sending rate by a factor called F.

The main facet establishing the value for F is the change in the RTT.

$F = \beta * RTT_{diff}$ (4)

The value of F represents the number of packets that should be removed from the cwnd to avoid congestion. By decreasing the cwnd in this way, TCP can ensure that it does not send more packets than the network can handle, which helps to avoid congestion and ensure reliable delivery of packets. As a result, F is used to adjust the cwnd as the new congestion window ($cwnd_{-new}$) in the equation (5)

$cwnd_{-new} = max\,(2,\ (cwnd - F))$ (5)

$ssthresh\ = \ max\,((cwnd_{-new}), 2)$ (6)

$cwnd\ = ssthresh + 3$ (7)

The sender initially determines the $cwnd_{-new}$ as in Eq. 5.

The sender then tunes the cwnd to the value of the ssthresh plus three or the quantity of duplicate acknowledgments received. For each duplicate acknowledgment received, the sender increments the cwnd by one and, if allowed, sends a new segment. When there is only a partial ACK, the sender continues and retransmits the segment after the Acked one. When there is full ACK, it activates the fast recovery and sets the cwnd to the ssthresh.

Steps of EFRA:

Invoke Fast Recovery

1-when-3 DUPACKs:

$cwnd_{-new} = max\{2, (cwnd\ -\ F)\};$

$ssthresh = max\{2, cwnd_{-new}\};$

$cwnd\ = ssthresh + 3;$

If DUPACK is received,

$cwnd\ + +;$

IF partial Ack;

Remain in Fast Recovery;

Retransmit the next lost packet (per RTT);

IF Full Ack:

$cwnd\ = ssthresh;$

Exit Fast Recovery;

call CAA;

IF TimeOut:

$cwnd_{-new} = \max\{2, (\text{cwnd} - F)\};$

$\text{ssthresh} = \max\{2, cwnd_{-new}\};$

$\text{cwnd} = 1;$

Implement Slow Star.

5. Performance Evaluation and Results Analysis

Wireless devices and networks have become increasingly popular in recent years, all sharing the same goal of delivering good performance, whether throughput growth or adherence to Quality of Service (QoS) standards, like acceptable throughput under high loss probability and low delay. However, since wireless networks have different characteristics than wired networks, protocols designed for wired networks must be carefully examined to ensure their effectiveness over wireless channels. Thus, it is important to evaluate the usability of these protocols over wireless networks. In this context, we will investigate the performance of TCP-E-newreno over 5G networks using the NS2 simulator. 5G NR will utilize the same mobile networking principles and support operation in two frequency ranges: FR1, below 7,125 MHz, and based on orthogonal frequency division multiplexing (OFDM). To incorporate the new protocol TCP-E-newreno into the NS-2 simulator, specific and intricate adjustments need to be made to the source code. The procedures involved in integrating this modification into NS-2 are meticulous and complex.

Effectively, the performance of the new TCP relies on the benchmark set by TCP-Newreno, adopting the same architecture and control mechanism in congested situations. However, this TCP will undergo modifications to alter its congestion control approach from the traditional AIMD.

NS-2 offers a wide range of features that facilitate the simulation of wired and wireless TCP, routing, and multicast protocols. It supports numerous protocols, enabling the testing of complex scenarios time-efficiently. Due to its robustness and modular design, NS-2 has gained significant popularity within the networking research community.

The Tool Command Language (TCL) is utilized in NS-2 to initiate the event scheduler, establish the network topology, and instruct traffic sources on when to commence and cease packet delivery during protocol simulations. Running a specifically tailored TCL program yields the desired output at the end of the simulation.

The evaluation criteria considered for this assessment were throughput, packet loss, and delay. TCP-E-newreno, which considers both traffic intensity and varying signal-to-interference ratios, accurately estimates the bandwidth to control the cwnd, making it well-suited for 5G networks. The results were presented in two manners. Firstly, a comparison was made between TCP-Newreno and TCP-E-newreno. Secondly, the evaluation was extended to include TCP-Vegas, TCP-Taho, and TCP-Westwood, alongside TCP-Newreno.

5.1 Experiment setup

We conducted a system performance evaluation based on the configuration parameters presented in Table 1, focusing on throughput, delay, and packet losses. We performed simulations using TCP-E-newreno to achieve our proposal, which doesn't require the sender or receiver modification. This is because TCP operates at the transport layer of the network stack and is designed to be independent of the underlying network infrastructure. This feature allows the proposed algorithm to be easily applied to the existing network infrastructure without requiring significant changes. We implemented and tested TCP-E-new reno in emulated networks to assess its performance in real-world scenarios. Our research aims to contribute to the ongoing efforts to enhance the performance of network systems and improve the user experience.

Table 1 Simulation Parameter Configuration

Parameter	Value
Simulation time	60 seconds
QoS feature	True
Multicast feature	True
Air interface latency	2 ms
Core network latency	2 ms
Remote host	100 ms
Errormodel	DataErrorModel
TCP variants	TCP-E-newreno,TCP-Newreno, TCP-Taho, TCP-Vegs, TCP-Westwood
packet size of background traffic	1500 bytes

Fig. 2 presents an overview of the 5G NS2 simulator, depicting multiple User equipment (UEs) in communication with the Base Station (BS), Core Network, and Remote Server.

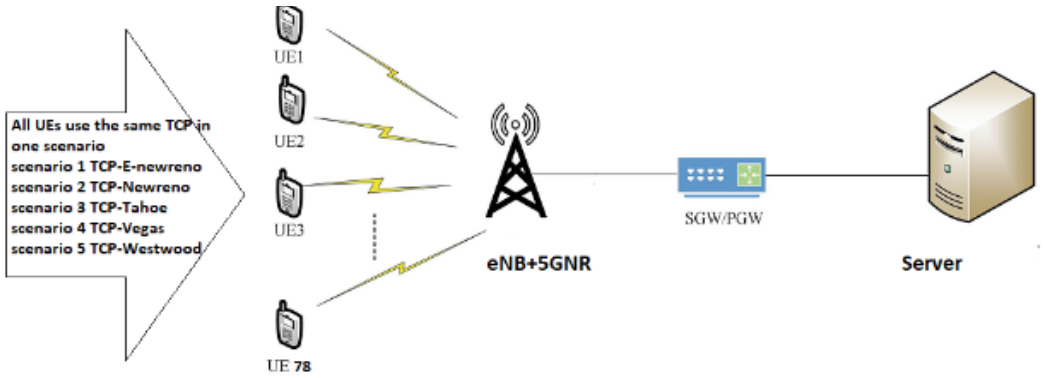


Figure 2. Model of simulation

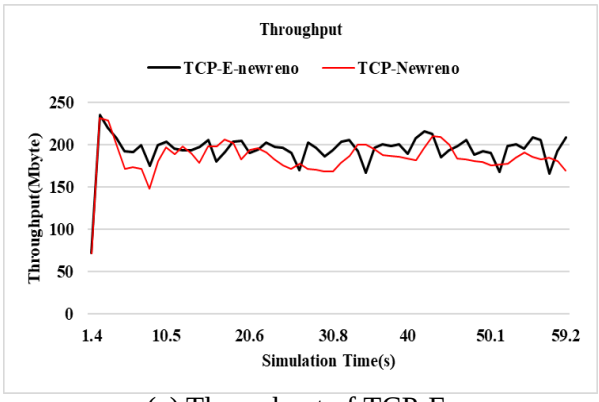
5.2 Analysis Discussion Results

5.2.1 Throughput

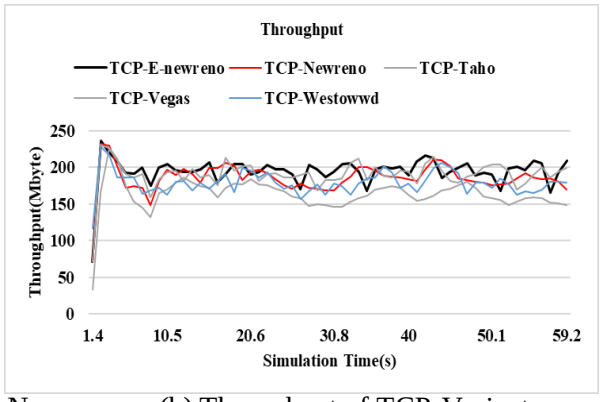
Throughput is the quantity of data a network can move from a sender to a receiver during a specified time, usually measured in seconds. Throughput is typically expressed in kilobits or megabits per second. Thanks to its enhanced Fast recovery algorithm, TCP-E-newreno achieves a higher throughput than TCP-Newreno.

This algorithm reduces the recovery period, enabling TCP-E-newreno to transmit more packets based on the network's condition, as illustrated in Fig. 3(a). In contrast, TCP-Newreno follows a similar approach by waiting for all lost packets to be recovered before sending a few new packets. However, during the fast recovery algorithm, TCP-Newreno reduces its cwnd by half and continuously checks the network until the error is resolved and an unduplicated ACK is received.

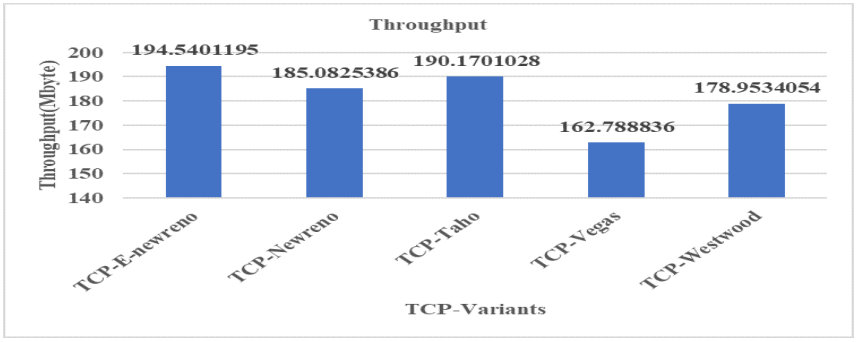
While the TCP-Westwood and TCP-Newreno algorithms are comparable, TCP-Westwood's bandwidth estimate is too cautious, limiting the cwnd excessively and making it difficult to restore throughput quickly. This makes it less effective than TCP-Newreno. Through the alteration in RTT, Vegas determines the link state and sets the cwnd. In the slow start stage, Vegas also uses a two-stage RTT reciprocating growth method, where the first stage sees exponential development and the second stage sees changes in RTT. When the bandwidth was constrained, we observed that the setting of cwnd in TCP-Westwood was almost identical to that in TCP-Taho. Therefore, TCP-Westwood and TCP-Taho cannot keep up with the immediate dynamics of the available bandwidth and cannot utilize it effectively, as shown in Fig. 3 (b) and (c). It is revealed that TCP-E-newreno outperformed other algorithms, followed by TCP-Taho and TCP-Newreno, while TCP-Vegas performed worst.



(a) Throughput of TCP-E-newreno against TCP-Newreno



(b) Throughput of TCP-Variant



(c) Average Throughput of TCP-Variant

Figure 3: (a) Throughput of TCP-E-newreno against TCP-Newreno (b) Throughput of TCP-Variant

5.2.2 Delay

Delay measurement is important in TCP/IP networks as it governs the transmission behavior, where a computer sends a restricted data volume to its intended destination and awaits an ACK before proceeding with further transmission. Nevertheless, in Millimeter wave (mmWave) networks, rapid shifts from Line of Sight (LOS) to Non-Line of Sight (NLoS) conditions result in substantial fluctuations in transmission rates. This presents a challenge for conventional congestion control protocols such as TCP-Newreno, as they cannot discern between losses caused by network congestion and those arising from deteriorating channel quality. This particular issue is further compounded in mmWave networks. Therefore, it is appropriate to consider adjusting the cwnd based on changes in the RTT to adapt to the mmWave band. TCP-E-newreno outperforms TCP-newreno in terms of performance, as seen in Fig. 4. However, TCP-E-newreno shows the second-smallest latency, as we can see that our results are consistent with the results reported in [22].

Additionally, TCP-Vegas, a delay-based protocol, provides less delay than other variants. This is because Vegas initially carefully increases the sending rate, setting the ssthresh to only two packets and adjusting its cwnd linearly founded on the difference between the current RTT and the minimum RTT. As mentioned earlier, TCP-Vegas effectively minimizes delay by continuously maintaining the RTT at a low level. On the other hand, TCP-E-new reno attempts to utilize available network resources by increasing the sending rate.

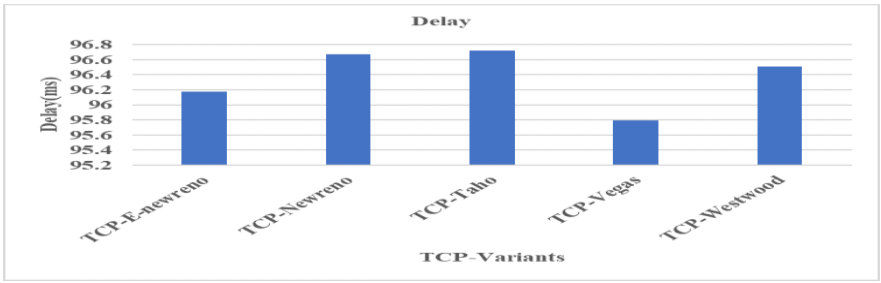
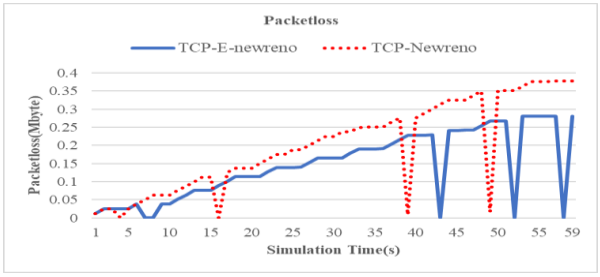
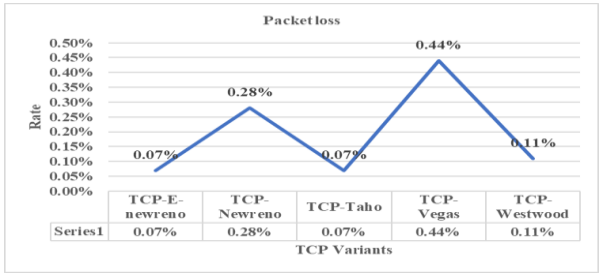


Figure 4: delay of TCP-E-newreno against TCP-Newreno and average delay of TCP-Variant

5.2.3 Packet loss

The error model is employed to examine the impact of the wireless environment on the proposed NS-2. The ErrorModel is utilized as the selected model, and the random variable/Uniform is deployed as the random variable. Additionally, Agent/Null is configured to manage any dropped targets. Fig. 5 (a) shows that TCP-E-Newreno can effectively reduce packet loss compared to TCP-Newreno. Furthermore, greater improvements are achieved when the number of busy users reaches 100. This is primarily due to a channel with an increased bit error rate (BER), which results in increased packet retransmissions and subsequent buffering of packets at the base station (BS). In addition, TCP-E-new reno dynamically adjusts its congestion window size based on changes in network load. It adapts its transmission rate during the fast recovery algorithm, effectively reducing the likelihood of packet loss. Fig. 5 (b) illustrates the advantages of employing TCP-E-newreno over existing algorithms such as Tahoe, Westwood, and Vegas.





(a) **Paket loss of TCP-E-newreno against TCP-Newreno** (b) **Paket loss of TCP-Variants**

Figure 5: (a) Paket loss of TCP-E-newreno against TCP-Newreno (b) Paket loss of TCP-Variants

5.2.4 Fairness

The fairness metric in network engineering assesses whether users or applications receive an equitable distribution of system resources. Congestion control algorithms that perform well with current TCP versions are crucial for

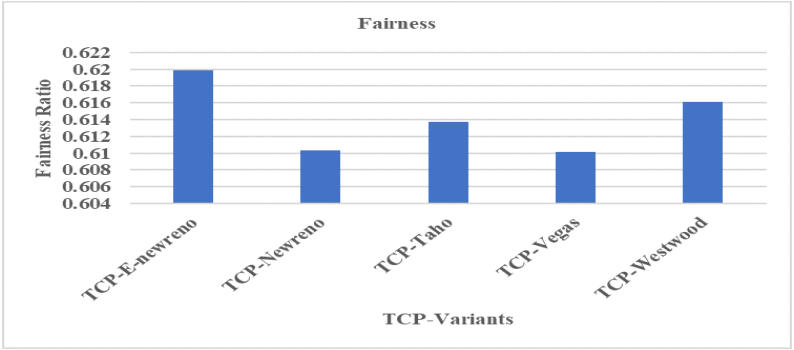


Figure 6: Fairness Rate

Peer-to-peer applications and new network transmission protocols. Fairness can be defined using various mathematical and conceptual approaches, including the Jain's Fairness Index (JFI) [22]. The fairness comparison of the proposed TCP-E-new reno with the existing algorithms is analyzed and presented. The fairness index from [23] is utilized to evaluate the typical fairness behavior of the proposed TCP-E-newreno algorithm. In Fig. 6, a comparison is shown between the fairness index of TCP-E-newreno and that of TCP-Newreno, TCP-Westwood, TCP-Tahoe, and TCP-Vegas algorithms. The graph illustrates that the fairness index of the TCP-E-newreno algorithm is the highest among all the algorithms, indicating its superior fairness. Moreover, this outcome acknowledges the algorithm's ability to achieve the highest throughput while efficiently utilizing network resources.

6. Conclusion

In mixed wired and wireless networks, effectively utilizing the available bandwidth is essential. Various approaches have been proposed in the dynamic internet environment to estimate available bandwidth and manage cwnd. This paper describes a current TCP-E-newreno implementation over 5G networks. The approach entails altering the TCP sender's cwnd following the degree of network congestion, which the RTT establishes as a gauge of network traffic volumes. By detecting changes in RTT, the sender adjusts the cwnd accordingly, reducing it by an appropriate amount.

TCP-E-newreno, in its fast recovery algorithm, employs a different approach to control the transfer rate than traditional TCP. While traditional TCP cuts the cwnd in half upon receiving a partial ACK during fast recovery, TCP-E-new reno maintains a constant cwnd. It only reduces it when multiple packet losses occur within the same window.

TCP-E-new reno offers a notable enhancement to the transmission performance in wireless networks that often face random losses not associated with congestion. Its remarkable capability to swiftly amplify the transfer rate during packet losses caused by frequent link faults rather than congestion plays a pivotal role in this improvement. Through a series of comprehensive simulations, we have convincingly demonstrated that TCP-E-newreno surpasses the performance of the other four implementations across multiple critical metrics, including throughput, packet loss, fairness, and delay. These simulations were conducted extensively to validate TCP-E-newreno's superior performance.

References

[1] J. McCann, M. Moore, and D. Lumb, "5G: everything you need to know," Techradar, 2019

[2] M. Panda, H. L. Vu, M. Mandjes, and S. R. Pokhrel, "Performance analysis of TCP NewReno over a cellular last-mile: Buffer and channel losses," IEEE Transactions on Mobile Computing, vol. 14, no. 8, pp. 1629-1643, 2014.

[3] M. Allman, V. Paxson, and E. Blanton, "TCP congestion control," 2070-1721, 2009.

[4] M. Zhang et al., "Transport layer performance in 5G mmWave cellular," in 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016: IEEE, pp. 730-735.

[5] I. Petrov and T. Janevski, "Design of novel 5G transport protocol," in 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), 2016: IEEE, pp. 29-33.

[6] Y Y. Cheng, F. Cheng, B. Deng, J. Li, and C. Mei, "ARQ Algorithm Optimization of Radio Link Control Layer in 5G System," in Proceedings of the 2nd World Symposium on Software Engineering, 2020, pp. 135-140..

[7] A. Gupta and R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," IEEE access, vol. 3, pp. 1206-1232, 2015.

[8] S. Ahmadi, 5G NR: Architecture, technology, implementation, and operation of 3GPP new radio standards. Academic Press, 2019

[9] J. Song, P. Dong, H. Zhou, T. Zheng, X. Du, and M. Guizani, "A performance analysis model of TCP over multiple heterogeneous paths for 5G mobile services," Sustainability, vol. 10, no. 5, p. 1337, 2018.

[10] J. Lorincz, Z. Klarin, and J. Ožegović, "A comprehensive overview of TCP congestion control in 5G networks: Research challenges and future perspectives," Sensors, vol. 21, no. 13, p. 4510, 2021.

[11] W. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," 2070-1721, 1997.

[12] V. Jacobson, "Congestion avoidance and control," ACM SIGCOMM computer communication review, vol. 18, no. 4, pp. 314-329, 1988.

[13] L. S. Brakmo, S. W. O'malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," in Proceedings of the conference on Communications architectures, protocols and applications, 1994, pp. 24-35.

[14] J. C. Hoe, "Start-up dynamics of TCP's congestion control and avoidance schemes," Massachusetts Institute of Technology, 1995.

[15] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," in Proceedings of the 7th annual international conference on Mobile computing and networking, 2001, pp. 287-297.

[16] S. Landström and L.-Å. Larzon, "Reducing the TCP acknowledgment frequency," ACM SIGCOMM Computer Communication Review, vol. 37, no. 3, pp. 5-16, 2007.

[17] H. El-Ocla, "TCP CERL: Congestion control enhancement over wireless networks," Wireless Networks, vol. 16, pp. 183-198, 2010.

[18] 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: IEEE, pp. 1-6.

[19] T. Azzino, M. Drago, M. Polese, A. Zanella, and M. Zorzi, "X-TCP: A cross layer approach for TCP uplink flows in mmWave networks," in 2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), 2017: IEEE, pp. 1-6.

[20] Xing, Xiaolin, et al. "Optimization of TCP Congestion Control Algorithms Loss-Based in Wireless Network." International Conference on Emerging Networking Architecture and Technologies. Singapore: Springer Nature Singapore, 2022.

[21] Zong, Liang, et al. "On Enhancing TCP to Deal with High Latency and Transmission Errors in Geostationary Satellite Network for 5G-IoT." Security and Communication Networks 2020 (2020): 1-7.

[22] Xiao, Kefan, Shiwen Mao, and Jitendra K. Tugnait. "TCP-Drinc: Smart congestion control based on deep reinforcement learning." IEEE Access 7 (2019): 11892-11904.

[23] R. Jain, The art of computer systems performance analysis: techniques for experimental design, measurement, simulation, and modeling. Wiley New York, 1991.

[24] Y.-J. Song, G.-H. Kim, and Y.-Z. Cho, "Bbr-cws: Improving the inter-protocol fairness of bbr," Electronics, vol. 9, no. 5, p. 862, 2020.