Improving the TCP Newreno Congestion Avoidance Algorithm on 5G Networks

Saleh M. Abdullah[1]*, Mohamed S. Farag[1], Hatem Abdul-Kader,[2] and S. E. Abo-Youssef [1]

[1] Al-Azhar University, Department of Mathematics and Computer Science, Faculty of Science, Cairo, Egypt;
Email: mo-hamed.s.farag@azhar.edu.eg (M.S.F.), abuyousf@hotmail.com (H.A.K.)
[2] Menoufia University, Information Systems, Faculty of Computers and Information, Menoufia, Egypt;
Email: hatem6803@yahoo.com (S.E.A.Y.)
*Correspondence: alraeesaleh15@gmail.com.

***Abstract*—Enhancing Transmission Control Protocol (TCP) performance is one of the key solutions for improving the performance of modern wireless communication networks. It is a highly dependable protocol for communication between network hosts based on an internet protocol (IP). It uses packet sequence numbering and acknowledgment (ACK) packets to ensure that data is recoverable in the event of problems like data loss, and corruption. Loss-based Congestion Control (CC) algorithms overreact and underperform in the presence of rapid channel oscillations, resulting in buffer bloat and significant delays. The goal of this paper is to develop an efficient congestion control mechanism for the new TCP agent Newreno over the fifth generation (5G) network, as well as to improve the increment and decrement strategy for adjusting the congestion window (cwnd) during the congestion avoidance phase, It also fast adapts to the current network requirements and adjusts the transmission rate accordingly. TCP enhancements can improve performance under dynamic traffic loads and over long product channels with high bandwidth. As a result, when such a TCP congestion control mechanism is improved, the network's throughput increases, and low packet losses, and delay are reduced.**

***Keywords*—TCP, congestion control, 5G networks, Newreno**

I. Introduction

5G networks are the most recent generation of mobile internet access. They provide faster and more dependable connections on smartphones and other devices than ever. They also offer faster connections with an average download speed of around 1 Gbps, providing the infrastructure required to transport massive volumes of data and allowing for a smarter and more connected world [1]. Therefore, high efficiency transmission protocols are required. In today's data networks, TCP is the amin protocol used to maintain reliable connections. The utilized Congestion Control (CC) technique has a significant impact on TCP performance. Over the last three decades, TCP CC algorithms have evolved, and a huge variety of CC algorithm modifications have been made to fit varied network situations. Here, TCP-Newreno is one of the TCP versions on the internet. It can recover from several losses within the same loss window, minimizing the need for frequent retransmission timeouts [2].

The standard congestion control algorithm TCP-Newreno isn't built to dynamically control or mitigate these losses. This results in unnecessary drops in the cwnd and in turn reduces network efficiency [3]. It also has a linear cwnd increase during the congestion avoidance stage. It is also noticed that, if outages occur on a specified time scale, it may not work well in our environment. These observations have motivated us to design novel congestion control approaches that are more adaptable to the network conditions. In applications that rely on short TCP connections, poor performance can be an issue [4]. There are two key reasons for typical TCP protocols' poor performance in high-speed networks:

1) One is that in which the cwnd linear growth by one packet per round trip time (RTT) is too slow in the congestion avoidance stage, and the multiplicative drop in the case of loss is too severe.
2) The second is that at the flow rate, keeping big average congestion windows needs a low balance loss probability and a stable flow dynamic design [5].

There is a variety of approaches to this problem, ranging from changing the sender's TCP congestion management algorithm to implementing Active Queue Management (AQM) at the base station.

In this paper, we propose an efficient congestion control mechanism for the new TCP agent Newreno during congestion avoidance phases and implement it over Long Term Evolution (LTE)- 5G network.

The paper is organized as follows: Begin with an outline of 5G and TCP in Section II. In Section III, describe related works, and then in Section IV, present steps of the development of TCP-Enewreno in the congestion

Manuscript received September 5, 2022; revised November 9, 2022, accepted November 28, 2022.

avoidance phase. As for section 5, it evaluates the performance and discusses the results. Finally, the conclusion contains the findings of this research paper.

II. Background

A. Overview of 5G Networks

Some countries and regions throughout the world began researching the 5G communication technology requirement and technologies in 2012 [6]. 5G's objectives include developing viable solutions for vertical industries including automotive, healthcare, transportation, and utilities. Besides, the Internet of Things (IoT) is a sophisticated, extensive network of wirelessly connected, individually identifiable objects that can communicate with one another through the Internet. Likewise in an IoT structure, the gadgets are frequently fitted with wireless sensors [7–9]. It should solve six issues that formerly released networks have not adequately solved, such as lower end-to-end delay, increased data rates, higher capacity, extensive-scale device connectivity, reduced costs, and consistent Quality of Experience (QoE) provisioning [10]. To meet the performance criteria as well as the 5G specifications, avoiding the congestion generated by a large burst of data at the start of transmission, which results in poor end-to-end transmission performance must be addressed too. This can be done by testing the available capacity of the network and increasing the window size.

B. LTE-5G Tight Integration

The next generation of mobile networks will combine new disruptive technology with the next evolution of current 4G networks, though breaking a disruptive technology into the market is never easy. So, it will make sense to use LTE networks as part of the upcoming 5G generation, as telecom operators have lately invested a lot of time and money in deploying them. For instance, 4G can provide a layer of coverage and make 5G networks more resilient to connection failures and service interruptions [11].

C. Congestion Control Mechanisms of TCP

TCP uses different mechanisms to avoid network congestion. The congestion control mechanism of TCP operates at the sender and is based on the cwnd. It includes two windows, a cwnd that imposes a constraint on the amount of data the sender can transmit into the network before receiving an ACK(Acknowledgement), and the receiver advertised window (rwnd), a receiver-side limit to the amount of outstanding data. The congestion control strategy of TCP consists of four algorithms: Slow start, congestion avoidance, fast retransmit and fast recovery [12]. We will first describe both the slow start and congestion avoidance algorithms as shown in "Fig. 1" [13]. When the three-way handshake is finished by TCP it bursts out as many packets as allowed by the agreed window size. This was not a major issue in small networks. Yet, as networks developed and the number of linked hosts increased, huge bursts became a source of trouble. Data began to accumulate faster than it could be transferred or received, causing congestion at network bottlenecks. The slow start algorithm initialized the cwnd to one segment, and each time ACKs are received, the cwnd is increased by the number of packets acknowledged, thus growing exponentially. However, the exponential increase of cwnd must be stopped to minimize network congestion. In small localized LANs, where the usual limitation is the window size, this is usually not an issue. Yet, with large WANs, there are many more servers that are expected to share the network capacity, and congestion is difficult to avoid if all hosts run at full capacity. When $cwnd > ssthresh$, the congestion avoidance algorithm is entered and cwnd is increased by approximately one segment per RTT and the algorithm continues its work until congestion is detected. It is recommended to increase cwnd as follows and adjust it each time an ACK is received:

$$cwnd += SMSS \times \frac{SMSS}{cwnd} \tag{1}$$

The sender's maximum segment size (SMSS) can be calculated using the network's highest transmission unit. The TCP/IP headers and options are not included in the size.

As for fast retransmit and fast fecovery, and based on arriving duplicate ACK, the sender should operate the fast retransmit algorithm to detect and repair the loss. After the fast retransmit sends what looks to be the missing segment, the fast recovery algorithm activates the congestion avoidance, rather than the slow start phase, allowing a greater throughput under moderate congestion, specifically for big windows. There are several implementations of TCP (Tahoe, Reno, Vegas, New Reno. …) here, but fundamentally the operation of TCP remains the same and is based on RFC 5681 [14].

In TCP-Tahoe the reception of three DUPACKs(Duplicate Acknowledgement), as well as a timeout, indicates packet loss [15]. The window size is raised exponentially throughout the slow start phase, while the window size is expanded linearly throughout the congestion avoidance phase. When it receives a timeout or three DUPACKs, it does quick retransmission, reduces the cwnd to one, and enters the slow start phase. The following is a summary of the congestion control algorithm. For each segment ACKed:

If $cwnd < ssthresh$

$$cwnd = cwnd + 1$$

else

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

For Timeout:

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

$$cwnd = 1$$

TCP-Reno is the traditional TCP. In wired networks, it ensures data transmission reliability. In wireless networks, however, it is forced to slow down in order to recover from numerous packet losses that occur. TCP-Newreno is a version of Reno that has some changes to repair what is basically a bug that repeatedly causes unnecessary timeouts in response to multiple packet congestion. This modification was first suggested by Janie Hoe [16].

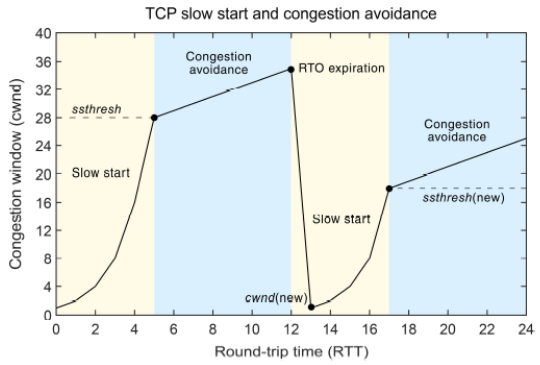

Figure 1. TCP slow start and congestion avoidance mechanism.

TCP-Westwood is wireless TCP utilizing end-to-end aggressive congestion control. It calculates the current network bandwidth at the sender side. The sender determines network bandwidth by using the rate and pattern of returning ACK through the reverse links. Because it does not distinguish the cause of packet loss [17].

III. Related Works

Shalunov *et al.* [18] created Low Extra Delay Background Transport (LEDBAT) as an experimental delay-based congestion control algorithm that seeks to utilize the available bandwidth on an end-to-end path while limiting the consequent increase in queueing delay on that path. It was designed for use by background bulk-transfer applications to be no more aggressive than standard TCP congestion control, thus limiting interference with the network performance of competing flows. Wang *et al.* [19] improved the network performance over large range of conditions and simultaneously maintained the fairness with TCP Cubic servers. Jiang and Jin [20] proposed an algorithm CLTCP(Congestion-Level-based TCP) that uses congestion levels for controlling the virtual parallel flow numbers in TCP connection. Moreover, Gwak and Kim [21] proposed a TCP for 5G wireless mobile networks that can distinguish network congestion from packet transmission failure, and simulation results showed that the proposed TCP outperforms previous TCPs in terms of throughput. Then, Xiao et al. [22] developed a framework for model-free, smart congestion control based on DRL(Deep Reinforcement Learning), which is robust to varying network conditions. Extensive simulations were conducted to validate its superior performance over five benchmark algorithms. TCP-Drinc achieves the highest throughput and the second lowest RTT for the entire range of propagation delay. Moltchanov et al. [23] developed an analytical model that expresses the TCP throughput as a function of the RTT, environmental, and radio system parameters. This allows TCP to benefit from the high bandwidth of the 5G NR air interface.

IV. Proposed Approach

The proposed TCP protocol enhancements, throgh increasing the TCP cwnd size or improving the congestion control algorithms, will improve performance for new radio technologieslike 4G and 5G systems.

The following is the detailed explanation of the proposed TCP-Enewreno during Enhanced Congestion Avoidance Algorithm (ECAA).

The main idea behind this mechanism is to use an improved congestion avoidance algorithm to dynamically modify the cwnd at the TCP sender based on network bandwidth.

Congestion control refers to a set of appreciation segments that can be injected into the network without causing congestion. When the connection is established and the slow start phase begins, the first cwnd value is set to one. When an ACK is received, the value is updated to

$$cwnd = cwnd + 1;\ per\ RTT. \tag{2}$$

The proposed mechanism consists of two steps:

1. Estimating the data transfer rate at the TCP sender depending on the network status.
2. Cwnd value change during congestion avoidance depends on data transfer rate estimation.

a) Calculating the data transfer rate

The principle behind estimating the data transfer rate is that the sender observes the ACKs streamlet obtained from the destination and calculates the current data transfer rate available across the TCP connection. Estimating the data rate provided to the destination is one way to do this. More exactly, the TCP sender monitors ACKs and determines the quantity of data delivered to the destination based on the data in ACKs and their reception rate. Then, if there are no losses in the transmission process, the requisite calculation of the current data transfer rate cDTR can be estimated through dividing the amount of the delivered data by the Ack interval time.

$$\text{cDTR} = \frac{\text{D}_{\text{ACK}}}{\text{T}_{\text{interval}}}, \tag{3}$$

where, D_{ACK} is the total quantity of data sent to the destination, and $\text{T}_{\text{interval}}$ is the ACK interval time, which is the difference between the last ACK received time and the present one. Because the ACK reception rate is determined by network status, and the transfer rate is calculated every time an ACK is received, the change in transfer rate is also determined by network status. To put it another way, because the source calculates the data transfer rate from end to end via a TCP connection, the higher the bandwidth of a given link, the higher the data transfer rate.

b) Adjusting cwnd

During the congestion avoidance stage, most of the existing congestion control mechanisms increase the cwnd size linearly by a constant value.

Let diff(DTR) to be the difference between the cDTR and the (pDTR), as:

$$\text{diff(DTR)} = \text{cDTR} - \text{pDTR}. \quad (4)$$

In the current proposed, as long as no losses are detected sender update's, we increase cwnd size when it receives an ACK packet from the receiver TCP according to the following Eq. (5):

$$cwnd = \begin{cases} cwnd + \frac{2}{cwnd}, & \text{if} \quad \text{diff(DTR)} > \alpha_E, \\ cwnd + \frac{1}{cwnd}, & \text{if} \quad \text{diff(DTR)} > \beta_E, \\ 0, & \text{otherwise}, \end{cases} \quad (5)$$

where α_E and β_E are two thresholds such that $\alpha_E > \beta_E$. When establishing a connection, α_E is set to 3 (An insatiable flow could try to seize the available bandwidth at a much faster rate by choosing a higher α_E and gaining an edge over competing flows), and β_E is set to 1 (By selecting a β_E closer to one, an insatiable flow can decide to give up bandwidth more slowly upon congestion). These variables are then dynamically updated dependent on network conditions. The idea subscribed to here is that the average number of packets is to be kept in the router buffer will be relational to α and β, so their value will be controlled by the network status. It could be remarked that the cwnd size increased by two Packets each RTT if cDTR higher enough than the last one calculated, i.e. diff(DTR)α_E or by one Packet per RTT if that difference is not high enough. Otherwise, if diff(DTR) $< \alpha_E$, the increasing cwnd size will be frozen. This is because the network is already overburdened, thus the transmitting rate is not increased.

Algorithm 1 Steps of algorithm Enhanced Congestion Avoidance algorithm:

Repeat Slow Start:
Initial: cwnd = 1;
(each packet Acked)
cwnd++;
until congestion *event, or, cwnd > ssthresh*
if cwnd > *ssthresh* **then**
Congestion Avoidance: our proposed
Every ACK:

$$\text{diff(DTR)} = cDTR - pDTR$$

Calculate *cwnd* by Eq. (5)
end if
until timeout or 3 DUPACKs
Fast Retransmit:
After receive 3
DUPACKs Send that packet;
Invoke Fast Recovery algorithm

V. Performance Evaluation

Wireless devices have become increasingly widespread in recent years, just as the use of wireless networks has expanded. Each of these networks has the same underlying requirement, i.e. to deliver good performance, either in terms of increased throughput or compliance with Quality of Services (QoS) restrictions, such as acceptable throughput under high loss probability and low delay. So, the protocols developed for wired networks should be carefully examined to be accountable for the characteristics of the wireless channel. This is important to define their usability over wireless networks. The performance of TCP-Enewreno will be investigated over 5G networks by Simulator 2 (NS2).

The same mobile networking principles will be used by 5G NR, which permits operation in two frequency ranges: FR1 below 7,125 MHz and is likewise OFDM-based (Orthogonal frequency-division multiplexing).

The approval of TCP-Enewreno as a new protocol in the NS-2 simulator requires executing changeovers on the source. The processes of adding this new modification in NS-2 are precise and complicated. In effect, the performance of the new TCP is based on the gauge of TCP-Newreno and takes the same architecture and the same control mechanism in the congestion condition. However,that TCP will be edited to change control in the traditional congestion based on Additive Increase Multiplicative Decrease (AIMD). NS-2 runs on the LINUX UBUNTU 12.04 LTS operating system It's an open-source discrete event simulator aimed at networking research. It has a lot of features that help in simulating TCP, routing, and multicast protocols over wired and wireless nteworks. It supports a wide range of protocols, and through it complex scenarios can be tested quickly. Given its resilience and modularity, NS2 is very popular in the networking research community. The Tool Command Language (TCL) is used to start the event scheduler, set up the network topology, and inform traffic sources when to start and stop delivering packets through the event scheduler while simulating routing protocols. The end output is obtained by running a TCL program created specifically for this purpose. To extract the data from the trace files (.tr), we used an AWK script, which is a powerful LINUX control function that can process the rows and columns of the file to calculate network performance parameters like throughput and end-to-end delay.

To exhibit the efficacy of the proposed mechanism, the proposed mechanism and the existing mechanisms (such as TCP-Vegas, TCP-Taho, and TCP-Westwood), are implemented, and evaluated considering throughput, packet loss, and Delay as the evaluation criteria. TCP-Enewreno estimates the available bandwidth to control the cwnd considering both the traffic intensity and the varying signal to interference, thus it is suitable for 5G networks. Three matrices: throughput, packet loss, and variations in delay with time, are investigated to demonstrate the advantages of TCP-Enewreno over existing mechanisms. The results were presented in two ways. The first is TCP-Newreno against TCP-Enewreno only. The second is with

TCP-Newreno, TCP-Vegas, TCP-Taho, and TCP-Westwood. Due to the considerable properties of the proposed TCP-Enewreno, it is capable of providing good scalability in wireless situations.

A. Experiment Setup

Based on the configuration parameters shown in Table I, we evaluated the system performance for throughput, delay and packet losses. Simulation is done to achieve our proposal. TCP-Enewreno does not require any sender or recever modification. This characteristic makes the proposed algorithm readily applicable to the current network infrastructure. TCP-Enewreno has been implemented and experimented with emulated networks.

TABLE I. SIMULATION CONFIGURATION PARAMETERS

parameter	Value
Simulation time	60 seconds
QoS feature	true
Multicast feature	true
Air interface latency	2 ms
Core network latency	2 ms
Remote host latency	100 ms
Error model	Data Error Model
TCP variants	TCP-Newreno, TCP-Taho, TCP-Vegs, TCP-Enewreno, TCP-Westwood
packet size of background traffic	1500 bytes

The overview of the 5G NS2 simulator is shown in Fig. 2 There are a number of UEs who are in contact with the base station (BS), core network and Remote server.

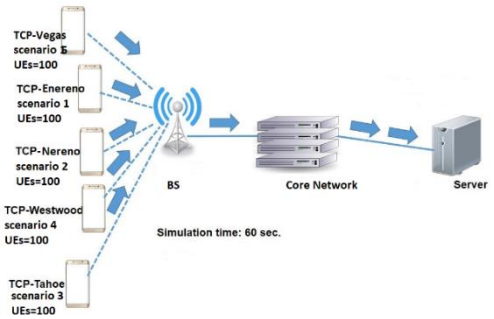

Figure 2. Model of simulation

VI. ANALYSIS DISCUSSION RESULTS

Based on the optimized configuration parameters, we evaluated the system performance in network throughput, average packet delay, and packet loss rate.

A. Throughput

TCP throughput is an essential metric to assess the performance of a network connection. It is the speed at which data is successfully transferred through a TCP connection. Typically, TCP throughput and RTT have an inverse relationship. TCP-Enewreno gets higher throughput than TCP-Newreno. As shown in Fig. 3, it starts with a throughput equal to TCP-Newreno in the first 7 seconds of simulation. It eliminates that unnecessarily hostile sending rate. With time, it achieves higher throughput due to the enhanced congestion avoidance algorithm. This is because, regardless of the current network load, the TCP-Newreno modifies the cwnd according to the blind rate adaption method, i.e., the ACK causes the cwnd to grow, and packet losses cause the window to drop. While in our approach TCP-Enewreno adjusts its transmitting rate based on the current bandwidth utilization by adjusting the cwnd, it achieves a more ratio throughout. During the fast recovery phase, TCP-Newreno similarly waits for the recovery of all lost packets before sending a few new packets. Yet, TCP-Enewreno was reduced by the network load, allowing it to send more new packets depending on the network status. The optimal current conditions for TCP are the main reason for achieving similar throughputs. TCP can function successfully when there are no chances of packet loss and no congestion caused by many existing UEs creating a bottleneck. As shown in Fig. 4, the proposed algorithm functions effectively in a wireless network environment. Compared to TCP Vegas, it has a substantially better throughput while having a hardly noticeable increase in delay because the cwnd of the TCP-Vegas algorithm is linearly adjusted based on the difference between the current RTT and the minimum RTT. It maintains RTT at a low level, but it is susceptible to changes in the network environment and has limited throughput. We saw that the setting of cwnd in TCP-Westwood was nearly the same as in TCP-Taho when the available bandwidth was reduced So, TCP-Westwood and TCP-Taho cannot follow the immediate dynamics of the available bandwidth, and thus cannot use the available bandwidth efficiently.

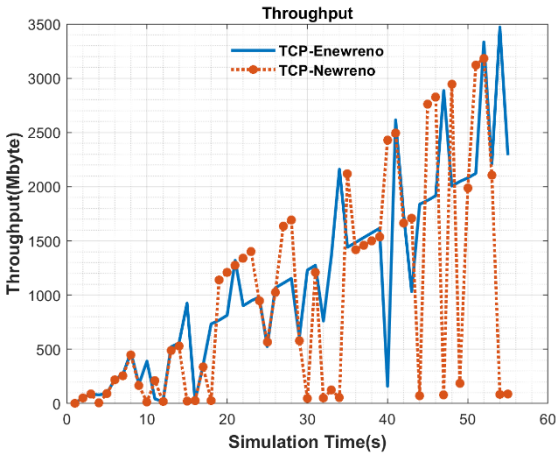

Figure 3. Throughput of TCP-Enewreno against TCP-Newreno.

B. Delay

Because a computer using a TCP/IP network delivers a finite amount of data to its destination and then waits for an ACK before sending any more data, the round-trip delay is a key metric. Therefore, the RTT has a big impact

on network performance. In most events, latency is measured in milliseconds (ms). Fig. 5 shows the delay for TCP-Enewreno and TCP-Newreno. TCP-Enewreno uses modified a algorithm that explores the network environment and adjusts the congestion window to reliably transport data. Which can better adapt to the changes in network bandwidth resources and reduce the probability of congestion, the queuing delay of packets is reduced, so the RTT of data transmission is reduced. Fig. 6 shows the comparison of normalized delay. It can be seen from the figure that TCP-Newreno has the largest delay, followed by TCP-Taho, TCP-Enewreno is Nearly similar to TCP-Westwood and TCP-Vegas has the smallest delay. This is because TCP Vegas is the first delay-based protocol. it is a latency-based strategy that linearly modifies its cwnd according to the difference between the current RTT and the minimum RTT.

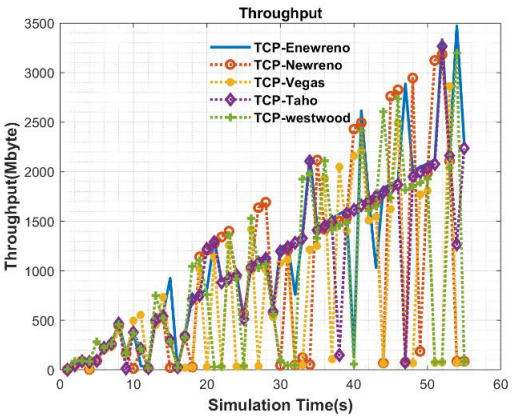

Figure 4. Throughput of TCP-variant.

Figure 5. Delay of TCP-Enewreno against TCP-Newreno.

C. Packetloss

The error model is used to study the wireless environments effect on the proposed in NS-2. The used model is ErrorModel, while the deployed random variable is RandomVariable/Uniform. Agent/Null is configured to handle the dropped target. Fig. 7 shows the packet loss performance. The result shows that the TCP-Enewreno can reduce the packet loss against TCP-Newreno and more improvements can be gained when the number of active users is 100. This is because the channel with a high bit error rate (BER) causes more packet retransmission, which leads to more packets being buffered in the BS In this case, the TCP-Enewreno based on changing cwnd dynamically will become more effective. Fig. 8 shows the benefits of using TCP-Enewreno over the existing algorithms, Taho, Westwood, and Vegas. TCP-Enewreno achieves less packet loss than the other algorithms because the TCP-Enewreno modifies the congestion window to modify its transmitting rate based on the current network load (see Fig. 8).

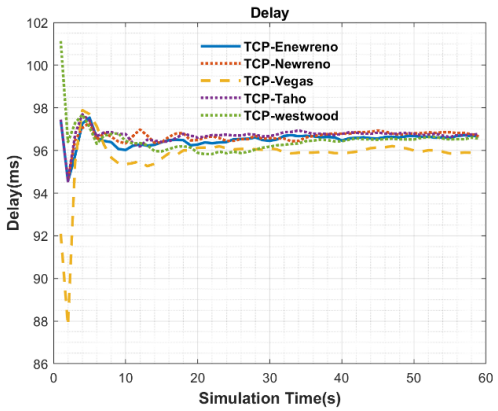

Figure 6. Delay of TCP-variant.

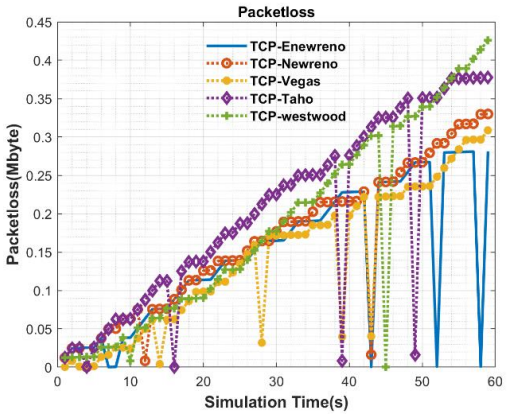

Figure 7. Packetloss of TCP-variant.

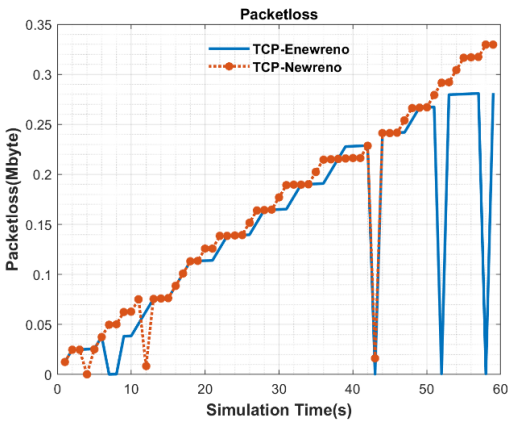

Figure 8. Packetloss of TCP-Enewreno against TCP-newreno.

D. Fairness

In network engineering, the fairness metric is used to assess whether users or applications are getting a fair share

of system resources or not. New network transmission protocols and peer-to-peer applications require congestion control algorithms that function well with the current TCP variations. There are several mathematical and conceptual definitions of fairness such as Jain's Fairness Index (JFI) [24–26] as shown in the following Eq. (6)

$$f(x_1, \dots, x_n) = \frac{(\sum_{i=1}^{n} x_i)}{n \sum_{i=1}^{n} x_i^2}, \quad (6)$$

where x_i is represents the throughput of $flow_i$, from N flows in the system. In Table II, all of the TCP variants were almost the same and they achieve similar Fairness Index because the bottleneck is not congested.

TABLE II. JAIN FAIRNESS INDEX OF TCP-VARIANTS

TCP-Newreno	0.510478
TCP-Taho	0.510746
TCP-Vegas	0.511756
TCP-Westeood	0.510163
TCP-Enewreno	0.510887

VII. CONCLUSION

An effective approach for utilizing available bandwidth is crucial in the wireless links of integrated wired and wireless networks. In a dynamic internet context, some approaches suggest effective methods for estimating the available bandwidth, while others successfully control the congestion window. This paper presentes a new implementation of TCP-Enewreno over 5G networks. The excellent performance that TCP-Enewreno achieves mostly comes from two basic mechanisms: Bandwidth Estimation and dynamically adjusting cwnd, where the sender measures the network bandwidth and controls the transfer rate adaptively according to the transfer rate. TCP-Enewreno in a congestion avoidance phase controls the transfer rate similar to the traditional TCP. It also improves the transmission performance in wireless networks due to the increasing transfer rate in case packet losses caused by link errors reduce the congestion window. Our simulation showed that TCP-Enewreno could achieve better performance than the other four implementations in the throughput, packet loss, and the second lowest delay. The TCP-Enewreno module is developed in NS-2 and lots of simulations have been carried out to verify the performance.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

S. M. Abdullah Methodology, Software, Validation, Formal analysis, Investigation, Writing – Original, Formal analysis; M. S. Farag Writing - Review & Editing, Visualization, Supervision; H. Abdul-Kader Conceptualization, Software, Investigation, Writing - Review & Editing, Supervision; S. E. Abu-Youssef Formal analysis, Investigation, Writing - Review & Editing, Supervision. all authors had approved the final version.

REFERENCES

[1] C. Gartenberg. 5g: Everything you need to know. [Online]. https://www.theverge.com/21284203/5g-need-to-know- health-safety-radiowave-spectrum-verizon-att-tmobile-cellular-data

[2] M. Panda, H. L. Vu, M. Mandjes, and S. R. Pokhrel, "Per- formance 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 W. Stevens, "Rfc2581: Tcp con- gestion control," 1999.
M. Zhang, M. Mezzavilla, R. Ford, S. Rangan, S. Panwar, E. Mellios, D. Kong, A. Nix, and M. Zorzi, "Transport layer performance in 5g mmwave cellular," pp. 730–735, 2016.

[4] I. Petrov and T. Janevski, *Design of Novel 5G Transport Pro-Tocol*. 2016.

[5] Y. Cheng, F. Cheng, B. Deng, J. Li, and C. Mei, "Arq algo- rithm optimization of radio link control layer in 5g system," in *Proc. 2020 the 2nd World Symposium on Software Engineering*, pp. 135–140, 2020.

[6] G. O. Melih and D. Mubeccel, "Average throughput performance of myopic policy in energy harvesting wireless sensor," *Sensors, Publisher MDPI Networks*,vol. 17, no. 10, p. 2206, 2017.

[7] G. O. Melih, "Asymptotically optimal scheduling for energy harvesting wireless sensor networks," in *Proc. 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, pp. 1--7, 2017.

[8] G. O. Melih and D. Mubeccel, "Asymptotically throughput optimal scheduling for energy harvesting wireless sensor networks," *IEEE Access*, vol. 6, p. 45004-45020, 2018.

[9] A. Gupta and R. K. Jha, "A survey of 5g network: Archi- tecture and emerging technologies," *IEEE access*, vol. 3, pp. 1206–1232, 2015.

[10] S. Ahmadi, *5G NR: Architecture, Technology, Implementa-tion, and Operation of 3GPP New Radio Standards*, Aca- demic Press, 2019.

[11] 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," *Sus- tainability*, vol. 10, no. 5, p. 1337, 2018.

[12] J. Lorincz, Z. Klarin, and J. Ožegović, "A comprehensive overview of tcp congestion control in 5g networks: Re- search challenges and future perspectives," *Sensors*, vol. 21, no. 13, p. 4510, 2021.

[13] W. R. Stevens, "TCP slow start, congestion avoidance, fast retransmit, and fast recovery algorithms," *RFC* 2001, Jan. 1997.

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

[15] J. Hoe, "Startup dynamics of tcp's congestion control and avoidance schemes," Master'S thesis, MIT, 1995.

[16] M. Saverio, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP Westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. 7th Annual International Conference on Mobile Computing and Networking*, pp. 287-297. 2001.

[17] S. Sea, G. Hazel, J. Iyengar, and M. Kuehlewind, "*Low extra delay background transport (LEDBAT)*," 2012.

[18] J. Wang, J. Wen, Y. Han, J. Zhang, C. Li, and Z. Xiong, "Cubic-fit: A high performance and tcp cubic friendly con- gestion control algorithm," *IEEE Communications Letters*, vol. 17, no. 8, pp. 1664–1667, 2013.

[19] X. Jiang and G. Jin, "Cltcp: an adaptive tcp congestion control algorithm based on congestion level," *IEEE Communi- cations Letters*, vol. 19, no. 8, pp. 1307–1310, 2015.

[20] Y. Gwak and R. Y. Kim, "A novel wireless tcp for 5g mobile networks," *World J. Wirel. Devices Eng*, vol. 1, no. 1, pp. 1– 6, 2017.

[21] K. Xiao, S. Mao, and J. K. Tugnait, "Tcp-drinc: Smart congestion control based on deep reinforcement learning," *IEEE Access*, vol. 7, pp. 11892–11904, 2019.

D. Moltchanov, A. Ometov, P. Kustarev, O. Evsutin, J. Hosek, and Y. Koucheryavy, "Analytical tcp model for millimeter-wave 5g nr systems in dynamic human body blockage environment," *Sensors*, vol. 20, no. 14, p. 3880, 2020.

[22] R. Jain, D. Chiu, and W. Hawe, "A quantitative measure of fairness and discrimination for resource allocation in shared computer systems," *DEC Research Report TR-301*, September 1984.

[23] G. O. Melih, "Achieving near-optimal fairness in energy harvesting wireless sensor networks," in *Proc. 2019 IEEE Symposium on Computers and Communications (ISCC)*, pp.1-6, 2019.

[24] M. Dimitrios, M. Bateman, and S. Bhatti, "Fairness of high-speed TCP stacks," in *Proc. 22nd International Conference on Advanced Information Networking and Applications*, pp. 84-92. IEEE, 2008.