IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 18, 2021, accepted June 3, 2021, date of publication June 7, 2021, date of current version June 15, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3087239

FB-TCP: A 5G mmWave Friendly TCP for Urban Deployments

REZA POORZARE AND ANNA CALVERAS AUGÉ
Department of Network Engineering, Universitat Politècnica de Catalunya–UPC BarcelonaTech, 08034 Barcelona, Spain

Corresponding author: Reza Poorzare (reza.poorzare@upc.edu)

This work was supported in part by the Secretaria d'Universitats i Recerca del Departament d'Empresa i Coneixement de la Generalitat de Catalunya under Grant 2017 SGR 376, and in part by the Spanish Government under Project PID2019-106808RA-I00 AEI/FEDER UE.

ABSTRACT 5G era opens a new horizon toward communication with new features and capabilities. The new mobile generation comes up with a multi-gigabit per second data rate along with its huge available bandwidths provided by millimeter-wave frequency bands. Dispensing a fully connected world throughout low latency is the supreme aim for 5G networks. However, attaining high-speed, reliable communication is challenging in the new generation, especially in scenarios with numerous obstacles such as the urban one. As the frequency rises, the signal's penetration power declines, which can mislead TCP in adjusting the sending rate because TCP cannot distinguish that a packet drop in a network is due to congestion or other shortcomings of the network such as blockage or random packet drops. This paper proposes a new TCP based on Fuzzy logic, which strives to prevent performance reduction in urban deployments. The Fuzzy rules are implemented in the congestion avoidance phase of the new protocol to adjust the sending rate intelligently and prevent blockage impacts. The ultimate aim of the protocol is to control the sending rate based on the current situation of the network so it can attain the highest possible performance. Moreover, it tries to reach its goal through low latency and keep the average sending rate as small as possible to restrain the buffer exhaustion. The extensive conducted simulations showed that the newly proposed protocol could attain higher performance compared to BBR, HighSpeed, Cubic, and NewReno in terms of throughput, RTT, and sending rate adjustment in the urban scenario.

INDEX TERMS 5G, end-to-end reliability, fuzzy logic, mmWave, TCP.

I. INTRODUCTION

By appearing new services and high-quality videos, exploiting high frequencies in 5G (Fifth Generation) is inevitable. As a consequence, millimeter-wave (mmWave) frequency bands are going to be deployed in the new mobile generation to provide high bandwidth for the wireless channels in order to fulfill the demand for large data rates [1], [2]. In addition to mmWave deployment, sophisticated spectrum management and a novel design for the core network are other enablers to support new features [3]. These features can support three primary use cases, including eMBB (Enhanced Mobile Broadband) to deliver high data rates, URLLC (Ultra-Reliable Low-Latency Communication) for satisfying latency-critical applications demands, and mMTC (massive Machine Type Communication) for providing connectivity to a large number of devices [4], [5]. The mmWave frequency suffers from a drawback called NLoS (Non-Line of Sight), which refers to a situation that a UE (User Equipment) and a gNB (gNodeB) cannot find each other because of an obstacle blocking the communication path [1], [2], [6]. When numerous NLoS to LoS (Line of Sight) transitions occur in a network, they can force performance degradation and reduce the quality of user experience. The principal reason is that reliable end-to-end communication relies on the protocol stack widely used transport protocol TCP (Transmission Control Protocol) [7], and without a suitable protocol that is able to exploit the 5G full potential, the wide bandwidth of mmWave will be wasted [2], [8]. So that one of the main pillars of the mmWave application relies on effective and sufficient TCP implementation [9].

Many TCPs cannot function sufficiently in high-speed networks, especially with the random packet losses, which is one of the utmost wireless network issues [10], [11]. The reason is

The associate editor coordinating the review of this manuscript and approving it for publication was Cunhua Pan.

that most of the protocols are not capable of proper reactions to 5G mmWave networks' different statuses, especially LoS to NLoS changes. Conventionally, TCP assumes a packet drop as an indicator to congested situations but this might differ in 5G networks because different reasons such as blockage or environmental impacts can force random packet losses and degrade the protocol's performance [1], [2]. TCP's improper reaction to lossy wireless channels of 5G and assuming them as congestion indicators can strangle the TCP congestion window (cwnd) mistakenly and cause underutilization of the network's resources [12].

A comprehensive analysis of 5G mmWave in practice has been done in [13], and the obtained results in various tests corroborate the benefits of deploying mmWave in order to fulfill the demanding data rate for the new generation, and its deployment was a success in 5G networks [14]. However, the susceptibility of mmWave to distance and NLoS states have been outlined.

As the predictions show, 5G is going to take 60% of the world population under its coverage by 2026 [15]. As a result, redefining the transport layer and creating a new TCP compatible with 5G is essential. Moreover, beyond 5G networks such as 6G (Sixth Generation) will exploit the Terahertz spectrum, where the issue could be more demanding [3]. Employing densification on account of micro antennas indeed is one of the candidates to mitigate higher frequency issues. However, it could lead to higher expenses, operational efforts, and energy consumption [14].

Several reasons can make the blockage effect or NLoS states more intense or relieve its negative impacts. Parameters such as the number of obstacles, the topology's layout, or the distance between a UE and a gNB can play a vital role in the ultimate performance. All these criteria can be changed based on the deployed scenario. As a result, one of the important criteria is the deployed scenarios based on the ITU-R (International Telecommunication Union- Radiocommunication) proposals [16], affecting the final performance.

We have exploited the full buffer traffic pattern, in which a huge amount of data is waiting to be transmitted [17] in the urban deployment scenario [18]. In this paper, we will focus on 3GPP's most popular scenario called urban deployment because one of the most important goals of ubiquitous networks such as 5G is providing appropriate communication for the population who reside inside cities in order to support having smart cities with new features and services. Moreover, deploying a high frequency in the urban deployment, existing many obstacles such as buildings, cars, and human bodies make it difficult for TCP to function properly due to the existence of dynamic LoS- NLoS transitions [19]. The newly proposed protocol's main goal should be utilizing the vast available resources in 5G mmWave networks. This protocol can give up being fair to some levels to use the new generation's full potential. To be more apparent, the sufficient use of the resources can be increased at the cost of fairness. This approach that has been used in BBR [20] (Bottleneck Bandwidth and Round-trip propagation time), which is one of the deployed protocols by large companies such as Netflix and Google, indicates that the era for friendly TCPs has been broken down, and the priority is using the substantial available resources [21].

A well-designed model-based TCP can work more efficiently than loss-based TCPs in 5G mmWave. The proactive mechanism in the nature of this type of TCPs can adjust the sending rate before the queues are full and packet drops start to happen [22], so our newly designed FB-TCP protocol (Fuzzy Based-TCP) adheres to this principle.

The extensive conducted simulations proved the proposed FB-TCP sufficiency over other TCP variants such as NewReno, Cubic, HighSpeed, and BBR in terms of salient KPIs (Key Performance Indicators), including throughput, RTT, and cwnd adjustment. Therefore, we will demonstrate that the proposed protocol can utilize the 5G mmWave bandwidth to its full potential by reaching an acceptable latency

The rest of the paper is organized as follows: section two presents the related work, section three talks about the Fuzzy logic, section four explains the newly proposed protocol, section five incorporates methodology and the simulation parameters, section six describes the simulation scenarios and the results, and finally, section seven concludes the paper.

II. RELATED WORK

TCP is the widely exploited protocol in the transport layer, which dispenses reliable end-to-end communication provided by its connection-oriented nature. This system tries to achieve the highest available throughput in a network by preventing congestion status [2]. There are various TCP variants with distinct characteristics, making them appropriate for different situations [10].

Based on the popularity, we have chosen four TCPs to be compared to FB-TCP. The first one is NewReno [23], which was created to improve some aspects of Reno [24], and exploits AIMD (Additive Increase Multiplicative Decrease) to adjust the sending rate. The second one is CUBIC [25], which employs a cubic function in controlling the sending rate. The third one is HighSpeed [26], which deploys an aggressive congestion control mechanism to utilize the high available bandwidth in a network, and the last one is BBR [20], [27], a model-based TCP that tries to adjust the sending rate by accommodating it to the bottleneck bandwidth along with retaining low RTTs [22].

NewReno has worked as the default TCP in many applications for years [1] and was the base one in designing the following protocols, especially the loss-based ones. CUBIC is the default protocol from Linux Kernel 2.6.26, Android, and iOS operating systems [8], [28]. HighSpeed is the default protocol for designing TCPs with aggressive mechanisms in increasing and decreasing the sending rate, making it suitable for networks with high BDP (Bandwidth Delay Product). Finally, BBR is a cutting-edge protocol deployed in popular services such as YouTube and Netflix [21]. There is a comprehensive analysis of TCP, its procedure, and parameters in [2].

After establishing the deployed protocols and why they have been chosen, the next pace is detecting the issue on the way of utilizing the full potential of 5G mmWave networks. A comprehensive investigation of the 5G mmWave network and its various aspects was done in [2]. Moreover, TCP's functionality and its compatibility to 5G networks were also analyzed throughout the paper. On the other hand, the impact of 5G network characteristics on the design of novel congestion control algorithms was discussed in [11]. These two papers could cover the challenges and hurdles on the way of designing an appropriate TCP for 5G networks and propose new insights that can ease the path of researchers in outlining new protocols befitting for the new mobile communication.

In addition to the presented full guidelines in the previous papers, a practical investigation was performed in [29] on the first implemented 5G network in Chicago. The exploited carrier frequency was 28 GHz, which could indicate the performance of 5G mmWave networks. The results revealed that the high-frequency ranges could support large bandwidth, but in reality, the performance showed severe fluctuations both in LoS and NLoS states. The root of this deficiency is that TCP cannot operate sufficiently in 5G mmWave networks and need some modifications.

Thorough analyses of high-Speed and indoor scenarios were done in [1] and [30], respectively. Both studies proved the inadequacy of TCP, especially in NLoS states. Moreover, an examination of the urban deployment scenario in [1] showed that some modifications should be done on TCP for attaining high throughput and low latency in 5G networks.

An analysis of different blockage types, the impact of handover, along with various flows, has been brought in [8]. The paper indicated that aggressive TCPs such as CUBIC could take advantage of modifying RTO (Retransmission Time-Out) and RLC (Radio Link Layer) buffer by reducing the former and increasing the latter one. The results also showed that TCP suffers especially from longer blockage.

As the new mobile generation supports high data rates, exploiting several interfaces can be beneficial. In this case, deploying MP-TCP [31] (Multi-Path TCP), which is capable of handling various interfaces, could be one of the appropriate choices. The simulation results in [32] showed that MP-TCP could have some advantages compared to SP-TCPs (Single-Path TCP), mostly in LoS states. However, we should notice that NLoS states can occur more frequently in 5G mmWave networks and makes it difficult for an MP-TCP to maintain its functionality. Additionally, when LTE (Long Term Evolution) and 5G coexist, it affects the MP-TCP functionality in an adverse way, as MP-TCP finds LTE channels more stable. On the other hand, the mid-band spectrum deployed in LTE cannot deliver the data rate that mmWave can support [13].

All the mentioned statements show that TCP suffers from a dearth of performance in 5G mmWave networks. This deficiency can be congestion window fluctuation and incompetent mechanism of TCP in adjusting its sending rate. Moreover, latency value, which is critical for 5G networks, can be damaged because of TCP's scant functionality, especially in NLoS states, in which the communication path is blocked by an obstacle, but TCP continues sending packets that lead to long queues in buffers and can cause a bufferbloating problem. All these issues can be seen in the results of [33], [34], which show TCP's functionality over the urban deployment. As the results revealed, the throughput can be degraded along with RTT (Round Trip Time) increment in NLoS states. Moreover, sending more packets blindly by different TCP variants in order to gain higher throughput can exhaust senders' buffers.

There have been some efforts to alleviate these issues, such as a Deep learning-based TCP called DL-TCP (Deep-Learning TCP), which was proposed in [6]. As a disastrous situation happens, numerous collapsed buildings and trees can act as blockers and make consecutive NLoS states. In this case, establishing an adequate communication path between a UAV (Unmanned Aerial Vehicles) and the control center would be challenging. As a result, DL-TCP, based on a Deep Neural Network engine, strives to distinguish RTOs caused by congestion from the ones caused by the blockage. The simulation results showed that the newly proposed protocol could outperform other TCP variants. The principal shortcoming of this scheme is that it has not been tested in new topologies to see the efficiency of the protocol when the layout is changed. As a result, being trained and tested in the same environment can be assumed as a flaw for this protocol.

Another approach called D-TCL (Dynamic-TCP) was proposed in [35]. This protocol's prime aim is to handle the adverse impact of random packet drops in 5G mmWave networks by the appraisal of the at-hand bandwidth. They tried to adapt the new TCP to the high BDP and lossy nature of 5G mmWave networks. The authors claimed that D-TCP can learn the available bandwidth, so it can tolerate high variations of the paths in higher frequencies. The sending rate in D-TCP is adjusted by making use of a congestion control factor, which is derived from the estimated available bandwidth. Not being compared to aggressive TCPs such as HighSpeed, which can be one of the well-suited protocols for 5G mmWave networks [33], [34], and not having a discussion on the obtained average RTTs can be mentioned as the downsides of D-TCP.

Moreover, they have been some efforts in enhancing TCP's functionality through Fuzzy logic. As an example, a Fuzzy-based TCP called TCP-FRTT was proposed in [36], which was striving to enhance the reliability in WLAN (Wireless Local Area Networks). The primary purpose of the new protocol was handling the mislead TCP due to random packet losses in wireless networks caused by mobility or the intermittent nature of the wireless channels. The protocol could achieve its aims by modifying the TCP Vegas congestion avoidance phase and could attain higher performance in terms of throughput, delay, and networks resources utilization. A parameter called FE (Fluctuation Estent) was employed in TCP-FRTT as an output to measure the scalability of the network.

Moreover, two inputs, diff, and p are exploited to calculate the FE. All of the parameters are computed through some Fuzzy memberships and rules. The protocol was compared to two TCPs, including TCP Reno and TCP-RM [37], and the results proved the superiority of the new protocol over them.

Another scheme called FPRD (Fuzzy Pattern Recognition based Differentiating) [38] uses a Fuzzy-based algorithm to distinguish different types of packet losses in a network that both wired and wireless communications are deployed. The algorithm tries to link a loss to distinctive patterns, such as a congestion pattern or a wireless channel error pattern. The justification behind this protocol is that a loss has a Fuzzy nature; thus, exploiting this technique can be helpful. The main chosen parameters for calculating the membership function are RTT, congestion window size, inter-arrival time between ACKs, and re-transmit rate, however, the most critical parameter is ROD (Relative One-way Delay). The simulation results indicated that the protocol could have better functionality compared to Reno, New Reno, and Vegas in terms of throughput.

Another approach based on Fuzzy to improve TCP called FFC-Snoop was proposed in [39]. This algorithm strives to give some feedback to the sender by calculating the buffer size based on Fuzzy memberships. The main goal of the scheme is to show fast reactions to the network's current situation by having a Fuzzy membership that reflects the buffer's ongoing situation and omits the historical events. The simulation results indicated that the proposed protocol could achieve higher throughput than NewReno.

TCP-C [40], which was a new protocol, used Fuzzy logic to label various packets. This marking mechanism is done in order to give some feedback to the sender from the network's current situation; thus, it can adjust the sending rate appropriately. The results showed that TCP-C could have better performance compared to NewReno and Westwood in terms of throughput.

The main drawback of the mentioned protocols is that they cannot achieve higher possible performance, and they have been compared to TCPs that generally are not able to attain the saturated throughput in wireless networks. As a result, they are not proper candidates for 5G mmWave networks.

Another Fuzzy-based scheme for improving the downlink performance in 5G networks was proposed in [41]. The protocol strives to use fuzzy membership and rules to control the downlink flow in order to enhance buffering mechanism in the network. The results revealed that the new protocol could adequately handle the queue sizes and achieved better performances in terms of buffer occupancy, loss rate, and delay compared to previous protocols. The main issue of this algorithm is increasing the computational overhead in the network, which should be reduced in future work.

To sum up, the effect of 5G mmWave characteristics on the performance of TCP is more intense than sub-6 GHz frequencies deployed in LTE or 3G. 5G mmWave can provide high data rates through a wide spectrum, but it brings novel characteristics that can degrade the transport layer's functionality; as a result, the user experience. These issues incorporate: (i) a lossy environment that can create numerous random packet losses, which leads to a high value for BER (Bit Error Rate). (ii) Being susceptible to obstacles, which creates NLoS states and makes it difficult to have stable channels for communication. (iii) Decreased throughput and increased RTT in NLoS states. (iv) Difficulties in sending rate adjustment, which leads to cwnd fluctuations. (v) Most TCPs cannot distinguish the drops caused by blockage and congestion and decrease or increase their sending rate without having a clue from the network's status, which can lead to the underutilization of the resources, exhausting the buffers, or creating bufferbloating problem. The latter issue can be alleviated by deploying some techniques of AQM (Active Queue Management) such as CoDel [42] or Fq-CoDel [43]. However, they are not sufficient and need some modification to be deployed in 5G mmWave networks [1].

III. FUZZY LOGIC

In this section, we are going to have a brief discussion on Fuzzy logic as the background in determining some of the deployed parameters in the FB-TCP protocol proposed in this paper. Fuzzy logic [44] or Fuzzy sets, a subset of AI (Artificial Intelligence), is for indicating membership of an object in a class with a membership function, which is between zero and one. If we have domain X as our objects, a fuzzy set of $f_A(x)$ links individual points of 'x,' a value of membership degree in A. In the classical sets or ordinary sets, $f_A(x) = 0$ or 1, which indicates whether x belongs to A or not. As a result, the main difference between the ordinary sets and fuzzy sets is that the membership function is zero or one in the former one, but in the latter one, the membership function is between zero and one. For example, if X is the real numbers and we have a set of numbers greater than 100, our $f_A(x)$ for different values can be $f_A(50)=0$, $f_A(100)=0$, $f_A(200)=0.1$, $f_A(1000)=0.5$, and $f_A(10000)=1$.

$f_A(x)$ is called the membership function, and the corresponding values are membership degrees. In $f_A(x)$, membership degrees show the belonging degree of x in A. In general, zero indicates non-membership, one shows full membership, and the values between them are for partial membership [44].

One of the main aims of introducing Fuzzy was enabling machines to do tasks that had been difficult and complex for decades because of the lack of intelligence. Historically, machines have not been able to perform tasks that human could do easily. Fuzzy has been excelling systems in reaching pinnacles that were impossible before. As a result, Fuzzy strives to model real-world events, which were hazy before it. The primary tool, which Fuzzy has is mathematical calculations that help it in order to model indeterminate problems. With the help of the inputs, output, and rules, Fuzzy enables systems to communicate with their surroundings and solve the issues that were not possible to be handled before [45].

As a result, Fuzzy logic is a suitable paradigm for decision-making and clustering problems, which can be used in complicated systems. In our proposal, we deploy Fuzzy logic to modify the TCP congestion control mechanism in order to work properly in 5G mmWave networks. The new protocol's primary goal can be dividing the network into different clusters and then adjust the sending rate based on the current cluster that the network is. Because of the nature of the Fuzzy, it can address various aspects in 5G mmWave networks. One of these features is handover, which is for changing the gNB that the user is connected to. In this case, the best antenna can be selected based on some Fuzzy memberships. This procedure provides intelligence so that the handover process can benefit from this smartness. Another usage of Fuzzy can be in the 5GCN (5G Core Network) design. SDN (Software-Defined Networking)/NFV (Network Function Virtualization) are two paramount enablers in the core of 5G, and if Fuzzy can yield smartness to these features, their functionality can be enhanced dramatically. In addition to the mentioned features, Fuzzy can be employed in the queueing algorithms and improve their performance. In this case, the RLC buffer can be controlled efficiently and reduce the end-to-end delay. As a result, lower latency, which is one of the essential pillars in 5G, especially in URLLC, can be decreased intensely. Finally, Fuzzy can be used to improve the controlling mechanism in always-on; as a result, it can assist in reaching an ultra-lean design [2]. All of the mentioned aspects can be refined with the help of Fuzzy logic. However, in this paper, we have decided to concentrate on the protocol side of the communication and improve throughput, latency, and sending rate adjustment.

IV. FB-TCP: FUZZY BASED-TCP

As it was mentioned, the first goal of designing a new protocol should be attaining the highest available throughput along with acceptable latency. Moreover, the protocol should be able to tolerate random packet drops, because if not, consecutive losses in NLoS states impair its functionality dramatically. Furthermore, the protocol should be able to detect different situations from each other and function based on the current one. To sum up, a newly proposed protocol should:

- Function close to the UDP saturated value.
- Prevent cwnd high fluctuation.
- Prevent consecutive RTO triggering in NLoS states.
- Prevent bufferbloating problem.
- Have a constant functionality.
- Be immune to random packet losses.
- Be immune to losses caused by NLoS states.
- Reach the highest available throughput through fast paces.
- Reach the highest available throughput through low average cwnd.
- Prevent consecutive RLC buffer overflow in NLoS states.

As a result, dividing the network into various sections from non-desirable to the desirable range is essential. In this case, LoS and NLoS states can be distinguished, and a proper functionality can be achieved. The critical aspect of the protocol is the time that the UE is in a NLoS state. In this way, the cwnd should be adjusted carefully to prevent the buffer overflow and keep the RTT as low as possible along with high throughput.

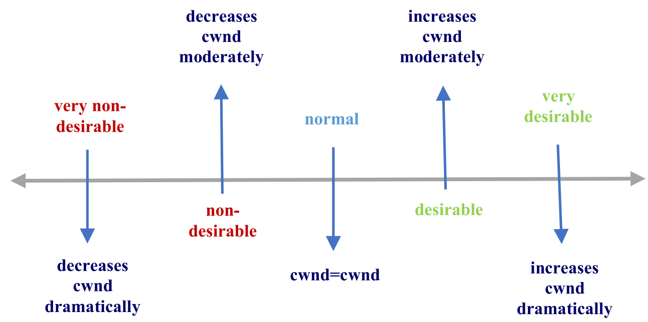

FIGURE 1. How clustering works.

FB-TCP strives to handle the issues that TCP encounters in 5G mmWave networks by relying on Fuzzy logic and deploying some novel features and parameters. The operation of FB-TCP is based on the division of the network into several sub-states and decides based on the current state. The main goal behind this clustering is to set a range of parts in the network representing a set of conditions from non- desirable to desirable ones. In this case, when the network is moving toward desirable situations, the protocol can operate optimistically. In contrast, when the protocol is in non-desirable conditions, FB-TCP will function pessimistically. The factor for the increasing and decreasing the cwnd is based on a higher level division that will be explained in the rest of this section. Not changing the sending rate is an option for the time that the network is between desirable and non- desirable situations. Figure 1 indicates these states and how FB-TCP reacts.

Furthermore, individual clusters can be divided into sub-clusters in order to assist the protocol in making more accurate decisions.

The first step is to calculate the maximum available sending rate by estimating the current BDP. As a consequence, every 100 ms, the number of delivered packets are counted, and by exploiting (1), the maximum value for cwnd is figured:

$$maxCwnd = ((DP^{*}minRtt)/8)/MSS))^{*}\rho \qquad (1)$$

where maxCwnd is the maximum value needed for cwnd to reach the highest available throughput in the network, the DP is the number of delivered packets every 100 ms in our experiments (a tradeoff of simulation time and performance), and minRtt is the minimum RTT in the network for a connection. MSS (Maximum Segment Size) is also the largest value for a TCP connection that a node can receive. The calculated value is multiplied to ρ. In our study, we choose 1.05, so we can set the upper bound 5% more than the estimated value to discover more bandwidth in the network. This value can be selected based on the level of aggressiveness that we want the protocol to have. By choosing 5%, the protocol always will try to turn up the extra available bandwidth in the network. Selecting a large number will lead to high throughput at the cost of RTT.

However, a small value can enhance RTT at the expense of throughput.

The next step is choosing some parameters and formulas that assist us in adjusting in-flight packets. The principal criteria in selecting these parameters were: 1) they could be exploited as Fuzzy membership functions, 2) they could reflect the current status of the network, and 3) they were independent of packet losses. One of the primary choices can be exploiting the difference between the current sending rate and the rate the cwnd should be adjusted to attain the maximum throughput, i.e., targetedCwnd. This can give a vision of how bad the sending rate has been tuned and whether we are moving so fast or not. If the difference between these two parameters gets higher, it can be assumed as a negative sign. In contrast, when the value is close to zero, it can be a positive sign.

RTT is another principal element that can indicate the different conditions of the network. The reason is that this KPI is differentiated in LoS, NLoS, or other situations such as congestion in the network. Moreover, by having the relationship between the minimum RTT of the connection and minimum RTT of the window, a proper insight from the network can be obtained.

The maxCwnd parameter will be exploited to divide the network into two main phases called Convergence and Divergence. Then, each phase is divided into several sub-phases using some parameters including, Diff (Difference), CSI (Congestion Status Indicator), and CAD (Cwnd ADjuster). The main reason behind dividing the network into two major sections is for determining the upper bound of the network and making decisions based on it. The crucial point of the network is the size for cwnd that can utilize the network's full potential. In FB-TCP, this point is ascertained by the maxCwnd parameter. Being below this point means that the network is not functioning at its full potential so we can increase the cwnd based on the difference between maxCwnd and the current cwnd size.

In contrast, being above this spot indicates that the cwnd is forcing more packets than the network's current capacity, and FBTP should decrease the cwnd. The intensity of this decrement is due to the difference between maxCwnd and current cwnd. As seen in Figure 1, when the cwnd size is around maxCwnd, FB-TCP assumes that the network is functioning normally. However, when the cwnd is less than this point, the network is in its desirable status, and the number of sent packets can be increased because the available capacity in the network can handle more data.

On the other hand, when cwnd is more than maxCwnd, the network is in its undesirable status, so FB-TCP reduces the sending rate to prevent more burden on the network. This can happen in different situations, such as congestion or NLoS states. In both cases, the network's capacity is less than its normal situation, and FB-TCP strives to adapt the sending rate to the available capacity. To sum up, the Convergence phase tries to handle desirable modes, and the Divergence one is tackling undesirable situations.

Different clusters are subsets of desirable or non-desirable states that are shown in Figure 1. As a result, being in the desirable situation in the Convergence is different from non-desirable in the Divergence. For example, the Convergence phase's aggressiveness during the desirable condition is much higher than the one for the Divergence.

The main aim is to divide the network into several sections, which can indicate different conditions from non-desirable to desirable ones. As the network moves toward non-desirable states, FB-TCP employs a conservative approach to relieve the conditions. This degree of conservativeness is based on how bad the network's current situation is. On the other hand, as the network moves toward desirable states, the protocol takes an aggressive mechanism in increasing the sending rate, which the level of the aggressiveness depends on to what extent the network's current condition is good. There is a direct relation between the accuracy of the protocol and the number of clusters so that as the number of clusters increases, the protocol functions more precisely.

Diff is calculated by deploying (2):

$$Diff = currentCwnd - targetedCwnd \tag{2}$$

where currentCwnd is the value of the sending rate at the moment, and targetedCwnd is the optimal value of cwnd, i.e., the minimum value that we need to set cwnd to attain the available throughput. The targetedCwnd has a direct correlation to RTT and current sending rate, which is calculated based on (3):

$$targetedCwnd = currentCwnd^{*} CSI \tag{3}$$

where CSI is calculated based on (4):

$$CSI = baseRtt/minRtt \tag{4}$$

where baseRtt is the minimum value for a connection and minRtt is the minimum value for a congestion window. CSI is always between zero and one; as a result, it can function as a Fuzzy membership, and based on the obtained values between zero and one, the Fuzzy rules can be set. This parameter is one of the principal leverages in dividing the network into several clusters.

These values are exploited in different parts of the protocol to help FB-TCP adjust the sending rate adequately. All these parameters are used in the protocol's congestion avoidance phase after exiting the slow start phase.

The Convergence phase is initiated when the current sending rate is lower than the estimated upper bound. This phase's primary aim is to ramp up to the highest available sending data rate and utilize the full potential of the network. Situations such as NLoS to LoS transitions, in which a quick increment in the sending rate is essential, can benefit from this phase. Moreover, it helps to prevent bandwidth wastage and save time in recovering from low data rates.

On the other hand, the Divergence phase is commenced when the sending rate is higher than the estimated upper bound. In this case, the protocol strives to use the available resources in the network and discover more bandwidth.

The Convergence and Divergence phase's ultimate goal is to create a framework for FB-TCP to function around the highest available sending rate in a way that can satisfy BDP for the packets in-flight. The first output of this mechanism is tackling the high fluctuation for the congestion window in a way that by approximating the highest available sending rate and trying to accommodate it, we can adjust cwnd more elaborately. Secondly, by keeping the cwnd a slight factor of the BDP, the protocol can attain a throughput close to the saturated value. Thirdly, by quick and attentive reactions to different conditions such as NLoS states, FB-TCP can prevent RTT increment in the network to avert bufferbloating issue.

Moreover, while the protocol sets the sending rate around the highest estimated available data rate, it does not increase the cwnd blindly because it has a clear insight from the network's current condition. In contrast, it tries to take careful steps in adjusting the congestion window; thus, it can achieve high throughputs along with low average congestion window size through a constant functionality. Finally, because FB-TCP controls the sending rate based on the feedbacks it gets from the network and is a model-based TCP, it is immune to random packet drops.

A. CONVERGENCE PHASE

Being in this phase means that the sending rate is less than maxCwnd, so we take an aggressive mode to reach the highest possible sending rate in fast paces or control the sending rate in an aggressive approach. Convergence and Divergence are parts of FB-TCP's congestion avoidance phase and are initiated when the slow start is finished.

Conventional TCPs double their sending rate when an acknowledgment is received during the slow start phase. However, the FB-TCP functionality is close to BBR and Vegas in this phase, as it doubles the sending rate in every RTT change. This approach might be slightly slower than the doubling approach of the conventional TCPs in every ACK. However, this mechanism helps FB-TCP to probe the bandwidth more appropriately. After reaching the cwnd to 900, which is roughly double the conventional TCP's slow start threshold, the congestion avoidance phase is initiated. The reason for choosing this value is to exit the slow start soon but not so fast that the protocol cannot examine the network. Because of that, we have decided to set its size twice the conventional one. This value can be used as a general threshold in FB-TCP in different use cases, scenarios and is an optimal value that could be achieved through extensive simulations among different topologies and layouts.

By triggering the congestion avoidance phase, if (5) is correct, the Convergence phase starts:

$$currentCwnd <= maxCwnd \tag{5}$$

We also need another parameter that can help us adjust the sending rate when moderate tuning is required. One of the appropriate approaches can be deploying the relationship between the targetedCwnd and currentCwnd. Thus, we can have an estimation of how far we are from the optimal value

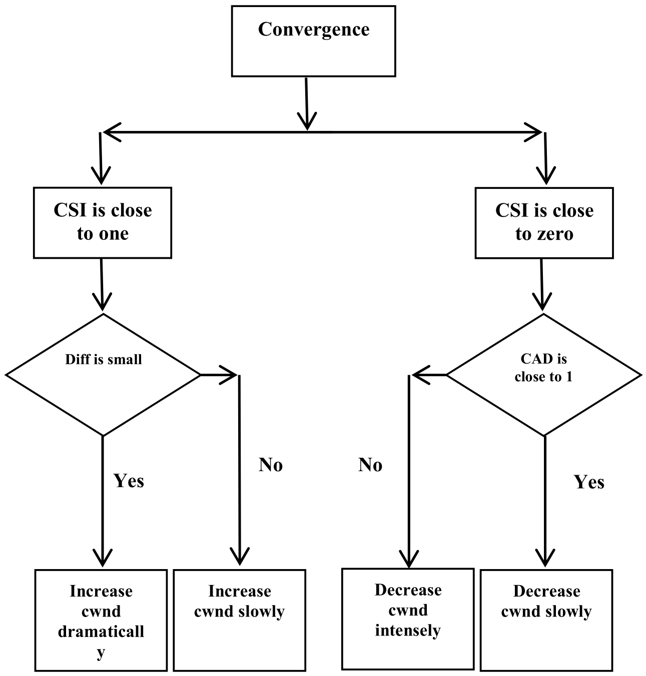

FIGURE 2. How the Convergence phase functions.

and to what extent the protocol is functioning poorly. If this value is close to one, it shows that the protocol is performing well. In contrast, being close to zero is not a good sign. By deploying CAD, we can adjust the sending rate more appropriately. This value is calculated based on (6):

$$CAD = targetedCwnd/currentCwnd \tag{6}$$

CAD is also between zero and one all the time and can be used as another Fuzzy membership function to help FB-TCP decide properly.

The Convergence phase's primary goal is to utilize the available high bandwidth of 5G mmWave networks when the network is empty by increasing the sending data rate in fast paces. As can be seen in Figure 2, when CSI is close to one, it is a sign that the protocol can increase the sending rate. In this case, if diff is a minor value, it indicates that the network is empty and its full potential is not utilized; thus, the sending rate can be increased dramatically. On the other hand, if diff is not close to one, it is a manifestation of slight underutilization, so that the sending rate will be increased negligently.

On the other hand, by using CSI and CAD, FB-TCP can have proper reactions to NLoS and congestion states when the cwnd value is below the estimated upper bound. In this case, when CSI is close to zero, it shows that the network's condition is getting worse. FB-TCP measures the intensity of the worseness based on CAD. As a result, if CAD is close to one, the protocol takes it as a worse situation but no very severe. However, when CAD is not close to one, it indicates that the network situation is heavily poor, and a drastic decrement in the sending rate is needed.

TABLE 1. How the convergence phase functions- sub-phase 1.

Sub-Phase	cwnd adjustment
C1. if ((0.98 <= CSI) && (CSI <= 1) && (Diff<=10))	currentCwnd= currentCwnd + a
C2. if ((0.98 <= CSI) && (CSI <= 1) && (Diff >10) && (Diff <= 15))	currentCwnd= currentCwnd + b
C3. if ((0.98 <= CSI) && (CSI <= 1) && (Diff > 15) && (Diff <= 20))	currentCwnd= currentCwnd + c
C4. if ((0.98 <= CSI) && (CSI <= 1) && (Diff > 20) && (Diff <= 30))	currentCwnd= currentCwnd + d
C5. if ((0.98 <= CSI) && (CSI <= 1) && (Diff > 30))	currentCwnd=currentCwnd + e
C6.if ((0.95 <= CSI) && (CSI <= 0.98) && (CAD >= 0.95))	currentCwnd=currentCwnd + f
C7. if ((0.95 <= CSI) && (CSI <= 0.98) && (CAD < 0.95))	currentCwnd=currentCwnd + g
C8. if ((0.7 <= CSI) && (CSI < 0.98) && (CAD >= 0.95))	currentCwnd=currentCwnd + h
C9. if ((0.7 <= CSI) && (CSI < 0.98) && (CAD < 0.95))	currentCwnd=currentCwnd + i

In a nutshell, the protocol tries to keep the sending, rate a slight portion of the BDP, which helps prevent unnecessary buffer overflows and reduce the RTT value close to the minimum possible one. In addition to RTT, this mechanism stably adjusts the cwnd size and alleviates the fluctuations.

Consequently, one of the Fuzzy primary memberships that FB-TCP employs in its initial steps to divide the network into different sections is CSI. When the value of CSI is decreasing, it shows that minRtt is increasing. The CSI value can be used as a sign of NLoS states, and in combination with CAD, they can help distinguish LoS states from NLoS and congestion ones. The reason is that when a UE is in NLoS states, packets are enqueued in buffers, and the RTT increases, which leads to high CSI values. A similar conclusion is correct for CAD, in which the difference between targetedCwnd and currentCwnd increases in NLoS states.

The Convergence phase functions are based on the rules in Table 1, Table 2, Table 3, and Table 4. When cwnd is lower than the upper bound, i.e., Convergence phase, we should notice that Diff is deployed to control the protocol's aggressiveness in the beginning sub-phases of the Convergence, and CAD is for slowing down. As the value of Diff is close to zero, it shows ideal conditions; thus, FB-TCP can increase

TABLE 2. Deployed parameters in the increasing sub-phases of the convergence.

Parameter	Value
a	120
b	100
c	70
d	60
e	50
f	40
g	30
h	25
i	20

the sending rate rapidly. However, if Diff is getting far from zero, it indicates a situation that the protocol can increase the sending rate but moderately till 0.98 <= CSI.

From C1 to C9 are the increasing sub-phases for the Convergence phase. Throughout these phases, FB-TCP tries to increase the cwnd in an aggressive way, which the aggressiveness of the protocol can depend on the deployed approach. This aggressiveness is determined by setting the values for *a* to *i*. As these parameters set to high values, the protocol will be more aggressive, but if they are set to smaller ones, the protocol decreases its aggressiveness. The primary goal of the selected values for FB-TCP is to make the protocol suitable for different scenarios. As a result, we have conducted numerous simulations in various conditions to determine the optimal values as shown in Table 2. However, we believe that the protocol is dynamic, and values can be changed in order to adapt to different scenarios. As an example, if the RTT is the most important KPI, so by tuning the values, the minimum RTT can be attained at the cost of throughput.

The increasing sub-phases, i.e., Table 2, are used when the network is in desirable conditions, and the available bandwidth is not utilized to its full potential, such as the switching times from NLoS to LoS states, when the protocol needs to recover its high sending rate quickly and ramps up to the highest available sending rate in the network. As an example, if we look at C1 in Table 1, (0.98 <= CSI) && (CSI <= 1) indicates that minRtt is close to baseRtt, which shows an empty network. Moreover, Diff <= 10 reveals that currentCwnd is near the optimal value of cwnd; thus, another positive sign, so by considering these conditions, it is concluded that the network's state is desirable and large bandwidth is available so that FB-TCP can increase the sending rate drastically.

The rest of the Convergence sub-phases control the sending rate when the network is going toward being congested or when NLoS states happen, i.e., non-desirable states. CSI and

TABLE 3. How the convergence phase functions- sub-phase 2.

SUB-PHASE	cwnd adjustment
C10. if ((0.3 <= CSI) && (CSI < 0.7))	currentCwnd=currentCwnd
C11. if ((0.05 <= CSI) && (CSI < 0.3) && (CAD >= 0.95))	currentCwnd=currentCwnd - j
C11.b. if ((0.05 <= CSI) && (CSI < 0.3) && (CAD >= 0.95)) C11 for more than two RTTs	currentCwnd=currentCwnd - k
C12. if ((0.05 <= CSI) && (CSI < 0.3) && (CAD < 0.95))	currentCwnd=currentCwnd - l
C12.b. if ((0.05 <= CSI) && (CSI < 0.3) && (CAD < 0.95)) C12 for more than two RTTs	currentCwnd=currentCwnd - m
C13. if ((0.0 <= CSI) && (CSI < 0.05) && (CAD >= 0.95))	currentCwnd= n * currentCwnd
C13.b. if ((0.0 <= CSI) && (CSI < 0.05) && (CAD >= 0.95)) C13 for more than two RTTs	currentCwnd= currentCwnd/o
C14. if ((0.0 <= CSI) && (CSI < 0.05) && (CAD < 0.95))	currentCwnd= p * currentCwnd
C14.b. if ((0.0 <= CSI) && (CSI < 0.05) && (CAD < 0.95)) C14 for more than two RTTs	currentCwnd= currentCwnd/q

CAD are used in these phases to adjust the sending rate. As a Fuzzy membership, CAD is a key parameter in adjusting the sending rate because it indicates that the network is moving toward desirable or non-desirable conditions.

If FB-TCP remains for thirty consecutive RTTs in C10, (7) will replace currentCwnd=currentCwnd, i.e., the sending rate will not be kept fixed. Reducing the sending rate will drain the buffers and prevent the network from moving toward non-desirable situations. Thirty has been chosen based on extensive simulations and is an arbitrary number, which can be tuned hinged on the desired tradeoff between RTT and throughput. For validating the sufficiency of our chosen parameters, we have tested them in four different scenarios. Moreover, the door toward attuning the parameters for various scenarios and layouts has been kept open in a way that the protocol has the capability of being altered.

$$currentCwnd = \alpha^{*} currentCwnd \tag{7}$$

where α is for keeping a tradeoff between throughput and RTT, as we increase α, the throughput value will be improved.

TABLE 4. Deployed parameters in the decreasing sub-phases of the convergence.

PARAMETER	Value
j	10
k	20
l	25
	50
n	0.75
o	2
p	0.4
q	3

In contrast, by decreasing α, the RTT value will be enhanced at the cost of throughput. As we want to reduce the sending rate when a user is stuck in this condition, α should be between zero and one. We have used 0.9 for setting α in order to keep high throughput by slightly improving the value of RTT. Moreover, maxCwnd will be reduced by using (8) to lowers the upper bound and reduces the aggressiveness of the protocol when it remains in C10 for more than thirty consecutive RTTs:

$$maxCwnd = \beta^{*} maxCwnd \tag{8}$$

Large β means more aggressiveness, and small β means less aggressiveness for the protocol. As a result, we have used 0.9 for setting β to attain high throughputs through acceptable RTTs. β also can be tuned to be suitable for different use cases based on needs and necessities. For example, by reducing β, we will have an enhancement in the value of RTT at the cost of throughput and vice versa.

We should notice that all the b sub-phases are the time that the protocol remains in the same sub-phase, i.e., the same state, for more than two consecutive RTTs. As an example, when the network is in the C12 sub-phase for two successive RTTs, adjusting the cwnd size will follow the rules in c12.b. These situations indicate that the network's condition is not ideal, and the sending rate should be decreased quickly in order to empty the network; thus, FB-TCP waits for a maximum of two RTTs.

The protocol operates more conservatively during sub-phases, i.e., b-subs, as the values of the parameter indicate for C11.b and Table 4.

We tried to select values that can be generally used in different urban deployment layouts and strived to prove it by extensive simulations. To prove our claim, we run FB-TCP in different layouts and various situations. However, individual TCPs can be sufficient for a particular scenario and show flaws in other ones [46]–[48]. Considering this fact, our protocol's targeted scenario is the urban deployment; nevertheless, it can show sufficiency in other ones.

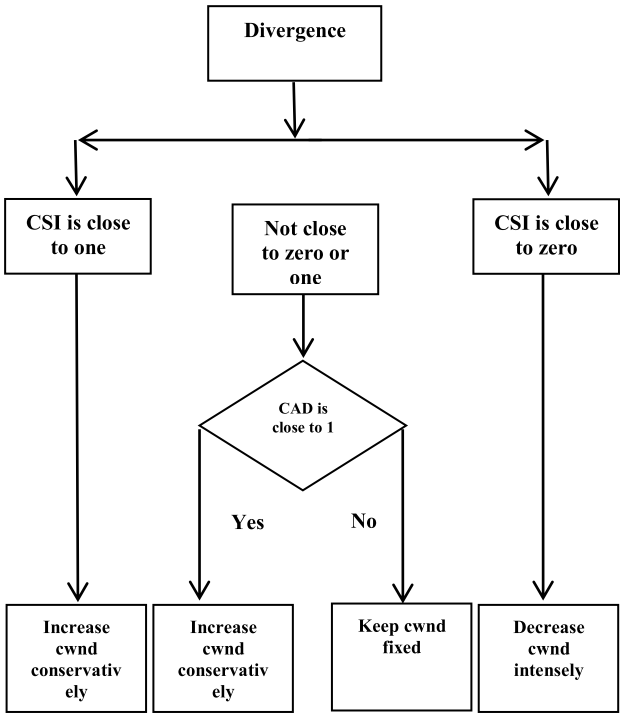

FIGURE 3. How the Divergence phase functions.

For choosing the values that can cover a vast range of urban deployments, shown in Table 2 and Table 4, we have conducted more than 200 simulations, and based on the obtained results, the best ones have been selected. The primary motivation behind choosing these values was satisfying the network's available capacity in different circumstances. The main goal was achieving high throughputs through acceptable RTTs among with preventing cwnd fluctuations. Moreover, the protocol can have stable functionalities through different conditions and is immune to the network's changes.

B. DIVERGENCE PHASE

In contrast to the Convergence phase, the Divergence strives to increase the sending rate conservatively, so it can prevent buffer overflows and also discover more capacity in the network if available. The reason is that cwnd is larger than maxCwnd, i.e., the estimated upper bound for the network, and moving faster can exhaust the buffers. Moreover, when it detects that the network is not functioning in the LoS state, it reacts more intensely in a way that can prevent consecutive packet drops in NLoS or congested states. This approach can drain the buffers, especially when a UE is behind an obstacle, and not having an appropriate strategy can lead to underutilization of the large bandwidth or packet losses.

Figure 3 depicts an overview of the Divergence phase's functionality. The critical parameter in this phase is CSI. When it is close to one, it gives some guarantees to the protocol in increasing the sending data rate in order to find more capacity in the network; however, as this parameter moves closer to zero, it indicates that the network's condition is getting worse and an aggressive reduction in the sending rate is necessary. When CSI is not so close to zero or one, CAD is the key player in adjusting the sending rate, and if it is close to one, FB-TCP can increase the sending rate because it seems that there could be more capacity. In contrast, when it is not close to one, the combination of CSI and CAD shows that the networ's functionality is neither very desirable nor non-desirable; thus, the sending rate can be kept fixed.

TABLE 5. How the divergence phase functions.

SUB-PHASE	cwnd adjustment
D1. if ((0.99 <= CSI) && (CSI <= 1))	currentCwnd=currentCwnd + r
D2. if ((0.7 <= CSI) && (CSI < 0.99) && (CAD >= 0.95))	currentCwnd=currentCwnd + s
D3. if ((0.7 <= CSI) && (CSI < 0.99) && (CAD < 0.95))	currentCwnd=currentCwnd
D4. if ((0.3 <= CSI) && (CSI < 0.7))	currentCwnd=currentCwnd
D5 ((0.05 <= CSI) && (CSI < 0.3))	currentCwnd= t * currentCwnd
D5.b. if ((0.05 <= CSI) && (CSI < 0.3))	currentCwnd=currentCwnd/u
D6 if ((0.0 <= CSI) && (CSI < 0.05))	currentCwnd=currentCwnd /v
D6.b. if ((0.0 <= CSI) && (CSI < 0.05))	currentCwnd=currentCwnd /w

The main constructed framework in this phase aims at creating some clusters that are suitable for the time that FB-TCP operates higher than the estimated upper bound; because of this, sending rate increment is done conservatively. However, when the network is close to non-desirable states, cwnd is decreased based on large factors. Table 5 indicates how the Divergence phase functions.

When FB-TCP remains in D3 and D4 for thirty consecutive RTTs, (9) will be applied instead of keeping the sending rate fix. Selecting thirty RTTs is an arbitrary choice and depends on how aggressive we want to react to situations that RTT is high; we have decided to choose thirty after testing a great number of values.

$$currentCwnd = \gamma^{*} currentCwnd \tag{9}$$

where γ is for keeping a tradeoff between throughput and RTT. As γ becomes larger, the throughput value will be increased. In contrast, by reducing γ, the RTT value will be reduced. We have exploited 0.9 for γ in order to achieve high throughputs through acceptable RTTs. Moreover, maxCwnd

TABLE 6. Deployed parameters in the divergence.

Parameter	Value
r	1
s	1
t	0.75
u	2
v	2
w	4

will be set by using (10) to drain the buffers, where θ equals 0.9 in our simulations:

$$maxCwnd = \theta^{*} maxCwnd \tag{10}$$

D5.b and D6.b indicate that the status of being in this condition has continued for the last two consecutive RTTs, and a more aggressive approach is needed to empty the network and prevent RTT increment and buffer overflows. This can happen by more aggressive reactions, as shown in Table 6.

The main difference between the two phases can be summarized as follow. The Convergence phase's primary aim is to reach the estimated upper bound whenever cwnd is lower than this threshold. Moreover, it can have a proper reaction to different situations such as NLoS states or transitions between different states. The increasing approaches for this phase are more aggressive, however, the recovery can be made through moderate mechanisms as the network is working at lower sending rates than the possible highest one.

On the other hand, the Divergence phase functions more conservatively in increasing the sending rate but intensely in recovering. In the former one, it tries to discover more capacity in the network, and in the latter one, it aims at draining the buffers. By combining these two phases, FB-TCP can have passable reactions to different conditions that a 5G mmWave can have in an urban deployment. It can increase the sending rate when LoS states exist, can have appropriate cwnd values in NLoS states, and show proper reactions to packet drops caused by buffer overflows or random ones. The principal aim of FB-TCP is operating around the maximum available sending rate by preventing buffer overflows in a way that can tolerate packet drops to some levels. The extensive simulations showed that the protocol could achieve these purposes and can outperform NewReno, CUBIC, HighSpeed, and BBR. The code for the FB-TCP is available online [49].

V. METHODOLOGY AND THE SIMULATION PARAMETERS

In this section, we elaborate step-by-step methodology and evaluation process of FB-TCP. To ensure that the new protocol works properly and can outperform conventional TCPs, we have conducted extensive simulations and analyzed various parameters and layouts. There exist different simulation tools such as LENA [50], which is available at [51], or 5G library for MATLAB [52], or another tool proposed by Seoul National University called K-SimNet [53]. Each one of these tools has advantages and disadvantages that make them suitable for distinct situations. However, we have decided to exploit a popular simulation tool called ns3-mmWave [54], [55], which is based on ns-3 (Network Simulator-3) [56], [57]. This module can support and provide various features such as channel model implementation [58], dual connectivity and handover [59], [60], and the possibility of connecting to Direct Code Execution [61] in order to deploy Linux stack TCP/IP. Moreover, it can support different spectrums in the range of 6-100 GHz to cover 3GPP's channel model [62]. A thorough analysis and tutorial of this module can be found in [63]. NS3-mmWave is a simulation tool that has been deployed in various researches such as [1], [6], [8], [32]–[35]. This module has proven its functionality and can be deployed as a powerful simulation tool for 5G mmWave.

After selecting the simulation tool, we have defined the scenarios and topologies that can manifest the operability and performance of FB-TCP in urban deployments. We need to test the new protocol in different circumstances to cover simple to complicated scenarios. We should figure out how FB-TCP reacts to various conditions such as small or big obstacles. Moreover, the protocol's functionality should be tested in static situations when the UE stops behind an obstacle such as a building.

After gathering the results, the final step is to have a thorough comparison between FB-TCP and other TCP variants. In order to satisfy all of the necessities mentioned above, we have decided to have four different scenarios, which can cover various conditions that a UE can have inside a city. It is worth saying that the parameters of the network are the same in different topologies, however, the layouts will be different. In all scenarios, a UE is connecting to a gNB at the height of 15 meters, which working at 28 GHz with a 1 GHz channel bandwidth. This antenna is connecting to a server operating at a 1000 Mbps sending rate. The simulation parameters can be seen in Table 7.

Moreover, we have used four different BERs to emulate situations with large (1.25e-10), moderate (1.25e-9), small (1.25e-8), and zero random packet drops. Random packet drops are one of the misleading sources in inducing the congestion control algorithms in a way that they cannot distinguish various losses from each other and reduce their sending rate even if the network is not congested [11]. Due to these reasons having this type of losses in the network is indispensable.

Moreover, selecting 2.5 MB of the RCL buffer is for satisfying the BDP buffer size in the network. The deployed path loss model is Buildings Obstacle Propagation Loss Model, and the BERs are spanned through the simulation and can occur in LoS or NLoS states. Finally, for simulating the obstacles, we have put some boxes and set their boundaries to mime small and big obstacles.

TABLE 7. Simulation parameters.

Parameter	Value
carrier frequency	28 GHz
bandwidth	1 GHz
outage threshold	-5 dB
TxPower	30 dBm
RLC MaxTxBufferSize	2.5 MB
RLC Acknowledged Mode	Enabled
Hybrid ARQ	Enabled
counter for SINR below threshold events	2
TCP Maximum Segment Size	1400 Bytes
Maximum Transmission Unit	1500 Bytes
TcpSocket maximum transmit buffer size	6400 KB
TcpSocket maximum receive buffer size	6400 KB
Initial TCP RTO	1 second

VI. SIMULATION SCENARIOS AND THE RESULTS

This section incorporates the results for various scenarios and compares them when five variants of TCPs, including NewReno, CUBIC, HighSpeed, BBR, and FB-TCP, are deployed. As mentioned, we have evaluated the functionality of FB-TCP in four different scenarios to ensure the conclusions' validity.

A. SCENARIO ONE

This scenario can assist us in evaluating the functionality of FB-TCP in scenarios that contain short NLoS states. It includes a user standing at a distance of 68 meters from the gNB and starts to move at the speed of 1.5 m/s at the second one. There are ten trees at the height of ten meters with 1.5 meters distance from each other on the user's path that are blocking the communication between the UE and the gNB. The simulation time is twenty seconds, the user starts walking at the second one and will stop at the second twenty. Figure 4 depicts the exploited layout in scenario one.

1) SIMULATION RESULTS FOR SCENARIO ONE

The obtained results in the first scenario confirm that obstacles can create blockage states and make the received SINR (Signal-to-Interference-plus-Noise Ratio) weaker, which is the main reason for TCP's confusion. The value for SINR is depicted in Figure 5. This figure reveals that individual trees can degrade the received signals' strength, and after passing the last tree, the UE is in the LoS state, and a proper connection can be established.

Based on the results, we can see that FB-TCP can have proper reactions to various situations and function better than other TCPs.

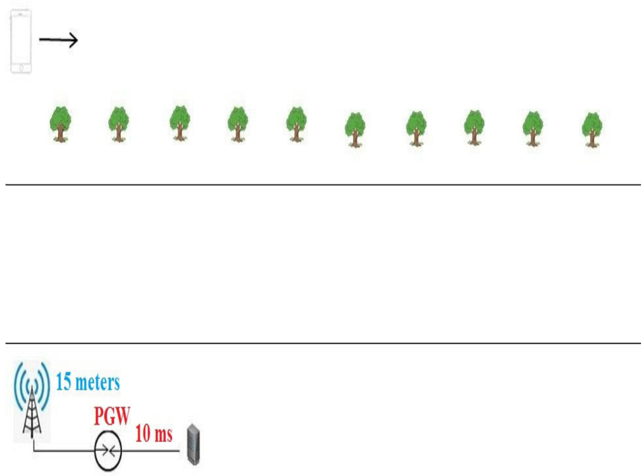

FIGURE 4. Scenario one.

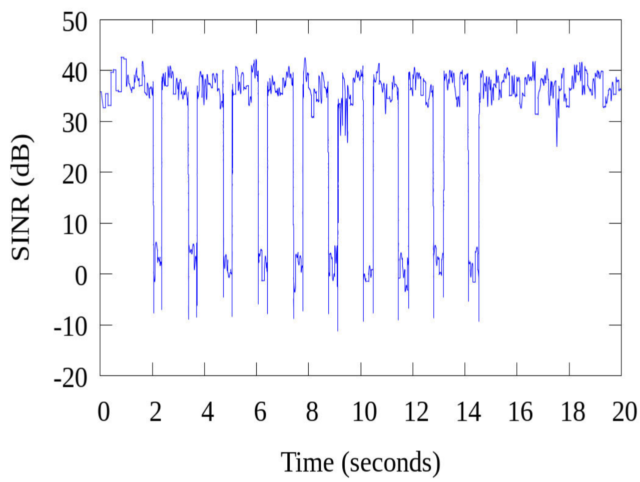

FIGURE 5. SINR fluctuation.

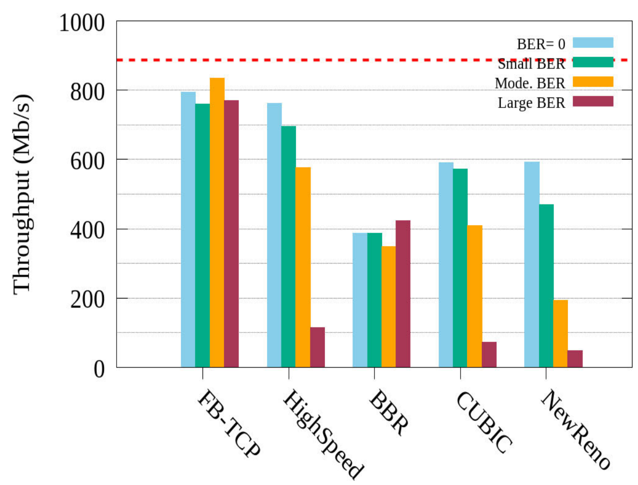

FIGURE 6. Average throughputs for different TCPs.

By looking at Figure 6, we can figure out that FB-TCP can attain higher average throughput compared to the other four TCPs, and it can function close to the saturate UDP value, which equals 886.72 Mbps and is shown by a red

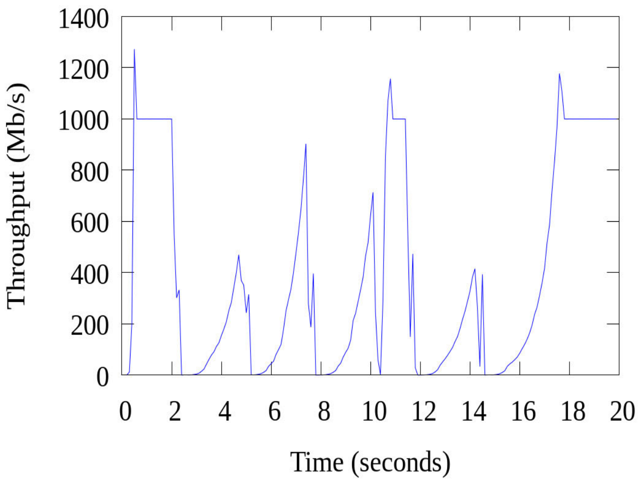

FIGURE 7. Throughput for BBR, BER=0.

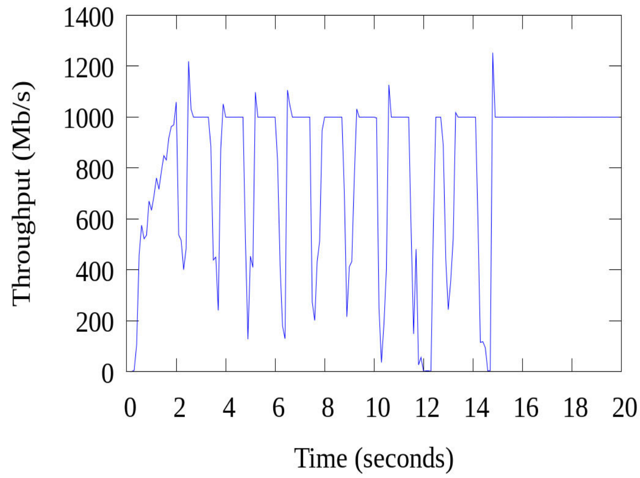

FIGURE 8. Throughput for FB-TCP, BER=0.

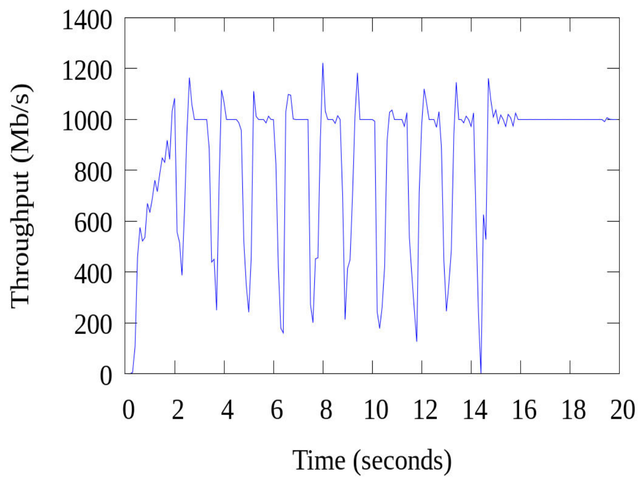

FIGURE 9. Throughput for FB-TCP, Moderate BER.

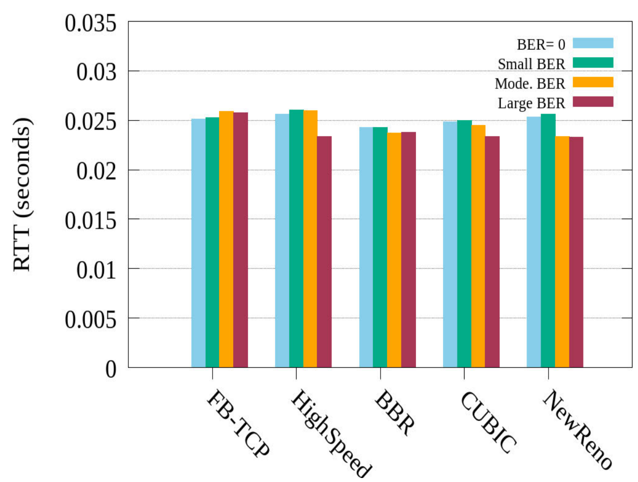

FIGURE 10. Average RTTs for different TCPs.

dashed line in the figure. Moreover, FB-TCP is immune to random packet drops because of its mechanism and can have stable performance through the four different circumstances. Between other TCPs, HighSpeed can work close to FB-TCP when BER is zero. However, this functionality is impaired when random packet drops appear in the network. This conclusion can be veracious for the other two loss-based TCPs, as they lose their performance in the existence of packet drops. The reason is that every single packet drop is assumed as a congestion indicator in loss-based TCPs and can trigger back off mechanism. However, in 5G mmWave networks, packet drops can happen because of other reasons such as blockage or environmental impacts. On the other hand, BBR, based on its estimated bottleneck bandwidth, can have a proper functionality but not close to the saturated UDP value. The principal reason is that NLoS states confuse the protocol in having an accurate estimation, and when the network is in a blockage state, it assumes the network is congested, and the buffers are filled, so initiates the drain phase, probe bandwidth phase, or miscalculate the bottleneck bandwidth, which lead to reducing the sending rate dramatically and empty the buffers as are clear in Figure 7, when the throughput degrades dramatically.

Another interesting point for FB-TCP is its higher average throughput when BER is moderate than the time BER is zero. This can be justified by looking at the instantaneous throughputs for these two circumstances.

Figure 8 shows the throughput for FB-TCP when there are no random packet drops in the network. If we compare this figure to Figure 9, it is evident that there are more drops when BER is zero. The fewer drops of the protocol when BER is moderate might be because of the emptier network than the former one that helps the protocol make appropriate decisions, in which the network is not so congested nor empty.

In terms of RTT, the five TCP variants can work closely as the NLoS states are short, and the time of filling the buffer cannot last for long. Figure 10 indicates the average RTTs for different TCPs in scenario one.

The main improvement of FB-TCP is the same attained RTT values compared to the other protocols by reaching high throughputs.

Based on the average throughputs and RTTs, we can compare FB-TCP and HighSpeed as the best candidate of the

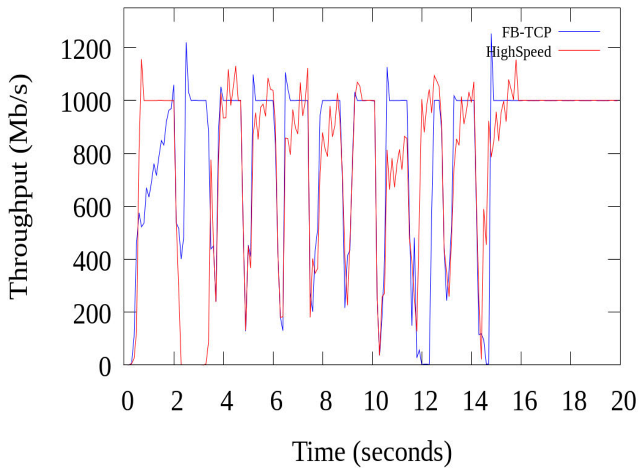

FIGURE 11. FB-TCP and HighSpeed throughput comparison, BER=0.

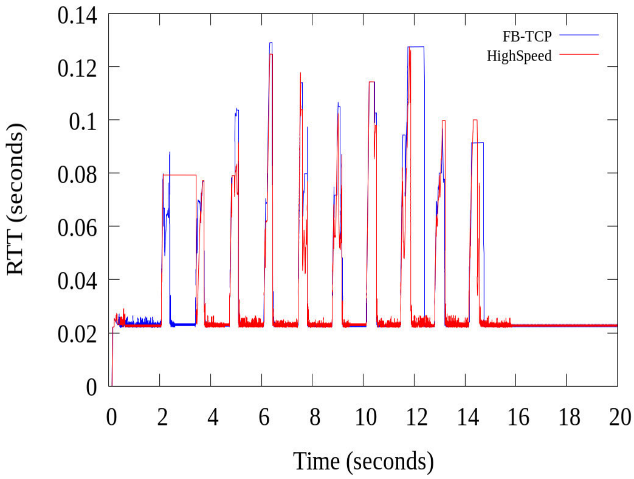

FIGURE 12. FB-TCP and HighSpeed RTT comparison, BER=0.

conventional TCPs to see the differences between them and having a more precise insight for the following scenarios. Figure 11 indicates that FB-TCP has a stable functionality, can react adequately to different situations, and can attain higher throughputs in NLoS states.

By looking at both protocols' beginning steps, we can see that FB-TCP can reach the highest available throughput later than HighSpeed. This is because of the attentive paces that FB-TCP takes and may lead to a little delay in utilizing the full potential but gives a clear insight to the protocol from the network.

The RTT comparison indicates that both protocols have similar functionality as shown in Figure 12. However, in some cases, in NLoS states, FB-TCP can reach lower values. Considering the high throughput value achieved by FB-TCP, this functionality for RTT is acceptable, as it can achieve higher throughput and lower RTT.

The principal cause of FB-TCP's sufficient functionality is behind its proper cwnd adjustments technique, which makes it capable of decision-making based on the current situation of the network. The protocol does not make blind

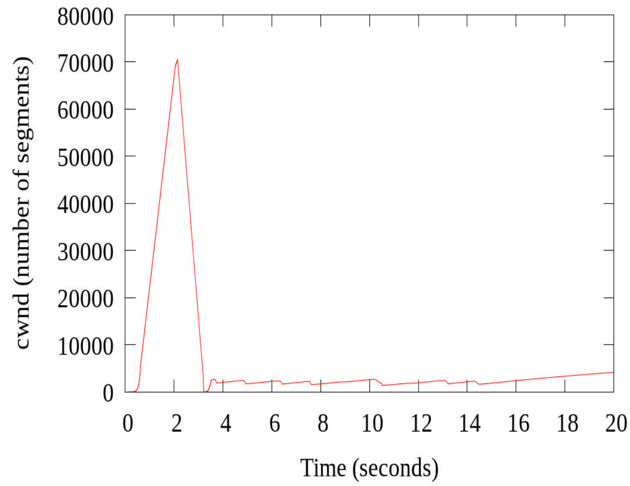

FIGURE 13. cwnd adjustment for HighSpeed, BER=0.

decisions, and it gets help from various parameters to reach proper conclusions. This can be seen in the comparison of the cwnd adjustment of the two protocols. Figure 13 shows how HighSpeed controls the sending rate. The slow start threshold should be a very high number [9], and we should notice that if we use a small slow start threshold, the protocol can not utilize the available bandwidth of the network and is not able to reach the highest sending rate in fast paces because of the premature congestion avoidance initiation. As a result, a large slow start threshold is exploited for conventional TCPs in 5G mmWave networks, and the window scaling option is enabled [2]. As an example, if we set the slow start threshold to its conventional value, i.e., 65500 bytes, the average throughput for HighSpeed decreases from 763.03 Mbps to 649.60 Mbps, for CUBIC from 591.39 Mbps to 532.18 Mbps, and for NewReno from 592.52 Mbps to 113.94 Mbps when BER is zero.

By considering the mentioned reasons, HighSpeed increases its sending rate in the slow start phase, and after entering a NLoS state, due to the high sending rate, a buffer overflow happens, and a packet drop occurs. However, increasing the sending rate in this way may exhaust senders' buffers.

To have a clear view of the HighSpeed's cwnd adjustment, we can look at the time after exiting the slow start and the initiation of the congestion avoidance, as seen in Figure 14.

The figure reveals the aggressiveness of HighSpeed in increasing its sending rate and recovering from losses, which makes it reach higher throughputs compared to other loss-based TCPs.

On the other hand, FB-TCP can control the sending rate sufficiently, as seen in Figure 15. This figure shows that FB-TCP reacts properly to different situations and increases or decreases the sending rate based on the received feedback from the network. One of the intriguing points of the figure is the moderate decrement of the cwnd size in NLoS states. This can help the protocol attain a higher throughput, reduce RTT's sharp increment, and prevent buffer overflows.

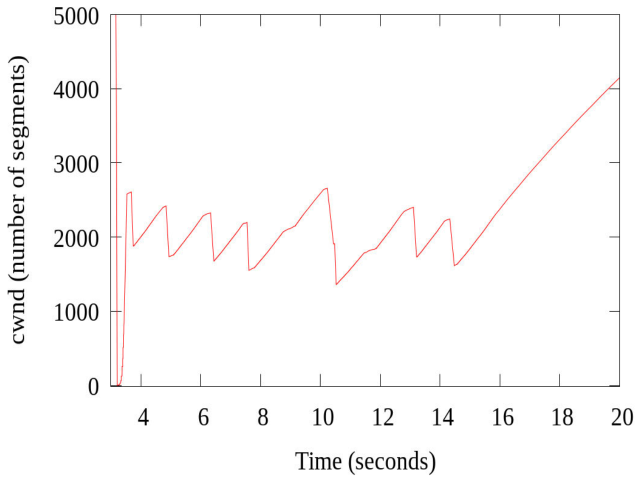

FIGURE 14. How HighSpeed adjust the cwnd in the congestion avoidance phase, BER=0.

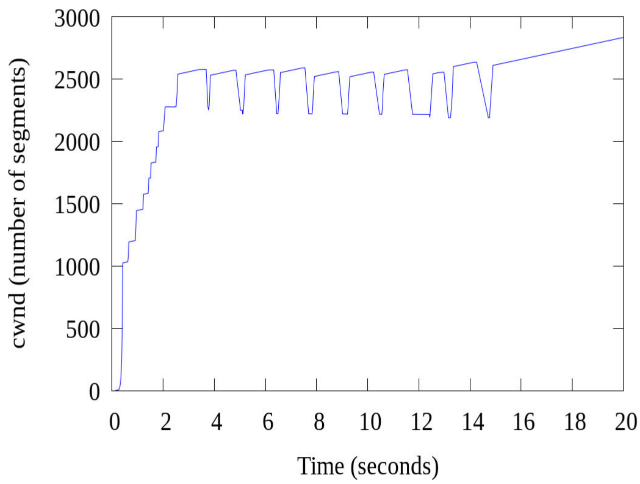

FIGURE 15. How FB-TCP adjust the cwnd, BER=0.

TABLE 8. Average cwnd values comparison of FB-TCP and HighSpeed.

BER	FB-TCP	HighSpeed
zero	2403	31369
Small	2223	31364
Moderate	2303	17771
High	1913	341

FB-TCP strives to estimate the available maximum sending rate at different conditions and move based on this value. Moreover, the Convergence and Divergence phases and their sub-phases succor the protocol to have a clear view of the network and control the sent packets into the network. This mechanism prevents from exhausting senders' buffers and deploys the available space in intermediate buffers more efficiently. For more clarity, we can have a look on Table 8 to see the average values for cwnd in different BERs.

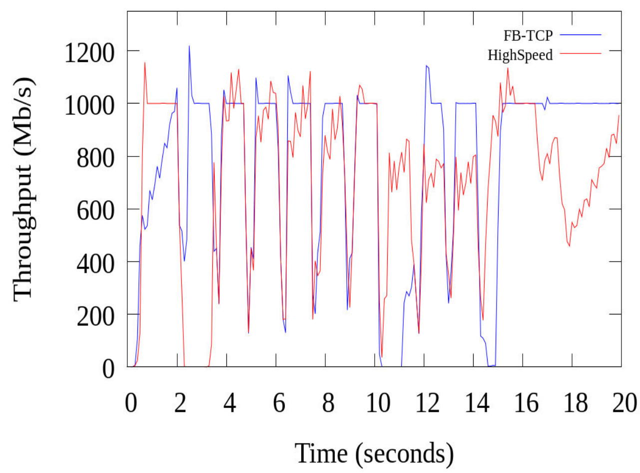

FIGURE 16. FB-TCP and HighSpeed throughput comparison, small BER.

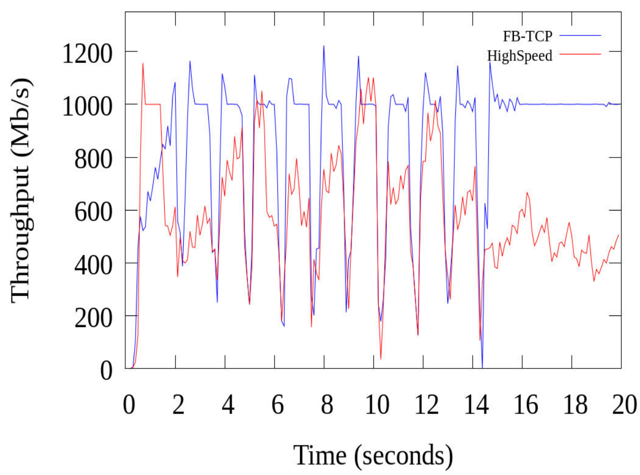

FIGURE 17. FB-TCP and HighSpeed throughput comparison, moderate BER.

This table reveals that HighSpeed increases its sending rate aggressively in a blind way without considering the sender's buffer exhaustion and the network's conditions. However, FB-TCP can attain higher throughputs by considerably low values for its cwnd. Moreover, the model-based mechanism of the protocol makes it capable of tolerating random packet drops.

For more detailed analysis and having a guideline for the other scenarios, we have a look on the throughput of these two protocols in other BER values.

Figure 16 indicates the two protocols' throughput when BER is a small value. In contrast to HighSpeed, FB-TCP is not affected by random packet losses because of its immune mechanism to packet drops.

The degradation in the functionality of HighSpeed can be more intense by increasing the number of packet drops in the network, as depicted in Figure 17.

This figure shows the deficiency of HighSpeed in having proper reactions to random packet drops, which leads to underutilization of the wide available bandwidth in the

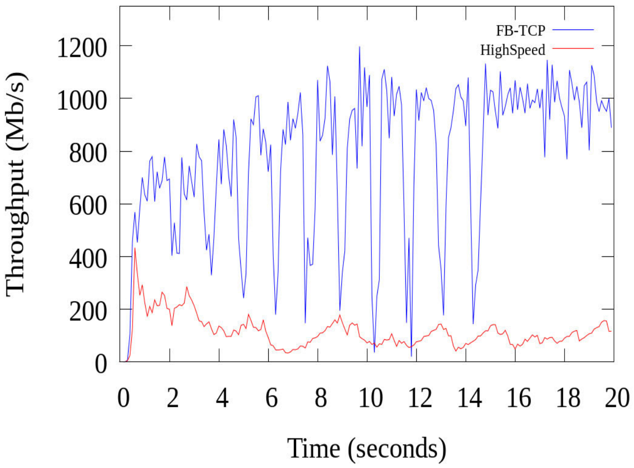

FIGURE 18. FB-TCP and HighSpeed throughput comparison, large BER.

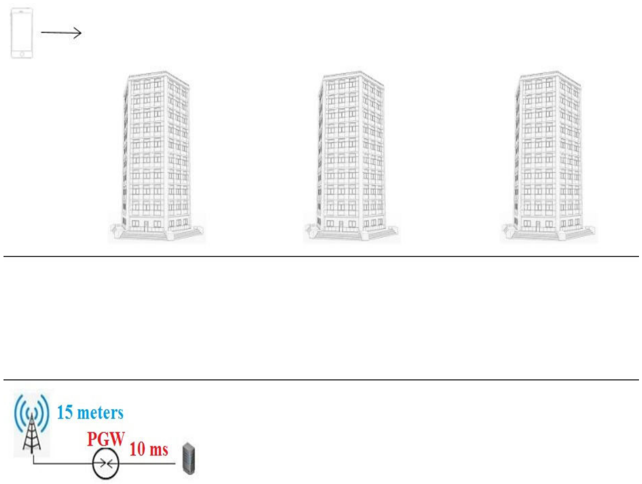

FIGURE 19. Scenario two.

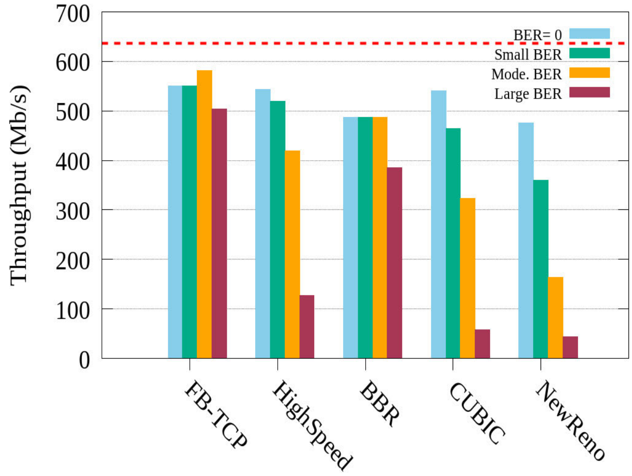

FIGURE 20. Average throughputs for different TCPs.

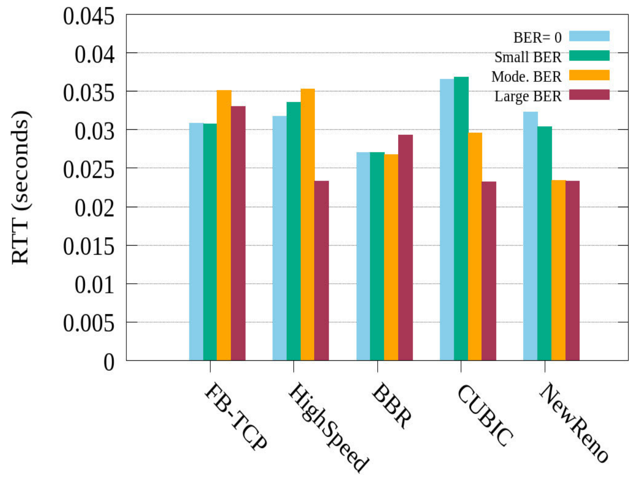

FIGURE 21. Average RTTs for different TCPs.

network. If we increase the BER value to a large number, this insufficiency can be more obvious, as seen in Figure18.

A large number of packet drops can mislead HighSpeed in a way that it loses its functionality and performs in lower throughputs.

B. SCENARIO TWO

This layout's main goal is to evaluate the performance of different protocols when the communication channel can be blocked by large obstacles. The specifications for the UE and the gNB are similar to the previous scenario. However, instead of ten trees, in this scenario, we have three buildings with a width of eight meters and a height of thirty meters that are at a distance of five meters from each other. The simulation time for this scenario is thirty seconds. Figure 19 indicates the deployed layout in the second scenario.

1) SIMULATION RESULTS FOR SCENARIO TWO

The conducted simulations revealed that FB-TCP could also outperform other TCPs in the existence of large obstacles. By looking at Figure 20, it can be seen that FB-TCP is the only protocol that can operate near the saturated UDP value, which equals 636.62 Mbps. Moreover, FB-TCP can retain this high functionality throughout different BERs, relying on its immune mechanism to random packet drops and analyzing the current condition of the network. Similar to scenario one, BBR shows a stable functionality. However, it cannot function close to the saturated value. Between the four TCPs, HighSpeed can attain the best throughput for low BERs, but when the number of packet drops increases dramatically, it loses its functionality.

In terms of RTT, BBR could show a better functionality compared to FB-TCP. However, considering higher throughput values that FB-TCP can attain compared to BBR compensate for this downside. The difference between the throughputs of the two protocols can reach 119.29 Mbps in some cases. Comparing to loss-based TCPs, FB-TCP can attain better RTT values. We should notice that the low RTT values for loss-based TCP in high BERs are because of the low throughput that they achieve. In this case, they send

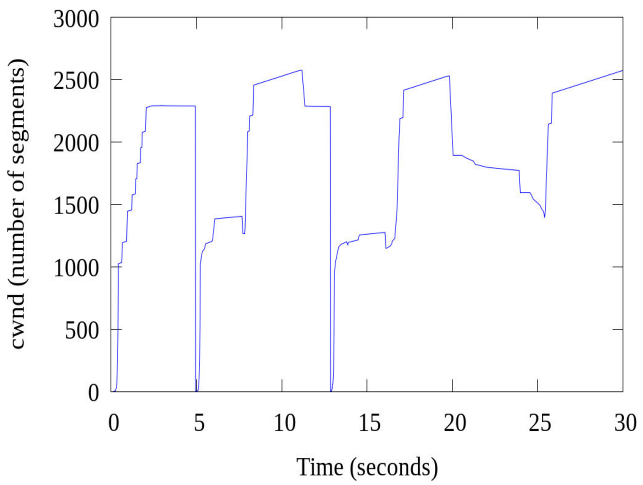

FIGURE 22. FB-TCP cwnd adjustment, BER=0.

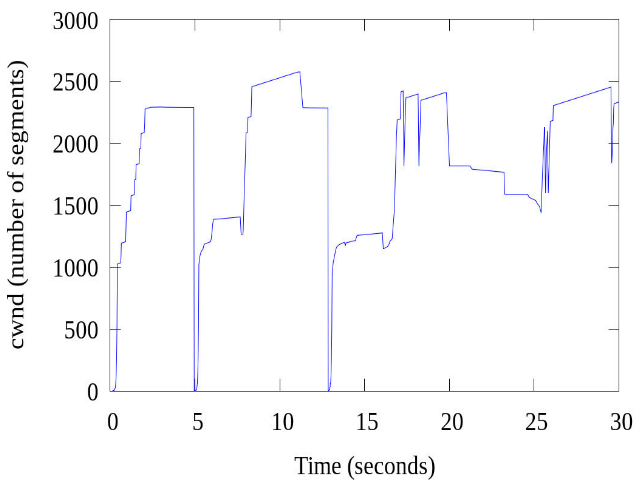

FIGURE 23. FB-TCP cwnd adjustment, small BER.

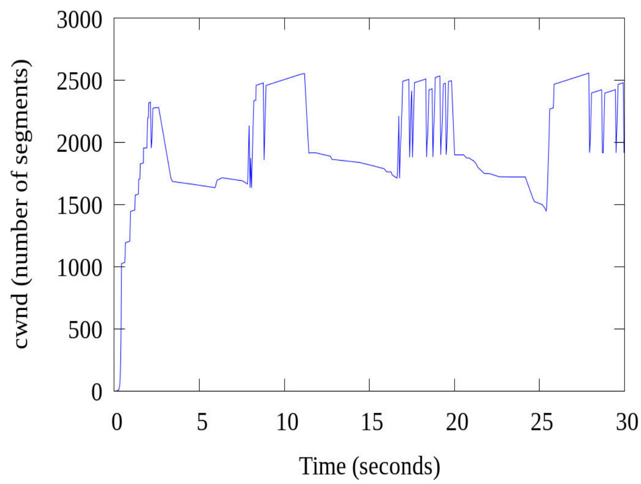

FIGURE 24. FB-TCP cwnd adjustment, moderate BER.

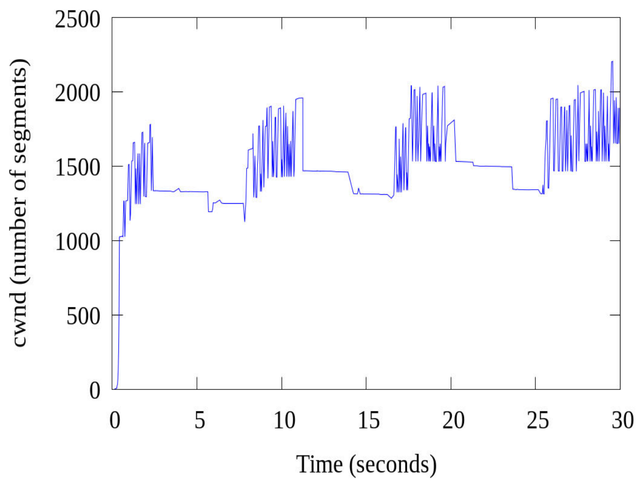

FIGURE 25. FB-TCP cwnd adjustment, large BER.

fewer packets to the network; as a result, long queues are not established in the buffers. Figure 21 indicates average RTT for different TCPs.

In contrast to other TCPs, FB-TCP tries to calculate some parameters that can reflect the network's status, and then, based on these parameters, it decides to adjust the sending rate. For more clarity, we can look at the cwnd adjustment of FB-TCP in the second scenario. Figure 22 indicates the cwnd adjustment for this protocol when there is no packet drop in the network.

This figure shows that FB-TCP can have proper reactions to different situations. It can reduce its sending rate when NLoS states happen in the network, it can recover quickly after finishing these states, and reach high sending rates in fast paces. Moreover, the protocol can find the upper bound of the network step-by-step, as can be seen in the beginning seconds, then when it is necessary, i.e., NLoS to LoS transitions, it can utilize the available bandwidth quickly. The most intriguing fact about Figure 22 is about the last building. After passing the two first buildings, FB-TCP can have a better insight into the network and can control cwnd in a way that by reaching the third building, no buffer overflow happens, which prevents unwanted packet losses. Instead of that, it reduces the sending rate a little sharper to drain the network.

Figure 23 depicts the cwnd adjustment for FB-TCP when a small number of packet drops appear in the network, i.e., a low value for BER.

This figure shows that the functionality of FB-TCP is immune to packet drops because of its model-based congestion avoidance mechanism

This can be proven by looking at Figure 24 and Figure 25. The first figure shows the cwnd adjustment when BER is a moderate value. The next two figures' appealing point is that when the number of random packet losses increases in the network, the network gets emptier, and it helps FB-TCP to analyze the network more efficiently. This can be emphasized by not having a single RTO triggering in the following two figures.

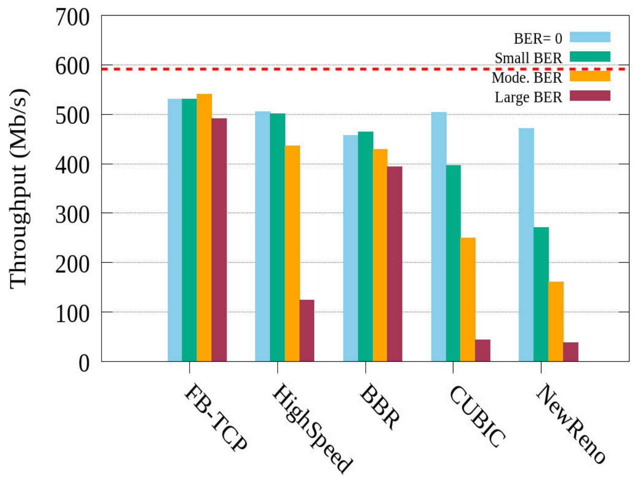

FIGURE 26. Average throughputs for different TCPs.

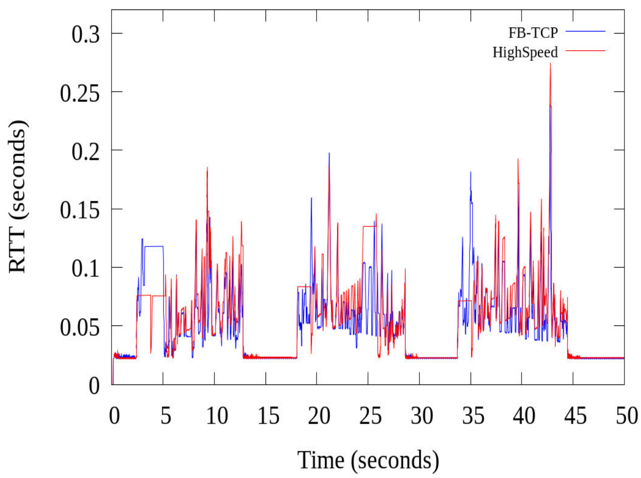

FIGURE 28. FB-TCP and HighSpeed RTT comparison, BER=0.

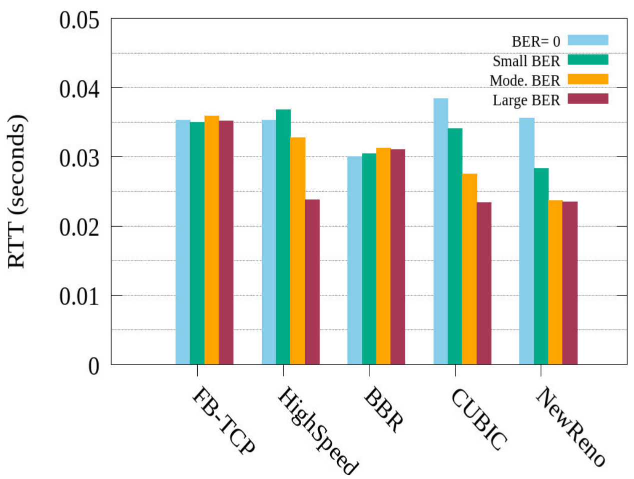

FIGURE 27. Average RTTs for different TCPs.

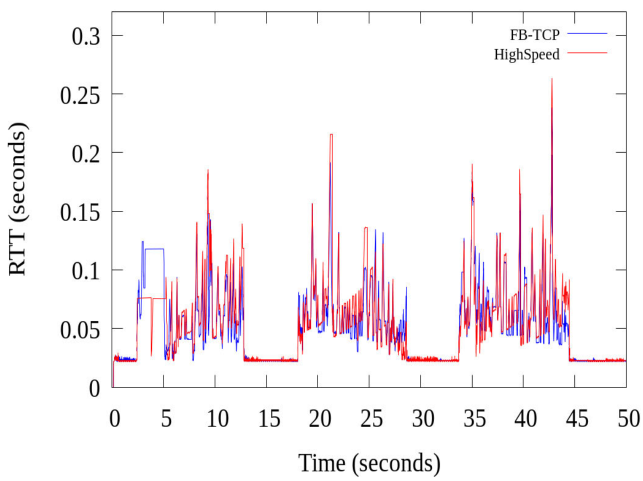

FIGURE 29. FB-TCP and HighSpeed RTT comparison, small BER.

Figure 25 also shows FB-TCP cwnd adjustment when BER is a large number.

To sum up, FB-TCP tries to estimate the upper bound of the network and updates it every 100 ms, or in some other circumstances such as having congestion or NLoS for more than two consecutive RTTs, or having the same sending rate for more than thirty successive RTTs. This mechanism aids the protocol function around the maximum sending rate and adjusts its cwnd size precisely in different conditions.

C. SCENARIO THREE

Scenario three is almost similar to scenario two, with some changes in the layout and some parameters. There are three buildings like the previous testbed in this scenario by increasing the distance between the buildings to eight meters. Moreover, the UE stops behind each building for five seconds to simulate static NLoS states, one of the common conditions that can impair the functionality of TCP over 5G mmWave networks drastically. The main purpose is to emulate a realistic situation inside a city. The simulation time is fifty seconds.

1) SIMULATION RESULTS FOR SCENARIO THREE

The obtained results showed that, like the previous scenarios, FB-TCP could outperform other TCPs. By looking at Figure 26, we can figure out that the new protocol can have more efficient performance in terms of throughput and can work close to the UDP saturated value, which equals 590.66 Mbps.

In addition to throughput, FB-TCP can reach low RTTs, which can be noteworthy by achieving higher throughputs. The average RTT for different TCPs can be seen in Figure 27.

For more clarity, we can have a comparison of RTT between FB-TCP and HighSpeed as the best candidate of the other tested TCPs. When there are no random packet drops, both TCPs' functionalities are almost the same as FB-TCP can reach 0.035264 seconds, and HighSpeed can reach 0.035266. These similarities are shown in Figure 28.

By increasing BER, FB-TCP can have a better functionality both in terms of throughput and RTT. The difference between the average RTT of the two protocols could reach 0.001845 seconds, which is an acceptable enhancement by achieving 29.54 Mbps throughput superiority for FB-TCP.

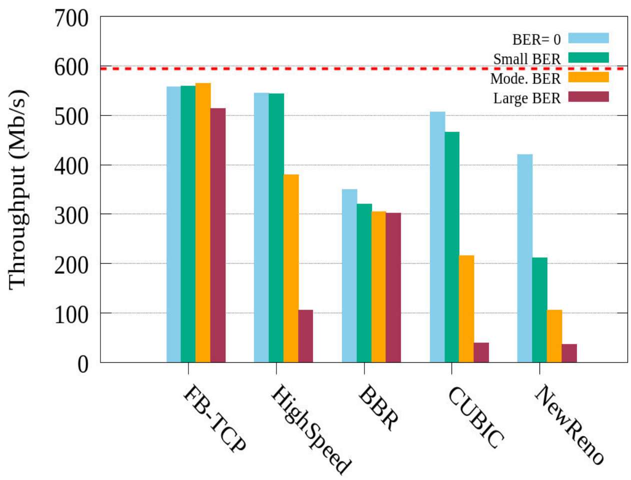

FIGURE 30. Average throughputs for different TCPs.

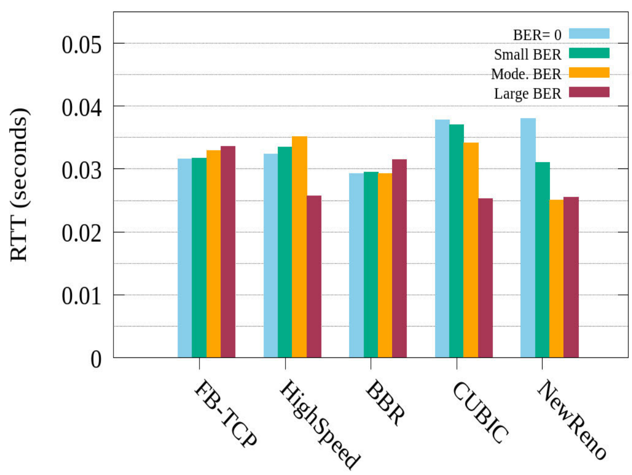

FIGURE 31. Average RTTs for different TCPs.

Figure29 indicates how FB-TCP could attain lower RTTs by having more efficient reactions than HighSpeed in different conditions. We should notice that by increasing the BER, the throughput of HighSpeed declines drastically, and low RTTs can be achieved, which is not worthy of comparing because of the low throughput values for HighSpeed.

D. SCENARIO FOUR

The primary aim of scenario four is to analyze the behavior of TCP in a long connection. As a result, we have set the simulation time to two minutes and put ten large buildings with a distance of eight meters from each other by a width of eight meters to make the topology sophisticated. When the UE is between the fifth and sixth buildings, it stops for ten seconds to mime static LoS situations. Moreover, having a long time for the simulation can assist in investigating the behavior of individual protocols in the presence of a large number of random packet losses. In addition to the previous BERs, we also analyzed the topology under 1.25e-7 and 1.25e-6 bit error rates to see how various protocols functionality under very lossy conditions.

1) SIMULATION RESULTS FOR SCENARIO FOUR

Like the previous scenarios, FB-TCP outperforms other TCPs in terms of throughput, as shown in Figure30. FB-TCP is the only protocol that can function close to the saturated value in all conditions.

By increasing the random packet drops in the network, other TCPs suffer from throughput impairment, especially the loss-based ones. In terms of RTT, FB-TCP can attain a significant superiority compared to other protocols, as shown in Figure 31.

This supremacy is because of the intelligent mechanism that FB-TCP exploits in adjusting the sending rate by dividing the network into different clusters and decides based on the current condition.

Moreover, we analyzed all TCPs in very lossy environments, i.e., BER 1.25e-7 and 1.25e-6, to see how different

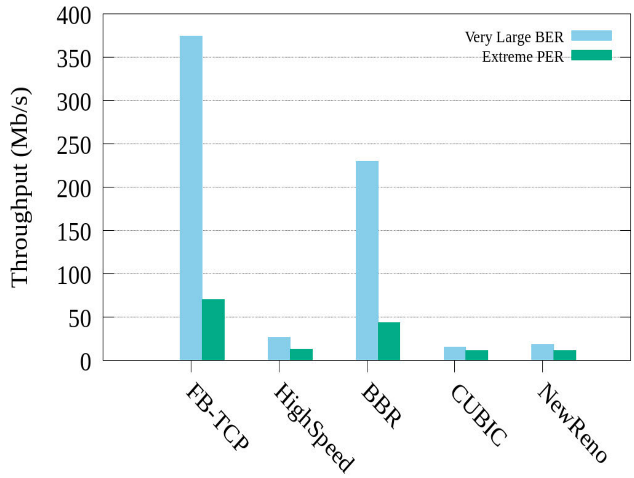

FIGURE 32. Average throughputs for different TCPs.

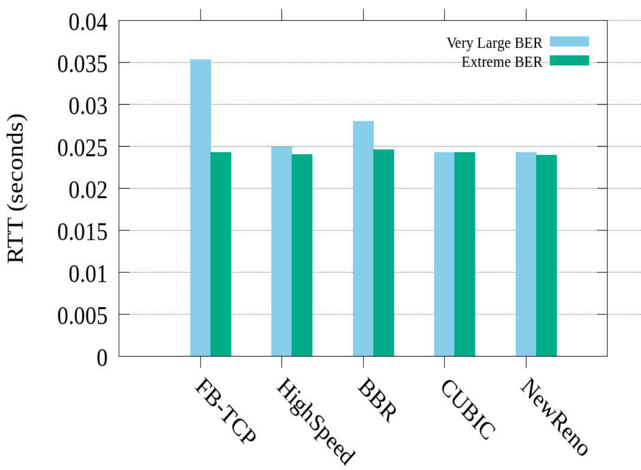

FIGURE 33. Average RTTs for different TCPs.

protocols function in very large and extremely lossy environments. As Figure 32 indicates, the only TCP that can have a proper functionality is FB-TCP. Among the other TCPs, BBR can attain higher throughputs than loss-based TCPs, as they lose their performance as random packet losses increase in the network. The degradation of different TCPs performance in lossy environments was also proved in [35].

The interesting part of higher throughput for FB-TCP is that it can achieve this value through acceptable RTTs, as seen in Figure 33.

To sum up, the new protocol relies on its model-based mechanism and can enhance the transport layer's functionality in 5G mmWave over the urban deployment.

VII. CONCLUSION

Due to the susceptibility of high frequencies, 5G mmWave networks encounter a drawback called blockage. This flaw can impair TCP's functionality by confusing the protocol in adjusting its sending rate adequately, which leads to throughput degradation, RTT increment, and cwnd fluctuation. In this paper, we proposed a new TCP called FB-TCP based on Fuzzy logic to tackle the existing issues. FB-TCP can estimate the upper bound of the network, analyze the current condition, and control the sending rate accurately. The extensive simulation results indicated that FB-TCP could outperform other TCP variants such as NewReno, CUBIC, HighSpeed, and BBR. It can also function close to the UDP saturated value, prevent throughput degradation and RTT increment in NLoS states, and control the cwnd fluctuation. Based on the attained results, FB-TCP can be exploited as one of the appropriate transport layer protocols in 5G mmWave networks, especially in the urban deployments.

REFERENCES

[1] M. Zhang, M. Polese, M. Mezzavilla, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi, "Will TCP work in mmWave 5G cellular networks?" *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 65–71, Jan. 2019, doi: 10.1109/MCOM.2018.1701370.

[2] R. Poorzare and A. C. Auge, "Challenges on the way of implementing TCP over 5G networks," *IEEE Access*, vol. 8, pp. 176393–176415, Sep. 2020, doi: 10.1109/ACCESS.2020.3026540.

[3] M. Giordani, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Toward 6G networks: Use cases and technologies," *IEEE Commun. Mag.*, vol. 58, no. 3, pp. 55–61, Mar. 2020, doi: 10.1109/MCOM.001.1900411.

[4] E. Dahlman, S. Parkvall, and J. Skold, *5G NR: The Next Generation Wireless Access Technology*, 1st ed. Amsterdam, The Netherlands: Elsevier, Aug. 2018, pp. 3–4.

[5] (Mar. 2020). *Brochure: 5G Network Support of Vertical Industries in the 5G Public-Private Partnership Ecosystem*. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2020/03/5PPP_VTF_brochure_v2.1.pdf

[6] W. Na, B. Bae, S. Cho, and N. Kim, "DL-TCP: Deep learning-based transmission control protocol for disaster 5G mmWave networks," *IEEE Access*, vol. 7, pp. 145134–145144, Oct. 2019, doi: 10.1109/ACCESS.2019.2945582.

[7] J. Postel, *Transmission Control Protocol*, document RFC 793, RFC 793, RFC 1122, RFC 3168, RFC 6093, RFC 6528, Sep. 1981. [Online]. Available: https://tools.ietf.org/html/rfc793, doi: 10.17487/RFC0793.

[8] P. J. Mateo, C. Fiandrino, and J. Widmer, "Analysis of TCP performance in 5G mm-wave mobile networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Shanghai, China, May 2019, pp. 1–7, doi: 10.1109/ICC.2019.8761718.

[9] Y. Ren, W. Yang, X. Zhou, H. Chen, and B. Liu, "A survey on TCP over mmWave," *Comput. Commun.*, vol. 171, pp. 80–88, Apr. 2021, doi: 10.1016/j.comcom.2021.01.032.

[10] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-host congestion control for TCP," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 3, pp. 304–342, 3rd Quart., 2010, doi: 10.1109/SURV.2010.042710.00114.

[11] H. Haile, K.-J. Grinnemo, S. Ferlin, P. Hurtig, and A. Brunstrom, "End-to-end congestion control approaches for high throughput and low delay in 4G/5G cellular networks," *Comput. Netw.*, vol. 186, Feb. 2021, Art. no. 107692, doi: 10.1016/j.comnet.2020.107692.

[12] C. Xu, J. Wang, Z. Ma, Y. Cheng, Y. Ni, W. Li, F. Qian, and Y. Li, "A first look at disconnection-centric TCP performance on high-speed railways," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 12, pp. 2723–2733, Dec. 2020, doi: 10.1109/JSAC.2020.3005486.

[13] (Jan. 2021). *Brochure: All Things 5G NR mmWave, an Update on 5G NR Millimeter Wave (mmWave) Network Performance and New Use Cases*. [Online]. Available: https://signalsresearch.com/issue/all-things-5g-nr-mmwave/

[14] M. Giordani and M. Zorzi, "Non-terrestrial networks in the 6G era: Challenges and opportunities," *IEEE Netw.*, vol. 35, no. 2, pp. 244–251, Mar. 2021, doi: 10.1109/MNET.011.2000493.

[15] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, and S. Carson, "Ericsson mobility report," Publisher Fredrik Jejdling, Ericsson, Stockholm, Sweden, Tech. Rep. EAB-20:009174 UEN, Nov. 2020. [Online]. Available: https://www.ericsson.com/4adc87/assets/local/mobility-report/documents/2020/november-2020-ericsson-mobility-report.pdf

[16] *Guidelines for Evaluation of Radio Interface Technologies for IMT-2020*, document ITU-M.2412, ITU, Geneva, Switzerland, Oct. 2017. [Online]. Available: https://www.itu.int/dms_pub/itur/opb/rep/R-REP-M.2412-2017-PDF-E.pdf

[17] J. Navarro-Ortiz, P. Romero-Diaz, S. Sendra, P. Ameigeiras, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "A survey on 5G usage scenarios and traffic models," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 905–929, 2nd Quart., 2020, doi: 10.1109/COMST.2020.2971781.

[18] *3rd Generation Partnership Project; Technical Specification Group Radio Access Network; Study on Scenarios and Requirements for Next Generation Access Technologies, V14.3.0*, document TR 38.913, 3GPP, Sophia Antipolis, France, Jun. 2017. [Online]. Available: http://www.3gpp.org/dynareport/38913.htm

[19] 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, Jul. 2020, doi: 10.3390/s20143880.

[20] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Queue*, vol. 14, no. 5, pp. 20–53, Oct. 2016, doi: 10.1145/3012426.3022184.

[21] L. Brown, G. Ananthanarayanan, E. Katz-Bassett, A. Krishnamurthy, S. Ratnasamy, M. Schapira, and S. Shenker, "On the future of congestion control for the public Internet," in *Proc. 19th ACM Workshop Hot Topics Netw. (HotNets)*, New York, NY, USA, Nov. 2020, pp. 30–37, doi: 10.1145/3422604.3425939.

[22] R. Al-Saadi, G. Armitage, J. But, and P. Branch, "A survey of delay-based and hybrid TCP congestion control algorithms," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3609–3638, 4th Quart., 2019, doi: 10.1109/COMST.2019.2904994.

[23] T. Henderson, S. Floyd, A. Gurtov, and Y. Nishida, *The NewReno Modification to TCP's Fast Recovery Algorithm*, document RFC 6582, Apr. 2012. [Online]. Available: https://tools.ietf.org/html/rfc6582

[24] M. Allman, V. Paxson, and E. Blanton, *TCP Congestion Control*, document RFC 5681, Sep. 2009. [Online]. Available: https://tools.ietf.org/html/rfc5681

[25] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008, doi: 10.1145/1400097.1400105.

[26] S. Floyd, *HighSpeed TCP for Large Congestion Windows*, document RFC 3649, Dec. 2003. [Online]. Available: https://tools.ietf.org/html/rfc3649

[27] D. Scholz, B. Jaeger, L. Schwaighofer, D. Raumer, F. Geyer, and G. Carle, "Towards a deeper understanding of TCP BBR congestion control," in *Proc. IFIP Netw. Conf. (IFIP Networking) Workshops*, Zurich, Switzerland, May 2018, pp. 1–9, doi: 10.23919/IFIPNetworking.2018.8696830.

[28] M. Pieska, A. Kassler, H. Lundqvist, and T. Cai, "Improving TCP fairness over latency controlled 5G mmWave communication links," in *Proc. 22nd Int. ITG Workshop Smart Antennas (WSA)*, Bochum, Germany, Jun. 2018, pp. 1–8.

[29] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, and Z.-L. Zhang, "A first look at commercial 5G performance on smartphones," Sep. 2019, *arXiv:1909.07532*. [Online]. Available: http://arxiv.org/abs/1909.07532

[30] M. Okano, Y. Hasegawa, K. Kanai, B. Wei, and J. Katto, "TCP throughput characteristics over 5G millimeterwave network in indoor train station," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Marrakesh, Morocco, Apr. 2019, pp. 1–6, doi: 10.1109/WCNC.2019.8886119.

[31] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, *TCP Extensions for Multipath Operation with Multiple Addresses*, document RFC 8684, Mar. 2002. [Online]. Available: https://tools.ietf.org/html/rfc8684
[32] M. Polese, R. Jana, and M. Zorzi, "TCP and MP-TCP in 5G mmWave networks," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 12–19, Sep. 2017, doi: 10.1109/MIC.2017.3481348.
[33] R. Poorzare and A. Calveras, "Open trends on TCP performance over urban 5G mmWave networks," in *Proc. 17th ACM Symp. Perform. Eval. Wireless Ad Hoc, Sensor, Ubiquitous Netw. (PE-WASUN)*, Alicante, Spain, Nov. 2020, pp. 85–92, doi: 10.1145/3416011.3424749.
[34] R. Poorzare and A. Calveras, "How sufficient is TCP when deployed in 5G mmWave networks over the urban deployment?" *IEEE Access*, vol. 9, pp. 36342–36355, Mar. 2021, doi: 10.1109/ACCESS.2021.3063623.
[35] M. R. Kanagarathinam, S. Singh, I. Sandeep, H. Kim, M. K. Maheshwari, J. Hwang, A. Roy, and N. Saxena, "NexGen D-TCP: Next generation dynamic TCP congestion control algorithm," *IEEE Access*, vol. 8, pp. 164482–164496, 2020, doi: 10.1109/ACCESS.2020.3022284.
[36] H. Liu, J. Hu, and H. Zhang, "A TCP congestion control mechanism based on fuzzy logic for wireless LANs," in *Proc. 7th Int. Conf. Signal Process.*, vol. 3, Beijing, China, Sep. 2004, pp. 1837–1840, doi: 10.1109/ICOSP.2004.1442086.
[37] J. Arauz, S. Banerjee, and P. Krishnamurthy, "MAITE: A scheme for improving the performance of TCP over wireless channels," in *Proc. IEEE 54th Veh. Technol. Conf. (VTC Fall)*, Atlantic City, NJ, USA, Oct. 2001, pp. 252–256, doi: 10.1109/VTC.2001.956596.
[38] S. Fang, F. Yinglei, L. Yong, and X. Huimin, "A ROD based fuzzy packet loss differentiating algorithm for TCP in the hybrid wired/wireless network," in *Proc. Int. Conf. Syst. Netw. Commun. (ICSNC)*, Tahiti, French Polynesia, 2006, p. 59, doi: 10.1109/ICSNC.2006.15.
[39] Y. Du and F. Du, "Adaptive fuzzy control for wireless network," in *Proc. 2nd Int. Conf. Ind. Mechatronics Autom.*, Wuhan, China, May 2010, pp. 654–658, doi: 10.1109/ICINDMA.2010.5538221.
[40] L. Jia and W. Hu, "A dynamic probability mark method of congestion control based on explicit feedback," in *Proc. 1st Int. Conf. Netw. Distrib. Comput.*, Oct. 2010, pp. 142–145, doi: 10.1109/ICNDC.2010.37.
[41] A. A. Cardoso and F. H. T. Vieira, "Adaptive fuzzy flow rate control considering multifractal traffic modeling and 5G communications," *PLoS ONE*, vol. 14, no. 11, Nov. 2019, Art. no. e0224883, doi: 10.1371/journal.pone.0224883.
[42] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, *Controlled Delay Active Queue Management*, document RFC 8289, Jan. 2018. [Online]. Available: https://tools.ietf.org/html/rfc8289
[43] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, *The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm*, document RFC 8290, Jan. 2018. [Online]. Available: https://tools.ietf.org/html/rfc8290
[44] L. A. Zadeh, "Fuzzy sets," *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.
[45] R. Czabanski, M. Jezewski, and J. Leski, "Introduction to fuzzy systems," in *Theory and Applications of Ordered Fuzzy Numbers*, vol. 356. Cham, Switzerland: Springer, Oct. 2017, pp. 23–43, doi: 10.1007/978-3-319-59614-3_2.
[46] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM*, Stanford, CA, USA, 1988, pp. 314–326.
[47] M. Pieska and A. Kassler, "TCP performance over 5G mmWave links—Tradeoff between capacity and latency," in *Proc. IEEE 13th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Rome, Italy, Oct. 2017, pp. 385–394, doi: 10.1109/WiMOB.2017.8115776.
[48] H. D. Le, C. T. Nguyen, V. V. Mai, and A. T. Pham, "On the throughput performance of TCP cubic in millimeter-wave cellular networks," *IEEE Access*, vol. 7, pp. 178618–178630, Dec. 2019, doi: 10.1109/ACCESS.2019.2959134.
[49] *FB-TCP's Source Codes*. Accessed: Feb. 2021. [Online]. Available: https://github.com/rezapoorzare1/FB-TCP-a-5G-mmWave-Friendly-TCP-for-Urban-Deployments/tree/main
[50] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An open source product-oriented LTE network simulator based on ns-3," in *Proc. 14th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst. (MSWiM)*, Miami Beach, FL, USA, 2011, pp. 293–298, doi: 10.1145/2068897.2068948.
[51] CTTC. (Jan. 2020). *LTE-EPC Network Simulator*. [Online]. Available: http://networks.cttc.es/mobile-networks/software-tools/lena/
[52] MATLAB. *5G Library for LTE System Toolbox*. Accessed: Feb. 2020. [Online]. Available: https://www.mathworks.com/products/5g.html
[53] S. Choi, J. Song, J. Kim, S. Lim, S. Choi, T. T. Kwon, and S. Bahk, "5G K-SimNet: End-to-end performance evaluation of 5G cellular systems," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, Jan. 2019, pp. 1–6, doi: 10.1109/CCNC.2019.8651686.
[54] M. Mezzavilla, S. Dutta, M. Zhang, M. R. Akdeniz, and S. Rangan, "5G mmWave module for the ns-3 network simulator," in *Proc. 18th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Cancun, Mexico, Nov. 2015, pp. 283–290. [Online]. Available: https://dl.acm.org/doi/10.1145/2811587.2811619
[55] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan, and S. Panwar, "TCP dynamics over mmWave links," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Sapporo, Japan, Jul. 2017, pp. 1–6, doi: 10.1109/SPAWC.2017.8227746.
[56] *Network Simulator 3*. Accessed: Feb. 2, 2020. [Online]. Available: https://www.nsnam.org/
[57] T. R. Henderson, M. Lacage, G. F. Riley, C. Dowell, and J. Kopena, "Network simulations with the ns-3 simulator," *SIGCOMM Demonstration*, vol. 14, no. 14, p. 527, 2008.
[58] M. Zhang, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Ns-3 implementation of the 3GPP MIMO channel model for frequency spectrum above 6 GHz," in *Proc. Workshop ns-3 (WNS3)*, 2017, pp. 71–78. [Online]. Available: https://dl.acm.org/doi/10.1145/3067665.3067678
[59] M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Improved handover through dual connectivity in 5G mmWave mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2069–2084, Sep. 2017, doi: 10.1109/JSAC.2017.2720338.
[60] M. Polese, M. Mezzavilla, and M. Zorzi, "Performance comparison of dual connectivity and hard handover for LTE-5G tight integration," in *Proc. 9th EAI Int. Conf. Simulation Tools Techn. (SIMUTOOLS)*, Prague, Czech Republic, Aug. 2016, pp. 118–123. [Online]. Available: https://dl.acm.org/doi/10.5555/3021426.3021445
[61] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, and W. Dabbous, "Direct code execution: Revisiting library OS architecture for reproducible network experiments," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol.*, Santa Barbara, CA, USA, Dec. 2013, pp. 217–228. [Online]. Available: https://dl.acm.org/doi/10.1145/2535372.2535374
[62] *Study on Channel Model for Frequency Spectrum Above 6 GHz, V14.2.0*, document TR 38.900, 3GPP, Sophia Antipolis, France, 2017.
[63] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-end simulation of 5G mmWave networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2237–2263, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2828880.

REZA POORZARE received the B.S. and M.S. degrees in computer engineering from Islamic Azad University, Iran, in 2010 and 2014, respectively. He is currently pursuing the Ph.D. degree in network engineering with the Universitat Politècnica de Catalunya. His research interests include 5G, mmWave, TCP, wireless mobile networks, and artificial intelligence.

ANNA CALVERAS AUGÉ was born in Barcelona, Spain, in 1969. She received the Ph.D. degree in telecommunications engineering from the Universitat Politècnica de Catalunya, Spain, in 2000. She is currently an Associate Professor with the Wireless Networks Group (WNG), Department of Computer Networks, Universitat Politècnica de Catalunya. She has been involved in several national and international research or technology transfer projects. She has published in many international and national conferences and journals. Her research interests include design, evaluation, and optimization of communications protocols and architectures for cellular, wireless multi-hop networks, *ad-hoc* networks, wireless sensor networks, the Internet of Things, and application domains, such as smart cities, building automation, satellite, and emergency environments.