UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Departament d'Enginyeria Telemàtica

Contribution to Reliable End-to-End Communication over 5G Networks Using Advanced Techniques

Ph.D. Thesis in Network Engineering

PhD Candidate: Reza Poorzare

PhD Advisor: Dr. Anna Calveras Augé

Date: January 2022

Abstract

5G cellular communication, especially with its hugely available bandwidth provided by millimeter-wave, is a promising technology to fulfill the coming high demand for vast data rates. These networks can support new use cases such as Vehicle to Vehicle and augmented reality due to its novel features such as network slicing along with the mmWave multi-gigabit-per-second data rate. Nevertheless, 5G cellular networks suffer from some shortcomings, especially in high frequencies because of the intermittent nature of channels when the frequency rises. Non-line of sight state is one of the significant issues that the new generation encounters. This drawback is because of the intense susceptibility of higher frequencies to blockage caused by obstacles and misalignment. This unique characteristic can impair the performance of the reliable transport layer widely deployed protocol, TCP, in attaining high throughput and low latency throughout a fair network. As a result, the protocol needs to adjust the congestion window size based on the current situation of the network. However, TCP cannot adjust its congestion window efficiently, which leads to throughput degradation of the protocol. This thesis presents a comprehensive analysis of reliable end-to-end communications in 5G networks and analyzes TCP's behavior in one of the 3GPP's well-known scenarios called urban deployment. Furthermore, two novel TCPs based on artificial intelligence have been proposed to deal with this issue. The first protocol uses Fuzzy logic, a subset of artificial intelligence, and the second one is based on deep learning. The extensively conducted simulations showed that the newly proposed protocols could attain higher performance than common TCPs, such as BBR, HighSpeed, Cubic, and NewReno in terms of throughput, RTT, and sending rate adjustment in the urban scenario. The new protocols' superiority is achieved by employing smartness in the congestions control mechanism of TCP, which is a powerful enabler in fostering TCP's functionality.

Acknowledgments

The thesis is not only the pinnacle of my research effort during the last years but also the upshot of triumph against hurdles and impediments on the path of research/academic career. In this regard, I would like to give my sincere appreciation to my supervisor professor Anna Calveras Augé, who has never let me down in difficulties and motivated me during this long quest. It was my pleasure to share incredible and unforgettable moments with her, memories that will abide in my mind forever. I am also grateful because of the truth that she has put on me and never let my hand be alone throughout my thesis, which would not be born without her help.

I also would like to thank my family and best friend because of their unconditional support during my Ph.D., I feel grateful for being abetted by them through this tough path. Moreover, I would like to thank Dr. Xavier Hesselbach for his guidance as the doctorate program coordinator and Aurora Rubio Rodriguez for her patience in responding to my questions.

Finally, I am going to thank all my friends and people whom I have met during this journey, assisted me, and shared indelible memories.

List of Contributions

The following publications have been produced during this thesis elaboration.

Journals:

- R. Poorzare and A. Calveras, "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.
- 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.
- R. Poorzare and A. Calveras, "FB-TCP: A 5G mmWave Friendly TCP for Urban Deployments," *IEEE Access*, vol. 9, pp. 82812-82832, Jun. 2021, DOI: 10.1109/ACCESS.2021.3087239.
- Submitted to IEEE/ACM Transactions on Networking: R. Poorzare and A. Calveras, "Deep Learning TCP for Mitigating NLoS Impairments in 5G mmWave,".

Conferences:

- R. Poorzare and A. Calveras, "Open Trends on TCP Performance over Urban 5G mmWave Networks," in *PE-WASUN'20: 17th ACM Symposium Performance Eval. Wireless Ad Hoc, Sensor, & Ubiquitous Netw. Proceedings*, Alicante, Spain, Nov. 2020, pp.85-92. DOI: https://doi.org/10.1145/3416011.3424749.

Codes for the new TCPs:

- FB-TCP: https://github.com/rezapoorzare1/FB-TCP-a-5G-mmWave-Friendly-TCP-for-Urban-Deployments
- DB-TCP: https://github.com/rezapoorzare1/DB-TCP

Contents

Contents

List of Figures

List of Tables

List of Abbreviations

5G	Fifth Generation
5GCN	Fifth Generation Core Network
AI	Artificial Intelligence
AIMD	Additive Increase Multiplicative Decrease
AM	Acknowledged Mode
ANN	Artificial Neural Networks
AP	Access Point
AQM	Active Queue Management
ARQ	Automatic Repeat reQuest
BER	Bit Error Rate
BBR	Bottleneck Bandwidth and Round-trip
BDP	Bandwidth-Delay Product
CA	Congestion Avoidance
CC	Congestion Control
CDN	Content Delivery Network
cwnd	congestion window
DB-TCP	Deep learning-Based TCP
DL	DownLink
DL-TCP	Deep-Learning TCP
DNN	Deep Neural Network

DOOR	Detection of Out-of-Order and Response
eMBB	enhanced Mobile Broadband
EPC	Evolved Packet Core
FB-TCP	Fuzzy-Based TCP
FL	Federated Learning
FSO	Free-Space Optical
GCP	Google Cloud Platform
gNB	gNodeB
HARQ	Hybrid Automatic Repeat reQuest
HetNet	Heterogeneous Network
HPN	High Power Node
HSTCP	HighSpeed-TCP
IMT	International Mobile Telecommunications
IoT	Internet of Thing
ITU-R	International Telecommunication Union- Radiocommunication
KPI	Key Performance Indicator
LFN	Long Fat Network
LoS	Line of Sight
LPN	Low Power Node
LP-TCP	Loss Predictor TCP
MANET	Mobile Ad hoc NETwork
MEC	Mobile Edge Computing

ML	Machine Learning
mMTC	massive Machine Type Communication
MSS	Maximum Segment Size
MTU	Maximum Transmission Unit
NB-IoT	NarrowBand-IoT
NFV	Network Function Virtualization
NLoS	Non-Line of Sight
NPN	Non-Public Network
NR	New Radio
NSA	Non-Stand Alone
PDCP	Packet Data Convergence Protocol
PER	Packet Error Rate
QoE	Quality of Experience
QUIC	Quick UDP Internet Connections
RAN	Radio Access Network
RLC	Radio Link Control
RNN	Recurrent Neural Network
RRC	Radio Resource Control
RTO	Retransmission Time-Out
RTT	Round Trip Time
SA	Stand alone
SACK	Selective Acknowledgement

SDN	Software-Defined Networking
SINR	Signal-to-Interference-plus-Noise Ratio
SNR	Signal to Noise Ratio
SON	Self-Organized Network
SP-TCP	Single-Path TCP
TCP	Transmission Control Protocol
TD-FR	Time Delayed Fast Recovery
TSC	Time-Sensitive Communication
UDP	User Datagram Protocol
UE	User Equipment
UL	UpLink
UM	Unacknowledged Mode
URLLC	Ultra-Reliable Low-Latency Communication
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything

1 INTRODUCTION

Due to the rise in demand for higher data rates by appearing new features and services, the necessity for increasing the bandwidth in new generation mobile networks is inevitable. As an indicator, it can be said that in the first quarter of 2021, seventy million new users started to utilize 5G making the number reach 290 million until the end of the quarter. Furthermore, at the end of 2021, this number will grow up to 580 million. By 2026, The largest monthly average mobile traffic will be for North America by reaching 84 percent of its subscribers employing 5G [1]. Generally, the motivation behind this high growing demand can be categorized into three groups, enhanced device capabilities, cheaper data plans, which lead to affordable services, and an increment in data-incentive content.

By transition from 4G to 5G, the transmission rate increases around 1000 times, and 5G is expected to handle around forty percent of 8.8 billion mobile communication devices in 2026, with around 3.5 billion users [1]. This prediction has been improved compared to the previous one as the impact of the Covid-19 pandemic is relieving. The pandemic could affect the speed of mobile telecommunication coverage progress and delayed some spectrum auctions. By expanding 5G networks, more than sixty percent of the world population will go under the coverage in 2026, which was around 5 percent in late 2019. It is interesting to say that Switzerland had a significant share of this coverage by providing 5G networks in more than 90 percent of the country at the end of 2019 [1].

The new generation enables three primary use cases [2], [3], eMBB (enhanced Mobile Broadband), which provides high data rates, mMTC (massive Machine Type Communication) that supports up to 10^6 devices per square kilometer, and URLLC (Ultra-Reliable Low-Latency

Communication), which aims to provide 1 ms latency for latency-critical communications such as V2X (Vehicle to Everything) [4]. Moreover, other applications requiring wireless access networks with low latency and high bandwidth, such as disastrous or remote healthcare ones, are also being a stakeholder for coming 5G technology. These three use cases' final goal is to come up with flexibility for networks and to connect everything, everywhere, anytime [5].

One of the significant upsides of using a mobile system is the eMBB feature, which provides connectivity and higher bandwidth for users and can cover a range of services such as hotspots and wide-area coverage. In the first one, a high data rate, large user density, and high capacity are the essential characteristics. While, in the second one, being connected in a seamless way and mobility are essential. The features of eMBB make it be categorized as human-centric communication [4].

URLLC aims to provide reliable communications with latencies close to zero. With the emergence of technologies such as autonomous driving, the necessity for reliable and low-latency services has become crucial. As a result, URLLC came into reality to fulfill the requirements. It has an essential role in covering both human-centric and machine-centric communications. In the latter one, latency, reliability, and high availability are critical in establishing connections, primarily in latency-critical communications such as V2Vs (Vehicle to Vehicles), which are categorized under machine-centric communication. For the human-centric, low-latency and higher data rates can be needed simultaneously in cases such as 3D gaming and video surveillance [2].

When there are many machine-centric devices with the need for transmitting a small amount of data, mMTC can be beneficial. Having a battery life of up to ten years, a large number of devices, a low transmission rate, and not being delay-sensitive are the principal characteristics of this use case. When IoT (Internet of Thing) solutions based on NB-IoT (NarrowBand-IoT) [6] are deployed in places such as underground or inside other devices such as cars or dynamic traffic lights, being able to penetrate materials is critical. These features can be provided by the mMTC use case [4], [7].

5G features will allow having new and robust capabilities compared to past generations. A higher peak data rate of up to 20 Gbps for DL (DownLink) and 10 Gbps for UL (UpLink) are

excellent advancements that emerged with 5G accompanying. These numbers are theoretical data rates and can be achieved in ideal conditions. However, the user-experienced data rate, which is one of the critical KPIs (Key Performance Indicators) in 5G, is 100 Mbps for DL and 50 Mbps for UL. The main difference between the user-experienced data rate and peak data rate is that the former one can be achieved in real-time for the majority of the UEs (User Equipment).

For attaining a higher data rate, high spectral efficiency is needed. Spectral efficiency refers to the achievable data rate over a specific bandwidth, and for 5G networks, it will be three times of IMT-Advanced Standard (International Mobile Telecommunications-Advanced Standard), so 30 bit/s/Hz in the downlink and 15 bit/s/Hz in the uplink are expected. We should consider that by increasing the frequency, the spectral efficiency will decline.

Latency is another crucial KPI of 5G and will be significantly improved compared to the previous generations. For the control plane latency, which is the time of transition from the idle state to the active one, is 10 ms, and for the user plane latency, it is 4 ms for eMBB and 1ms for URLLC.

One of the main goals of 5G is providing seamless connections for mobile UEs. Mobility interruption, which is the time that a device cannot have coverage of a gNB (gNodeB), i.e., the base station for 5G, for transmitting its data, can play an essential role in such a case. As a result, for having seamless communications, it aims to be zero in 5G.

From the aspect of mMTC, battery life is one of the most critical KPIs, and the predicted target for it in the coming generation is beyond ten years. Besides these KPIs, a 5G network needs to be reliable, supports up to 500 km/h mobility for a device, and 10^6 devices in a square kilometer. Moreover, consuming up to 100 times less energy compared to LTE and having area traffic capacity up to 10 Mbit/s/m^2 are other improvements [4], [7], [8], [9].

In the beginning steps, 5G NR (New Radio) established connections through the LTE core network called EPC (Evolved Packet Core), which was defined in early Rel-15 drop and called the NSA (Non-Stand Alone) mode. Then, the following specifications completed the standalone mode, which made it possible to have a fully connected end-to-end 5G network and was initially defined in regular Rel-15. Detailed information about the standalone mode and the frequencies

can be found in [10], [11]. Deploying gNBs and 5GCN (5G Core Network) together creates a fully 5G end-to-end communication called SA (Stand-Alone) [12]. In the former implementation, a connection between a UE and a gNB is established utilizing 5G RAN (Radio Access Network) while using EPC. However, in the latter one, the deployed RAN, and the core network are entirely 5G ones. The inceptive step for introducing 5G SA is by deploying low-band frequencies. Nowadays, using third-generation chipsets in devices helps gain optimized performance [1].

The deployment of 5G and implementation of the infrastructure have speeded up recently. Since 2018, when the first 5G device was launched, the deployment of 5G has been accelerating, and as time passes, the transformation from the old generations to the new ones gets more interest. For example, in 2019, GSM/EDGE had a large portion of the India region. However, in 2026, LTE and 5G are predicted to have 66 and 26 percent mobile communication in this region, respectively [1].

Having a detailed look on different regions proves that the penetration of new technologies is speeding up, and more nations intend to exploit cutting-edge ones in their mobile communication. As an example, in the Middle East and North Africa, LTE had 32 percent of mobile communication by the end of 2020. However, in 2026 LTE and 5G are expected to have 51 and 18 percent of the market, respectively. Until the end of 2021, around 580 million users will be able to connect to 5G networks, and this number is predicted to be 3.5 billion in 2026. Table I shows the penetrations of LTE and 5G in different regions [1].

Because of the new features and capabilities of 5G, new networking terms have been introduced. Network slicing is one of the most significant terms that has been included in 5G and opened a new horizon in mobile communication. The importance of network slicing becomes obvious when realized that various use cases of 5G require different resources, and the capacity needed by the end-users must be delivered efficiently based on the requisites. For example, eMBB demands high bandwidth, mMTC needs ultra-dense connectivity, and for meeting URLLC necessities, providing low latency is paramount [8], [13], [14]. These unique features make 5G capable of delivering new services to Industrial IoTs, TSCs (Time-Sensitive Communications), NPNs (Non-Public Networks) [5], reliable communication between vehicles [15], novel services to industrial stakeholders (i.e., vertical industries) [16], location-based

services [17], and NB-IoTs [18]. Combining these features satisfies the requirements to build a low latency [19], high speed, and fully connected world, one of the 5G era aspirations.

TABLE I

THE PENETRATION OF LTE AND 5G IN DIFFERENT REGIONS

Region	LTE by the end of 2020	Expected LTE by the end of 2026	Expected 5G by the end of 2026
The Middle East and North Africa	32%	51%	18%
Sub-Saharan Africa	15%	28%	7%
India	61%	66%	26%
Southeast Asia and Oceania	42%	57%	33%
Central and Eastern Europe	50%	65%	33%
Latin America	59%	48%	34%
North-East Asia	83%	33%	65%
Western Europe	78%	27%	69%
North America	89%	16%	84%

For attaining high bandwidth and meeting 5G requirements, radio frequencies that have been used in the previous generations, such as LTE and 3G, seem obsolete to be exploited in the new generation or need to be refarmed; as a result, there is a need for using more suitable spectrum for 5G networks.

Between 300 MHz and 3 GHz are radio frequencies, from 3GHz to 30 GHz are microwave bands, and from 30 GHz to 300 GHz are named mmWave. Each frequency has a distinct characteristic behavior that separates it from the other ones.

Mobile telecommunication was using bands up to 2 GHz until 3G. However, by the expansion of telecommunication technologies and the advent of 4G networks, higher frequencies up to 6 GHz were employed because the lower ones were not able to fulfill the new demands. With the recent advances in mobile devices, using mmWave frequencies is becoming available too, and

the IMT 2020, i.e., 5G, has the capability of deploying higher frequencies, including mmWave bands. The main characteristics of each frequency band are explained as follows.

Low-frequency bands, which are below 2 GHz, can cover a wide area and penetrate deep locations. When a device is located in a hard-to-reach place, these frequencies can be convenient. Due to this feature, NB-IoTs use low frequencies to take advantage of them in penetrating materials [18]. The drawback of these frequencies is that they cannot provide a broad spectrum, so the channel bandwidth needs to be set at most to 20 MHz.

Medium-frequency bands that span from 3 GHz to 6 GHz provide high data rates, wider channel bandwidth, and extensive coverage. These frequencies can support wider channels up to 100 MHz, five times larger than low-frequencies. This number can even be extended to higher frequencies by deploying carrier aggregation.

Higher-frequency bands are larger than 24 GHz and incorporate mmWave. They are suitable for high capacity and data rates, especially in hotspot coverage. Considering the wide bands, channels up to 400 MHz can be supported in high-frequency bands. This number can even be extended to 6.4 GHz by aggregating sixteen channels. Besides its advantages, higher frequencies (i.e., mmWave) suffer from some drawbacks. The most important ones are: 1) they cannot cover wide areas, 2) they are unable to penetrate materials, and 3) they are absorbed by rain. The critical difference between mmWave and other frequency bands is the wide range of frequencies, which for implementing it, both devices and base stations need novel technologies compared to the previous generations [4]. This thesis will make special attention to mmWave due to its grand role in 5G networks.

The existing downsides in 5G mmWave can lead to some issues, such as blockages that can affect these networks' performance. It means that obstacles such as buildings, cars, and human bodies can block the channel, which is in charge of data transmission, and degrade the network's performance [20]. Although some solutions, such as beamforming and handover, can mitigate the adverse effects to some levels, they cannot compensate for the signal quality reduction [21].

These problems can be more intense when requiring a reliable end-to-end connection over 5G. The reason is that the end-to-end reliable transport layer's widely used protocol, TCP

(Transmission Control Protocol), has a critical role in the performance of end-to-end connections, so there will be a necessity of making it compatible with 5G networks [22].

In order to gain high performance in 5G networks, the first important issue is the blockage problem, which can degrade the strength of mmWave signals by interrupting the communication and affecting the TCP congestion control mechanism due to the reaction of TCP to packet losses [23]. TCP is unable to perform appropriately when frequent interruptions occur in the network because it cannot distinguish a packet loss is due to congestion or other shortcomings of the 5G network, such as blockages, misalignments, and even random packet losses [20], [22], [24]. Therefore, to improve end-to-end performance and have stable connections, problems such as blockage need to be addressed; if not, these adverse effects can decline cellular networks' performance and prevent them from fulfilling the 5G requirements. Due to the characteristics of the high-frequency bands, this problem is more highlighted in mmWave.

1.1 Thesis objectives

In order to contribute to the blockage problem in 5G mmWave networks, the following main objectives have been identified:

General Objectives:

- The identification of advanced techniques for achieving high-speed reliable end-to-end communications over 5G networks.
- A contribution to reliable end-to-end communications over 5G networks based on advanced techniques.

Specific Objectives:

- The analysis of the effects of deploying TCP in 5G mmWave networks.
- The discussion of TCP mechanisms and parameters involved in the performance of 5G networks.
- Analyzing the impact of edge and remote server deployments.
- State of the art study, a survey of current challenges, solutions, and proposals.
- Analyzing TCP performance in-depth in 5G mmWave in the urban deployment scenario.

- A feasibility analysis proposal of Fuzzy logic and machine learning-based approaches to apply to improve reliable end-to-end communications in 5G networks.
- Implementation of a learning-based approach to improve high-speed, reliable end-to-end communications in 5G networks

1.2 Thesis outline

Considering the mentioned motivations, the rest of the thesis is as follows:

- Chapter two gives a comprehensive analysis of TCP, TCP over 5G mmWave networks, and the background of the accomplished researches.

- Chapter three designs and analyzes a novel protocol based on the fuzzy logic and presents the results for the new protocol.

- Chapter four is for designing and analyzing another novel TCP based on deep learning and its superiorities.

- Finally, chapter five concludes the thesis and talks about possible future work.

2 STATE OF THE ART

This chapter presents an analysis of TCP (as the most used protocol for reliable end-to-end communications), 5G mmWave networks, TCP compatibility with 5G mmWave networks, different procedures and parameters for them, a thorough analysis of TCP's functionality in urban deployments, and related work.

2.1 Fundamentals of TCP and TCP variants

TCP [25] is the most widely used protocol for reliable end-to-end communications in the transport layer of the TCP/IP protocol stack. Apart from end-to-end reliability, TCP has a congestion control mechanism to handle the unacknowledged packets (i.e., packets in-flight) in order to utilize the available bandwidth and retransmit the lost ones. This mechanism is mainly controlled by a so-called congestion window (cwnd), which is used to adjust the sending rate.

The TCP's CC (Congestion Control) mechanism incorporates four phases: slow start, congestion avoidance, fast retransmit, and fast recovery. In the slow start, the congestion window size is increased by one segment per received ACK, doubling in every RTT (Round Trip Time). This process will continue until the cwnd size is larger than a defined threshold (ssthresh) [26], a packet loss occurs in the network, or the window size exceeds the maximum transmission window announced by the receiver. If a packet loss occurs during the slow start, ssthresh will be set to half of the current cwnd, and if this packet loss is due to time out, cwnd size will be set to one.

When ssthresh is reached, TCP initiates the CA (Congestion Avoidance) phase. This phase can be different based on the deployed TCP variant because each one has a unique mechanism.

Generally, CA is a way to tackle the problem of packet loss events in the network by adjusting the sending rate. Two signals can commonly indicate a packet loss, occurring a time out and receiving three duplicate ACKs. This phase's goal is to slow down the sending rate when a packet loss occurs and accommodate the sending rate and the network's congestion status. The ultimate goal could be functioning at the BDP (Bandwidth-Delay Product).

When a segment is received out of order, duplicate AKCs are created. Plus, a lost segment, a reordering process, can also be the source of a single duplicate ACK. As a result, TCP waits for at least three duplicate ACKs to ensure that a packet loss has occurred. In this case, without waiting for the time-out to be triggered, TCP resends the lost packet. This phase is called fast retransmit because it speeds up the retransmission process in the network. When the fast retransmit is finished, TCP enters the congestion avoidance phase, not the slow start; this process is called fast recovery. The goal of fast recovery is attaining high throughput during moderate congestion in a network.

When a sender receives no ACKs for a particular amount of time, RTO (Retransmission Time-Out) is triggered, and TCP initiates the retransmission mechanism.

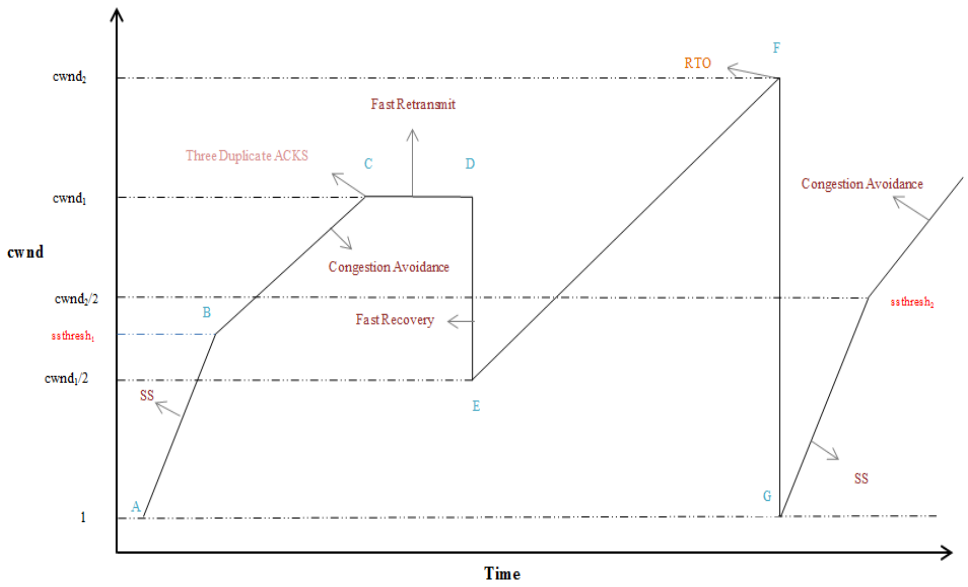

Figure 1. How TCP works

This process aims to ensure that sent packets are delivered to their destinations and prevent TCP from working with high sending rates when the network is considered heavily congested.

Although the value of RTO can be different based on the used approach, the recommended value by RFC 6298 [27] is a minimum of one second, however, Linux uses a 200 ms one. A discussion on the value of RTO will be included in section 2.3.3. After RTO detection, TCP sets the cwnd value to one and enters the slow start phase. Figure 1 shows how TCP performs.

This figure shows all of the conventional TCP phases and how they react to different situations in a network. In this figure, SS indicates the slow start phase, ssthresh stands for the slow-start threshold, and RTO represents retransmission time-out. From A-B, TCP is in the slow start phase, and after exceeding ssthresh in B, TCP enters the CA phase until it is interrupted in C by detecting three duplicate ACKs. From C to D, TCP is performing in the fast retransmit phase to be sure that all of the lost packets are retransmitted. Then in E, through fast recovery, TCP starts CA over until F, where an RTO indicates congestion in the network and forces TCP to enter the slow start again.

One of the most critical questions is when a network is heavily congested and how it can mitigate this situation. Because in a congested network, fairness and throughput, which are significantly important in the TCP implementation, are degraded dramatically. The immediate solution is using the backoff mechanism, which was first introduced in [28]. In TCP, backoff strives to set RTO values more efficiently and reschedule them after packet loss events. Because the value of RTO when the network is congested is vital and the essence for adjusting it accurately is crucial. This can be highly important when several retransmissions happen in a network and using a static value can exhaust the network [25], [26], [29], especially in 5G networks when various problems can cause segment retransmission. If the value for RTO is set to a small number, short interruptions can initiate it based on the buffers' size. As a result, blockages caused by small obstacles will have the capability of triggering the RTO if it is set to a small value. If it is set to a relatively high value in order to avoid RTO triggering by small and normal obstacles, the probability of congestion occurrence in the network will increase, and controlling the bottleneck will be almost impossible. As a consequence, exhaustion of the network's bandwidth will be inevitable. Adjusting RTO in 5G networks is highly paramount and needs to be done accurately.

Today there are different types of TCPs, loss-based [30], [31], [32], [33], [34],[35], [36] such as NewReno, HighSpeed, and Cubic, delay-based [37], [38], [39], [40], [41], [42] like Vegas, and loss-based with bandwidth estimation (i.e., hybrid) [43], [44], [45], [46] such as Westwood

and Jersey. Based on the popularity and implementation of the current networks, we have decided to focus on four of the TCPs, NewReno, CUBIC, BBR (Bottleneck Bandwidth and Round-trip), and HighSpeed.

From the network type's point of view, TCP can be divided into five categories [23]. The first group, which is the basic specification for the other TCPs, was striving to deal with the congestion collapse problem in networks. Congestion collapse is exceeding the sending rate above the network capacity, leading to packet losses after that rate. The protocols that belong to this group could somewhat solve the congestion problems. However, they created a new issue that is the underutilization of the network resources. The base protocol in this group is TCP Tahoe, which is a loss-based one, and others tried to improve its functionality by making some modifications or establishing new concepts such as delay-based TCPs as in TCP Vegas [41], or Vegas+ [47], which is an extension of Vegas.

With the emergence of new networks such as MANETs (Mobile Ad hoc NETworks), packet reordering was another issue that TCP needed to deal with, which lead to the creation of the second TCP group. These TCPs aim to distinguish loss packets and reordered packets. Because the previous protocols behave with both of them as lost ones; as a result, a reordering process can cause a congestion window reduction without considering the fact that no packet has been lost [23]. TD-FR (Time Delayed Fast Recovery) [48], Eifel [49], [50], and DOOR (Detection of Out-of-Order and Response) [51] are categorized in this group.

The third group's focus is on the different services that can exist in the network. TCPs that belong to this group try to give different priorities to various services. For example, if a background service such as an automatic update tries to initiate a connection, it gets less priority in comparison to foreground services. To be more precise, these protocols make an unfair network to give the network resource to services with higher priorities [23]. TCP Nice [52], which is based on Vegas, can be mentioned as an example in this group.

With the advent of wireless networks and appearing random packet losses due to the channel fluctuations and interferences between the different frequencies, the necessity for establishing new TCPs to handle this issue was inevitable because TCP assumes all packet losses as congestion indicators. When a packet is lost due to wireless channel characteristics, it is not a

sign of buffer overflow in the network, so reducing the sending rate in order to drain the buffers is not a sensible action [23]. This group's protocols are based on TCP Westwood [53] and Westwood+ [54].

By appearing faster networks and networks with long delays, the fifth group of TCPs appeared. This group aims to deal with the BDP problem, which happens for the legacy TCPs working in networks with high bandwidth such as optical networks or extensive delays such as satellite ones. The protocols that belong to this group try to use the network resources efficiently in a fair way and are able to react to the network changes [23]. High-Speed TCP [30] was the first protocol proposed, and the others tried to improve its functionality.

2.1.1 TCP NewReno

NewReno [55] follows an AIMD (Additive Increase Multiplicative Decrease) approach and is an extension to TCP Reno with a slight modification in its fast recovery phase [23]. In the CA phase, it increases the cwnd size by 1/cwnd for each received ACK. Comparing the slow start and CA reveals that NewReno increases cwnd by one MSS (Maximum Segment Size) for each received ACK during the slow start phase. However, during CA, cwnd is increased by one MSS for every RTT. Therefore, for increasing the cwnd size by one during CA, the entire cwnd should be acknowledged during a RTT. By perceiving three duplicate ACKs, NewReno reduces the cwnd by half and enters the fast retransmit phase.

On the other hand, when the RTO triggers, NewReno assumes the loss is due to congestion in the network. As a result, it sets the cwnd size to one and starts the slow start phase. By considering the CC mechanism in NewReno, it is evident that it is a loss-based TCP.

2.1.2 TCP CUBIC

This variant is the default TCP [34] in Linux since Kernel 2.6.26, Android, and MAC operating systems [24], [56]. The mechanism that the CUBIC approaches the congestion problem is based on a cubic function, and there are two different ways of increasing or decreasing the size of the congestion window. The first one is a concave portion when the cwnd size ramps up quickly to the size before the last congestion event. Next is the convex mode, where CUBIC probes for more bandwidth slowly at first, then continues rapidly. CUBIC considers the time right before the last drop and tries to reach that capacity in fast paces during a short time. The

aim is to reach the size that the cwnd had just right before the last drop. It means that the cwnd size adjusting process is independent of RTT, but rather on time between two consecutive drops (i.e., congestion). Being independent of RTT helps CUBIC to be more fair to different flows. When CUBIC is far from the value of the previously saturated cwnd, it adds up aggressively. However, when it is close to it, it increases it slowly, like in a NewReno mode. This can be achieved by using equation (1):

$$\text{cwnd} = C\left(\Delta - \sqrt[3]{\beta . max\text{cwnd}/C}\right)^3 + max\text{cwnd} \quad (1)$$

Where C is a constant, fast Recovery's multiplicative decrease coefficient is β, maxcwnd is the size of the congestion windows before the last packet drop incident, and Δ is the elapsed time from the last packet drop.

The main goal of CUBIC is to optimize the congestion control mechanism for high bandwidth networks with high latency as called LFN (Long Fat Networks). Furthermore, CUBIC improves fairness in the network. Because in conventional TCPs, flows with short RTTs receive their ACKs faster than ones with longer ones, so they can rapidly add up their congestion windows. CUBIC is a loss-based TCP, too.

2.1.3 TCP BBR

TCP BBR is a cutting-edge congestion control algorithm developed by Google in July 2017 [57] and is being used on Google, YouTube, and GCP (Google Cloud Platform). BBR, as the name indicates, tries to keep the most excellent cwnd size based on the current bottleneck bandwidth and RTT and tries to achieve high bandwidth with low latency. In conventional TCPs, a loss event is a sign of congestion in the network. In contrast, BBR ignores loss events and strives to estimate available bandwidth and minimal RTT in some predetermined periods. The ultimate goal of BBR is to deliver the highest available throughput along with minimal congestion by using these estimations. The approach that BBR deploys to attain its objective is being sure that the bottleneck is saturated without being congested; a bottleneck is the part of a network with the lowest bandwidth. So if BBR can maintain this part saturated, it can reach the highest available performance.

BBR has four different phases, including Startup, Drain, Probe Bandwidth, and Probe RTT. The first phase is an adoption of the conventional slow start TCPs, and during it, the sending rate is doubled every RTT. When the estimated bandwidth does not increase for the last three RTTs, BBR presumes that the bottleneck bandwidth is saturated but not congested. In such a case, BBR tries to drain the sending rate by slowing it down. This phase's primary goal is to empty the buffers that may have been filled during the start-up phase.

After finishing the Drain phase, Probe Bandwidth is initiated. This phase includes eight cycles, with the duration of each one equals the round-trip propagation delay, where BBR tries to probe for higher available bandwidth. In this process, BRR employs a variable called pacing gain to adjust the amount of sent data.

If the RTT has not been decreased for the last ten seconds, BBR starts Probe RTT. During this phase, in-flight packets are reduced heavily so the protocol can drain the network's buffers and estimate an accurate value for RTT. The period of this phase is the maximum of 200 ms or a RTT. By obtaining a minimum value, RTT propagation will be adjusted to the new one and deployed for the next ten seconds [58].

BBR is a model-based TCP and strives to deliver higher throughput and lower latency. One of the most critical problems BBR has is when it coexists with other TCPs, in a case that fairness can be a severe problem because BBR flows can be the dominant ones [22], [24].

2.1.4 HighSpeed TCP

This TCP variant [30] is suitable for networks with high bandwidth-delay products and has been designed for networks in which a fast growth of cwnd is essential. The reason for developing this protocol is that conventional TCPs perform deficiently in networks with large bandwidth-delay products, and it takes a long time for them to utilize the available bandwidth, especially in the congestion avoidance phase. This protocol makes some slight changes to the congestion control mechanism of the standard TCP to overcome this problem.

In the congestion avoidance phase, when an ACK is received, cwnd is increased by $(a / cwnd)$, which means cwnd = $(cwnd + a / cwnd)$, and when a packet loss is detected through triple duplicate ACKS, the cwnd size equals $((1-b) * cwnd)$, which means $cwnd = ((1 – b) * cwnd)$.

The value of a and b depends on the network's status, and HighSpeed and NewReno are similar when cwnd is small. As a result, *a* equals one, and *(1-b)* equals 0.5. However, when cwnd exceeds a defined threshold, unlike NewReno, HighSpeed tries to keep a high value for the congestion windows. In this case, as the size of the congestion window increases, the value of *a* is increased so that the protocol can send more packets, and the value of *(1-b)* is decreased; as a result, it can recover faster than NewReno in high sending rates when it detects a packet loss by a triple duplicate ACK. This process makes HighSpeed-TCP (HSTCP) add up the cwnd faster than NewReno and recover more rapidly.

When HSTCP is implemented in a network, the mentioned parameters are selected from a lookup table. HighSpeed is a loss-based TCP, too [22], [30]. A thorough analysis of TCP over 5G networks can be found in [59].

2.2 5G mmWave networks procedures and parameters for reliable end-to-end communication

5G network parameters and their characteristics have a critical effect on the delivered performance to end-users. Procedures such as handover, techniques such as beamforming, parameters such as RLC (Radio Link Control) buffer size, system architecture, ultra-lean design, and how they are implemented can play essential roles in 5G networks. This section aims to investigate these parameters and procedures and their impacts on the behavior of 5G networks.

2.2.1 Simulation parameters

To achieve a better understanding, in this section we present some results drawn from numerous simulations we have done. Therefore, first, we introduce the simulation scenarios and parameters. In all scenarios, a UE connects to a gNB at the height of 15 meters, which works at 28 GHz with a 1 GHz channel bandwidth. This antenna is connecting to a server operating at a 1 Gbps sending rate. The simulation parameters can be seen in Table II. It is worth mentioning that there were five trees and a building in our simulations for miming small and big obstacles behaviors. Moreover, the UE stopped for a couple of seconds behind the second tree and the building, as shown by the gray and cyan areas, respectively. The last part

of the figure, in which the SINR (Figure 2) value is reducing gradually, is because of the increased distance between the gNB and the UE, which is inside a car during this period.

Moreover, we have used four different BERs (Bit Error Rates) to emulate situations with small (1.25e-10), moderate (1.25e-9), large (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 [61], [62]. Due to these reasons having this type of losses in the network is indispensable. Furthermore, selecting 2.5 MB of the RCL satisfies 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, so they 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. The comprehensive analysis of TCP over the urban deployment can be found in [62], [63].

TABLE II

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

2.2.2 Blockage and misalignment

Apart from high packet loss probability, there are some issues such as blockage and misalignment in 5G networks. Blockage means that high frequencies, i.e., mmWave, cannot

pass through obstacles properly, and misalignment happens due to non-matching beams of transmitters and receivers. The existence of these problems can degrade the performance of 5G mmWave networks. In particular, they dramatically impact TCP's performance, which is responsible for establishing end-to-end connections. These problems make a transition from a LoS (Line-of-Sight) connection to a NLoS (Non-Line-of-Sight) one. In LoS, the data transmission can be performed through the established connection between the user and base station, but in NLoS, the channel's reduced bandwidth can harm the network's performance.

Figure 2 shows the SINR (Signal-to-Interference-plus-Noise Ratio) value taken out from our conducted simulations, which analyzes TCP thoroughly in urban deployments. The figure indicates that the signal strength is reduced when there is an obstacle between the UE and the gNB, which is the principal cause of throughput degradation. As a result, a proper connection can not be established, which misleads the conventional TCPs in utilizing the 5G mmWave networks' available capacity.

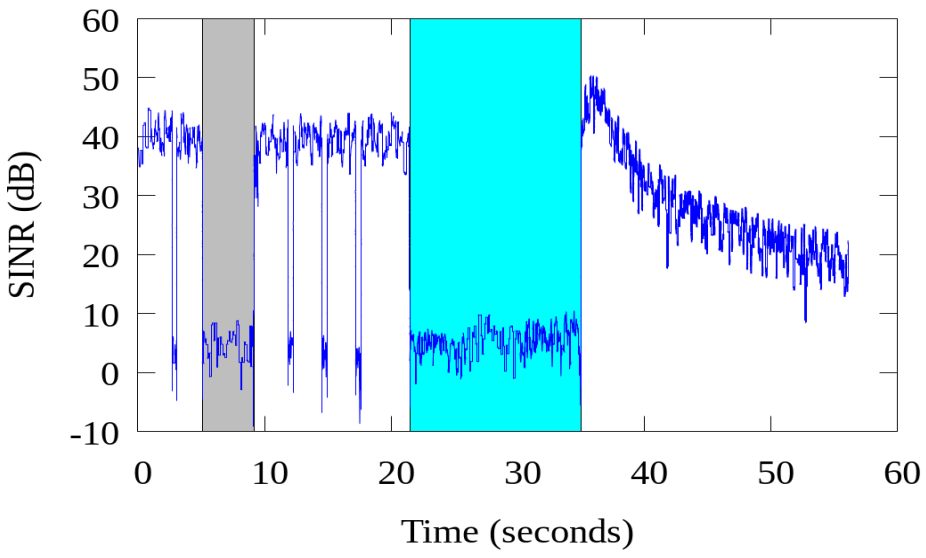

Figure 2. SINR fluctuation

2.2.3 Handover and beamforming

Handover or handoff means the process of changing an ongoing data session from one cell to another. Beamforming focuses on sending powerful signals toward a particular device. The reasons for deploying this technique are to prevent signal attenuation and tackle bandwidth degradation in blockage situations. One of the main reasons for exploiting these techniques

over 5G networks is to compensate for the intermittent nature of mmWave signals. For example, when a UE exits from a cell's coverage and enters another cell's area of coverage, in order to prevent the connection termination, handover can be initialized to establish a connection with the new gNB. Moreover, when a cell's capacity is reached and a new device intends to connect to it, a handover process can be triggered to find another gNB that can serve the new UE.

One of the most essential sources for handover initialization in 5G networks is due to blockage occurrence, especially when deployed with other generations such as LTE. In this sense, when the channel quality of the mmWave is degraded by occurring a blockage, a handover may initiate and connect the UE to a LTE eNB (eNodeB). This can be generalized when any channel quality degradation happens in wireless communication. When a connection is blocked by an obstacle, handover can find another cell to keep the connection on. The mentioned situations, i.e., blockage, outage, and reaching the maximum capacity of a gNB, can affect the performance of TCP in keeping the packet drops low. In this case, handover and beamforming can relieve the effect of negative factors on the performance of TCP and improve its functionality by preventing numerous packet drops that may happen. If we could have a channel with an adequate amount of bandwidth with the help of these techniques, it could prevent the throughput reduction and RTT increment of TCP to some levels. However, it may not omit the adverse impacts in some situations.

There are two kinds of handovers, horizontal and vertical. Horizontal handover refers to base station changes, i.e., changing from a gNB to another to maintain the connectivity. However, when a wireless technology change occurs, for example, from 5G to 4G, it is called a vertical handover. Experiments in [60] exhibit that both handovers can degrade the network's performance, though this effect can be severe in vertical ones.

2.2.4 5G core network and architecture

5GCN is based on the EPC with three novel improvements: service-based architecture, network slicing support, and SDN (Software-Defined Networking)/NFV (Network Function Virtualization).

Being service-based architecture means that the concentration is on the provided services and functionalities by the core network. Network slicing is a new term introduced in 5G and means, instead of separating a network into different physical parts, it is divided into some logical parts based on the service demands and necessities. In such a case, different slices are run on the same physical infrastructure, but from the user's view, they seem separate. Control-plane/user-plane separation, based on SDN/NFV, is one of the new features supported by 5GCN in order to use different capacities within them. As an example, it is possible to use more capacity for the user-plane without affecting the control-plane.

5G Core network and architecture can be analyzed in detail in future research to see that they can help TCP to improve its functionality. The serviced-based architecture can be an enabler in exploiting different TCPs for different services based on their needs. For services with high data rate necessity, high-speed TCPs can be used, or for those with delay sensitivity, the appropriate TCPs can be deployed. Other schemes can include using proper TCPs in different slices to seek the optimal functionality for the network. Finally, by separating the control-plane and user-plane, the deployed TCPs for each one can be distinct, and the ideal one can be chosen.

Deploying 5GCN eases the path to SA 5G networks and makes it possible to use the new generation's full privilege. New Features, i.e., service-based architecture, network slicing, and SDN/NFV, can be enabled based on a service requirement, which leads to an improved end-to-end user experience [12].

2.2.5 A close look on the behavior of TCP in urban deployments

When NLoS connections exist, the network may have temporary disconnections, confusing TCP in adjusting its congestion window size. These interruptions can vary based on the size of obstacles and speed of UEs, which can cause short to long failures. Although both disconnections can affect the performance of TCP, the effects of the long ones are more potent due to the high probability of triggering the RTO, which leads to a congestion window initializing and slowing down the sending rate dramatically.

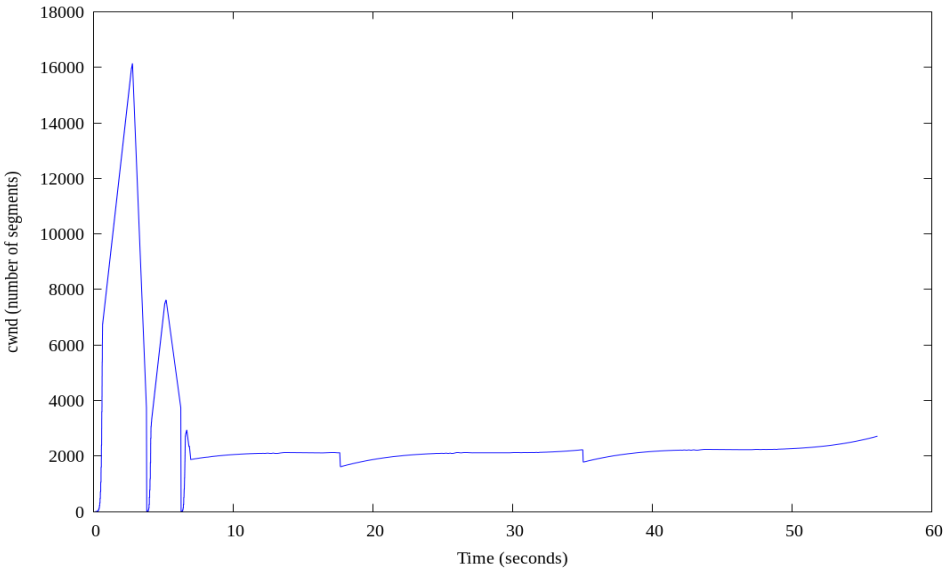

Figure 3. TCP CUBIC congestion window adjustment in an urban deployment, BER=0

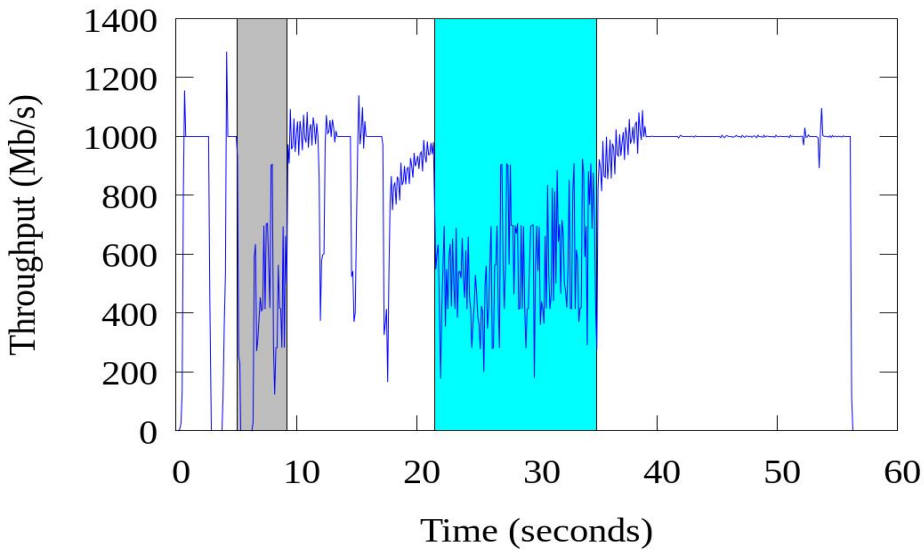

Figure 4. TCP CUBIC throughput in an urban deployment, BER=0

As an example, Figure 3 shows the congestion windows adjustment for CUBIC. As it is clear in the figure, there are two initializations during the first ten seconds because of the RTO triggerings caused by obstacles between the UE and the gNB.

This can be worse when the network is not congested because initializing the cwnd size in this situation can degrade the throughput profoundly. This confusion in TCP functionality leads to a drastic performance reduction of 5G mmWave networks, as shown in Figure 4.

This figure indicates that when there is an obstacle between a UE and a gNB, the achieved throughput can be degraded, and attaining the saturated value can be challenging. In addition to throughput, the RTT value can be affected negatively.

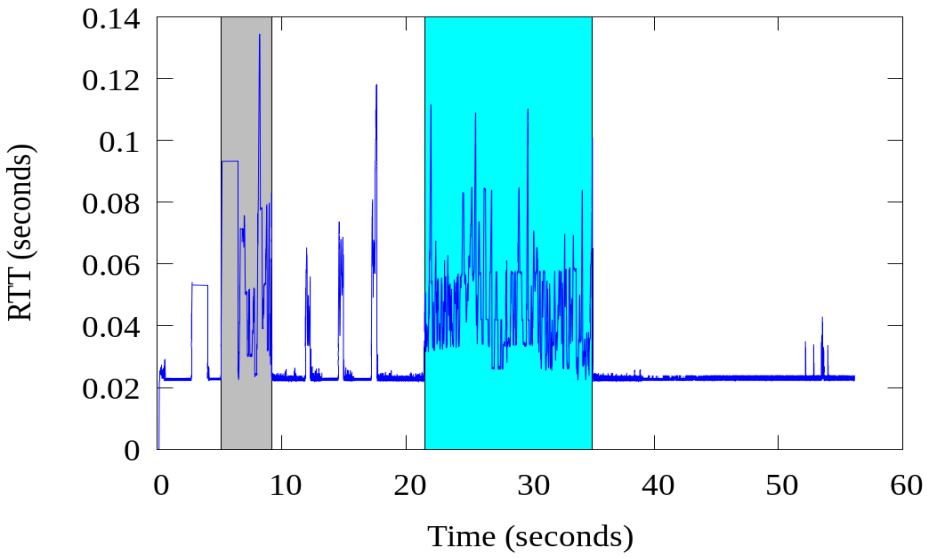

Figure 5. TCP CUBIC RTT in an urban deployment, BER=0

Figure 5 indicates that when there is an obstacle between a UE and a gNB, the enqueued packets in buffers can cause RTT increment.

There are some immediate solutions, such as installing several gNBs in order to broaden the LoS regions or putting some mirrors to act as relays that can get the signals in NLoS states and reflect them to LoS areas [20], in order to mitigate the adverse effects of NLoS conditions. However, they are expensive or can alleviate the problem to some levels, but are not good enough to eliminate it.

Blockage can happen because of different objects such as buildings, buses, cars, human bodies, and pillars. Almost anything except some thin materials like clear glass can create it. This can be emphasized when we know it is hard for 5G mmWave signals to penetrate a hand or a

human body and makes the problem more difficult in urban areas. Another issue that makes the problem more severe is using UV-protective windows, which is common these days because they can act as hurdles in the way of signals. These windows can attenuate 5G signals and reduce the quality of the received ones, which leads to performance reduction.

When there is a blockage in a static situation, the chance of passing the obstacle is low, and the negative effect will be tougher. As a result, in some cases, being dynamic can be beneficial by increasing the chance of reconnection between a UE and a gNB. The static mode can create persistent conditions and reduce the quality of the received signal by a UE intensely, as seen in the gray and some parts of the cyan areas of Figure 6, which shows TCP HighSpeed throughput in an urban deployment when BER=0. Throughout these periods, the UE stops behind the obstacles; as a result, compared to the other parts of the figure, the degradation during these periods is intense.

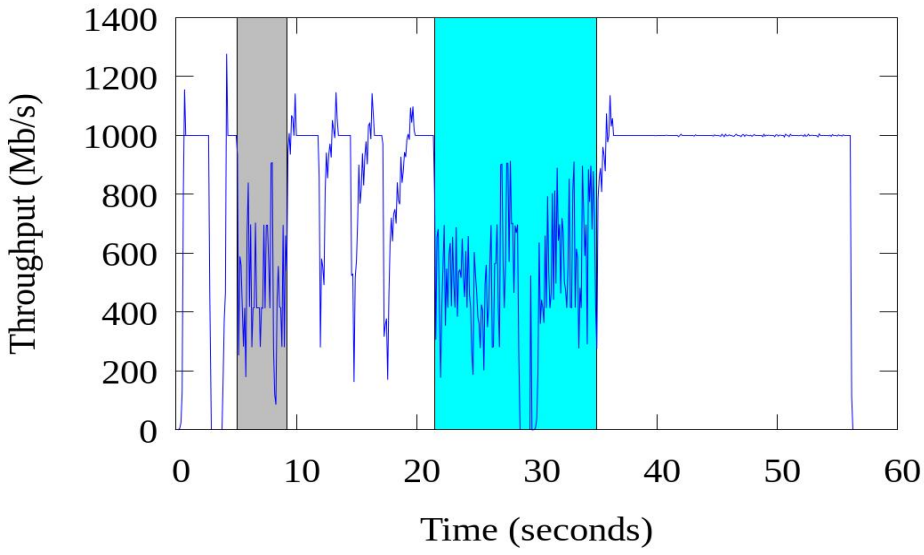

Figure 6. TCP HighSpeed throughput in an urban deployment, BER=0

Employing handover can reduce the negative effect of blockage by changing the associated gNB and keeping the connection on. As a result, a channel is blocked by an obstacle, it helps to have sustainable connections by finding a new base station [20], [22],[60] [62], [64].

To sum up, blockage creates longer RTTs, higher packet loss probability, and may trigger TCP RTO, which all of them degrade the performance of TCP over 5G mmWave networks.

In addition to blockage, other factors can harm the performance of 5G. Distance between a UE and gNB is one of these parameters that can play an essential role in the performance of the network. However, the negative impact that distance can have is low compared to blockage. In addition to the blockage and the distance between a UE and gNB, the orientation between them is another effective player in the performance of 5G networks. In this case, a 90 degree one is the worst case, and a zero-degree one is the most favorable [60].

Like blockage, misalignment can degrade the performance of 5G networks, especially by having adverse effects on the functionality of the transport layer. Misalignment is a severe issue in environments with high mobility compared to static conditions. This means a persistent connection cannot be established when a transmitter and receiver phases are not matched. During the initial connections, communication can perform well; however, UE mobility can cause a change in the angle between a UE and a base station, leading to mismatching pairs between the gNB and the UE.

Although some techniques, such as beam sweeping, try to match the pairs after misalignment occurrence, these techniques lose their efficiency when the UE keeps its mobility. If beam sweeping finds a matched pair between the UE and gNB, there are no guarantees that the communication will remain consistent because misalignment can happen frequently. The main effect of misalignments on TCP is that TCP cannot differentiate between packet losses due to misalignment or congestion and assumes all of them are because of congestion.

To conclude, misalignment and blockage's adverse effects on the performance of 5G networks are somewhat similar. Because in both cases, TCP cannot distinguish a packet loss is due to congestion or disconnections.

2.2.5.1 Blockage Effect on the Different Deployment Scenarios

The deployed scenario plays a vital role in the blockage's effect on the performance of reliable end-to-end communication on mmWave 5G networks. In situations that the blockage effect is low, TCP can work more efficiently. In contrast, circumstances with a high number of blockages will have some difficulties maintaining the high throughput and need more attention. As a result, knowing the effect of blockage on different deployment scenarios can give an insight and provide a clear vision of creating new protocols.

TABLE III

IMPACT OF THE BLOCKAGE ON DIFFERENT DEPLOYMENT SCENARIOS

Deployment Scenario	Carrier Frequency	Number of Obstacles	Blockage Effect
Indoor Hotspot	4, 30, and 70 GHz	High	High
Dense Urban	4 and 30 GHz	High	High
Rural	700 MHz and 4 GHz	Low	Low
Urban Macro	2, 4, and 30 GHz	High	Medium
High-Speed	4, 30, and 70 GHz	Low	Low
Extreme Long Distance Coverage in Low Density	3 GHz and below 1 GHz	Low	Low
Urban Coverage for Massive Connections	700 and 2100 MHz	High	Low
Highway	6 GHz	Low	Low
Urban Grid For Connected Cars	6 GHz	Medium	Low
Commercial Air to Ground	Below 4 GHz	Low	Low
Light Aircraft	Below 4 GHz	Low	Low
Satellite Extension to Terrestrial	1.5, 2, 20, 40, and 50 GHz based on the deployments scenario	Low	Low

Table III shows the impact of the blockage on different 3GPP deployment scenarios [65]. It compares the various scenarios and the level of blockage effect on each one, as Low, Medium, and High. Low means having a few obstacles, medium indicates a range of obstacles that can interrupt the communication but not in a continuous way, and high means having many obstacles that create frequent disconnections.

From the ITU-R (International Telecommunication Union- Radiocommunication) point of view, there are five different scenarios: indoor hotspot-eMBB, dense urban-eMBB, rural-eMBB, urban macro-mMTC, and urban macro-URLLC [66]. These five scenarios are considered test environments to evaluate the performance of IMT-2020, i.e., 5G networks [67].

The blockage effect can be intense in the indoor hotspot, which focuses on high throughput for users inside buildings and intends to use high carrier frequencies. The first reason is many obstacles that create interruptions in the connections and make the channels intermittent. In this situation, having consistent communication is almost impossible. The second reason is using high frequencies as we know that the signal can be attenuated easily when the frequency increases.

Dense urban deployment focuses on providing high throughput for many users in a downtown or dense areas inside a city. In this scenario, besides data rate, coverage is another factor. Besides high frequencies, there are a large number of obstacles like buildings, cars, buses, and humans in this deployment, which can have both positive and negative effects. As the number of obstacles grows, the probability of having an interruption because of a blockage increases. On the other hand, large obstructions like buildings in a high number can reflect the signals sent by gNBs. As a result, it can help to mitigate the negative effect of the blockage. In this scenario, the majority of the users are inside buildings or moving at a speed of 3 km/h, making it hard to have constant connections.

The rural deployment aims to cover large areas. In this scenario, the most crucial factor is supporting high mobile vehicles in broad areas. Because of using low frequencies around 700 MHz and 4 GHz, and areas with a few obstacles, the blockage cannot have a substantial effect in this scenario.

Like the rural deployment, the urban macro scenario focuses on the coverage of broad areas but inside a city. It uses both higher and lower frequencies based on the requirements. Most of the UEs are considered to be inside buildings, which makes it hard to reach them. Although blockage can have adverse effects in this scenario, it is less than indoor hotspot deployment because in the first one, all users are considered inside buildings, and 70 GHz frequency can be used, which is highly sensitive to obstacles. But in this scenario, some of the users are outside, and frequencies around 30 GHz or even lower can be deployed, which mitigate blockage effects.

The high-speed scenario strives to cover UEs inside high-speed trains. High mobility up to 500 km/h is the key characteristic of this scenario. To support all of the users, many small cells (i.e., gNBs) are deployed along tracks. By using handover techniques, the blockage effect can be

eliminated entirely because a UE is connected to a gNB all the time. Moreover, the number of obstacles in this scenario is low compared to the urban ones.

For large areas with few users, extreme long-distance coverage in low-density scenarios can be the primary candidate. In this scenario, macrocells with frequencies below 3 GHz are used to provide extensive coverage with moderate bandwidth because a high sending rate is not a priority. As a result, the blockage is not an essential issue in this scenario, especially when frequencies under 1 GHz are deployed.

For fulfilling mMTC requirements, the urban scenario for a massive number of connections can be exploited. The most important parameter here is the high number of devices, which can be indoor, outdoor, or inside cars. However, considering the penetrating power of low frequencies used in this scenario, it is almost immune to the blockage problem. Frequencies around 700 MHz or 2 GHz are favorite in this scenario, which can satisfy the requirements.

There is a scenario similar to high-speed but with lower mobility supporting up to 300 km/h, which is called highway, and focuses on supporting mobile vehicles on highways. Using a lot of small cells, not existing obstacles, open area of connection, and using frequencies around 6 GHz are the main characteristic of this scenario that mitigate the adverse effects of the blockage.

When freeways end in cities, they can cause heavy traffic with a large number of cars. For supporting this scenario, the urban grid for connected cars is the leading candidate. The aim of this deployment is to provide reliable and available connections with an acceptable latency for cars. Like high way scenario, this one is deploying macrocells with frequencies around 6 GHz, which makes it somewhat immune to blockages.

Both commercial air to ground and light aircraft scenarios are for supporting machines on the air. In the first case, the goal is providing connections for UEs boarded on airplanes, and in the second one for UEs boarded on helicopters and small airplanes. The main goal in both of them is supporting a large area of coverage upward. Throughput and user density are not KPIs here, and providing basic data and voice services is convenient by using frequencies below 4 GHz. Existing almost zero obstacles and exploiting lower frequencies omit the adverse effects of the blockage. However, it was not a severe problem from the beginning.

Satellite extension to terrestrial is the last deployment scenario. It is helpful in supporting those areas where providing terrestrial services is impossible or not noteworthy to be deployed. Because of using satellites for broadcasting, the blockage is not an issue in this scenario [65].

2.2.6 RLC buffer size and new references

RLC buffer size can have a crucial effect on the performance of 5G networks. Although attaining higher performance exploiting large buffers can be beneficial, it leads to higher latency values. There are two main reasons that large buffers yield higher performances. First, when big buffers are used, the chance of packet drops due to buffer overflow becomes low. However, deploying big buffers can make long queues and causes packets to wait longer in buffers, which leads to bufferbloating issue. Reliable transport layer protocols, such as loss-based TCPs, will be affected intensely by this drawback. Second, the network becomes less sensitive to high link variations of the 5G mmWave channel. The reason is that in NLoS states, RLC buffers can store a large number of packets and prevent them from being dropped. This process alleviates the sending rate reduction, which TCP may force to the network in every packet drop. Moreover, exploiting techniques such as HARQ (Hybrid Automatic Repeat reQuest) can mask some packet drops from the higher protocol stack layers, even so, this technique can lead to higher latencies.

In contrast, when small buffers are used, latency can be decreased at the cost of declined performance. Maintaining a tradeoff between performance and latency is critical, especially when remote servers are deployed and severely affect TCP.

To see the impact of the RLC buffer size on the performance of TCP over 5G mmWave networks, we can look at Figure 7 and Figure 8, which show the functionality of the network for unlimited and 100% BDP buffer size, extracted from our simulations of urban deployments.

The UDP (User Datagram Protocol) saturated values are shown by the red dashed lines in the figures. These figures show that TCP can benefit from a larger RLC buffer size deployment in terms of throughput.

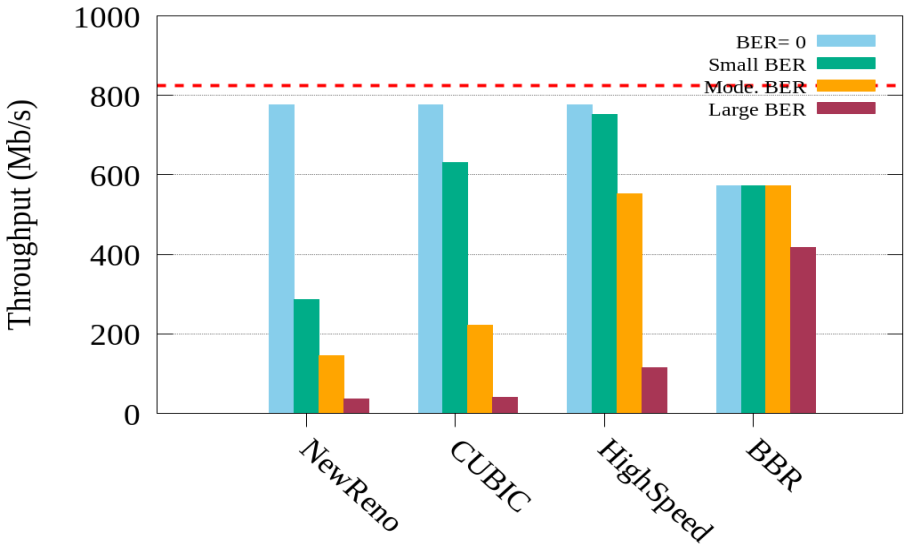

Figure 7. Average throughput for different TCPs, unlimited RLC buffer size

This tradeoff can be attained by using AQM (Active Queue Management) techniques, such as CoDel [68] and Fq-CoDel [69]. However, these techniques require some modifications to be adapted to 5G networks [22].

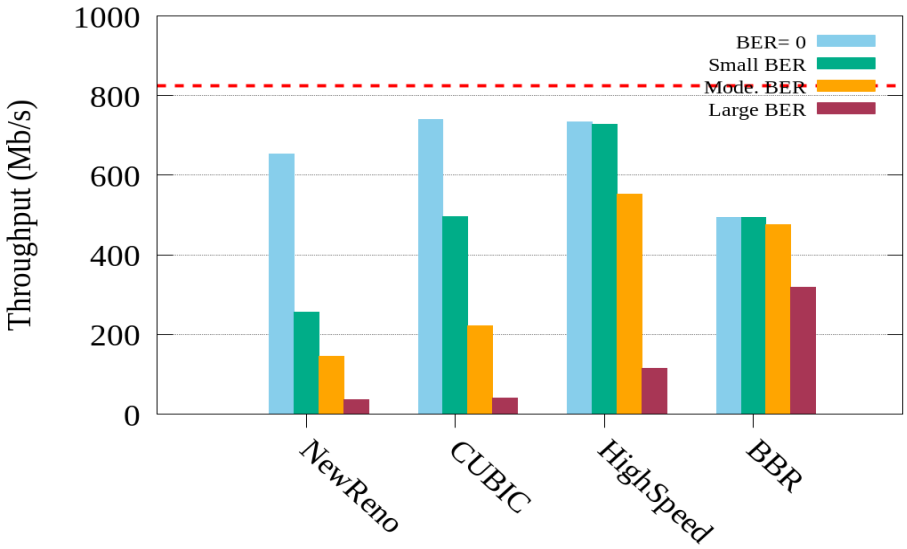

Figure 8. Average throughput for different TCPs, 100% BDP RLC buffer size

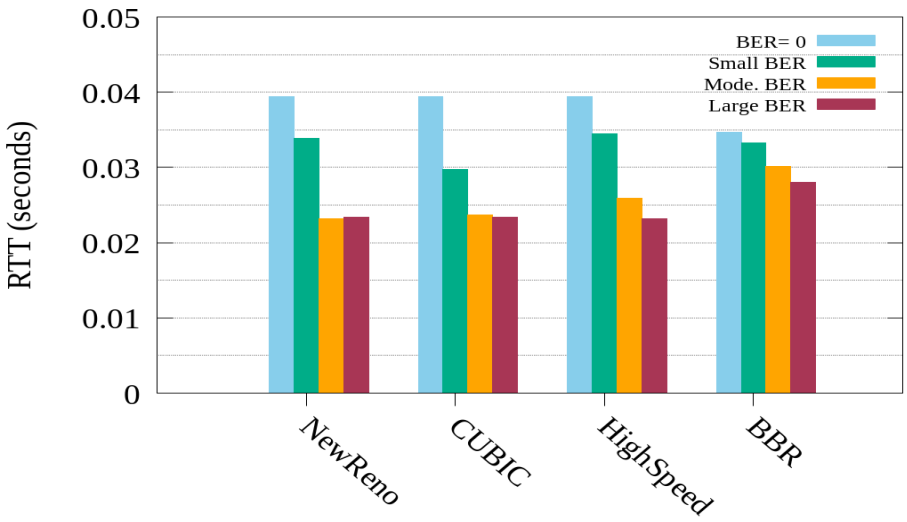

Figure 9. Average RTT for different TCPs, unlimited RLC buffer size

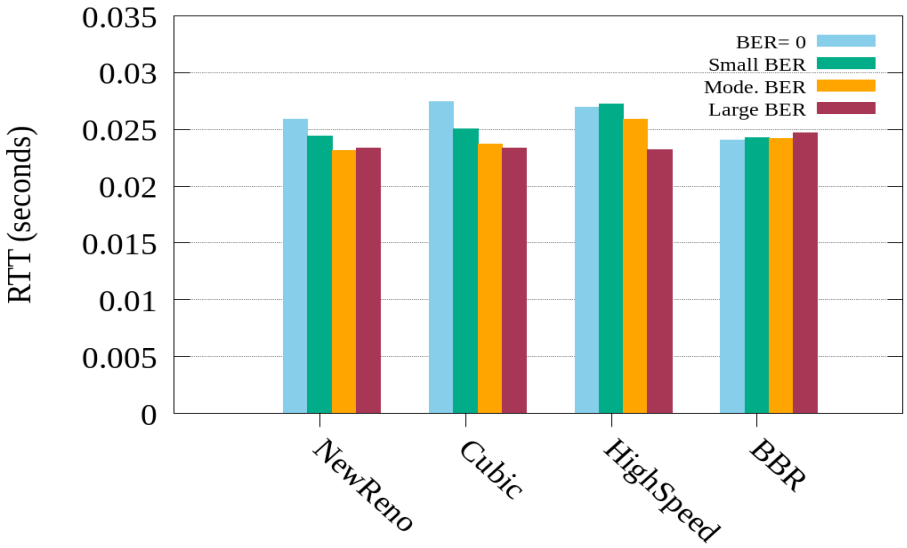

Figure 10. Average RTT for different TCPs, 100% BDP RLC buffer size

However, from the RTT point of view, TCPs are affected negatively when the RLC buffer size increases, as shown in Figure 9 and Figure 10. Comparing these values to the minimum RTT, which is around 20 ms shows that TCP cannot benefit from a large RLC buffer size deployment in terms of RTT. We should notice that in real-world scenarios, a trade-off between throughput and RTT needs to be kept considering the required use case.

2.2.7 Latency

One of the most stringent requirements for 5G networks is the value of latency. Latency is the time interval from when a source sends a packet to the time the destination receives it and will be improved significantly in 5G networks, especially for critical latency devices that exploit the URLLC use case of 5G. The value of latency can be damaged in 5G networks due to the existence of adverse impacts, and the occurrence of blockage and misalignment can create long latencies in the network [22] [56]. Therefore, this parameter must be a performance optimization too.

Generally, conventional TCPs can benefit from reduced delays in a network. One of the significant efforts in improving TCPs functionality is to bring the servers close to users by employing techniques such as CDN (Content Delivery Network) in order to reduce the delay, which can improve TCP's functionality. In such a case, servers and data centers are distributed in different locations. By exploiting CDN in bottlenecks, the availability of services, page load time, and other networking features will be enhanced. The reason for reducing the delay in a network is that most TCPs increase the congestion window by receiving an acknowledgment; as a result, reduced delays lead to shorter loops and faster reactions.

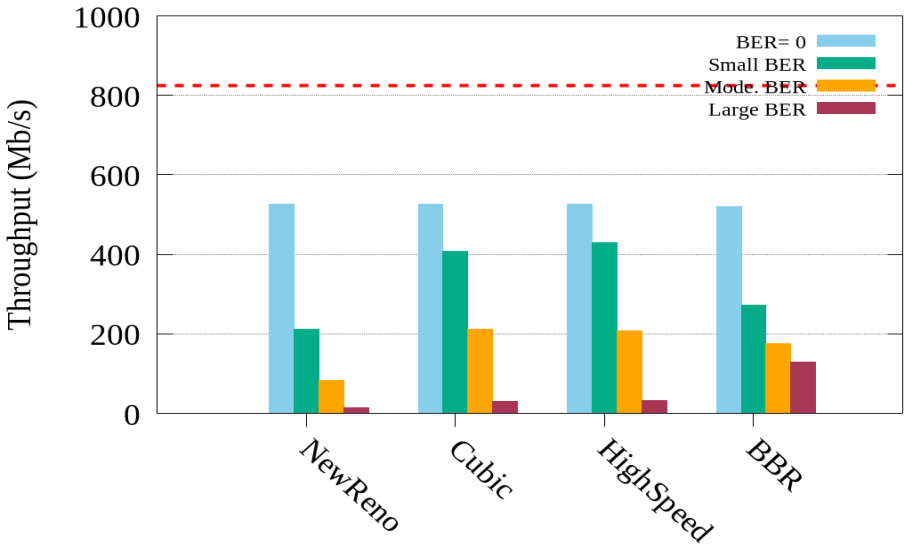

Figure 11. Average throughput for different TCPs during a remote server deployment

In this case, the reduced latency in 5G networks can help TCP ramp up to the high sending rates and have better functionality. In addition to faster reaction, by improving latency and having similar values, fairness between different flows can be enhanced [56].

The latency impact can be seen in Figure 11, which shows the remote server deployment (80 ms) effect on the throughput of different TCPs. Comparing this figure to Figure 8, which has been simulated under an edge server deployment, indicates that different TCPs can benefit from the edge server deployment and reduced latency.

2.2.8 Ultra-lean design

One of the most important problems that current mobile communication has is the existence of "always-on" signals, especially in highly dense areas with an extreme traffic load. The presence of these signals is regardless of user traffic and can occupy a portion of the bandwidth in the network. Signals such as base station detection, system information broadcast, and channel estimation reference are categorized under always-on signals. TCP's ultimate goal is to handle the congestion issue in networks and have fairness between users and flows. However, always-on signals can increase the traffic in a network and affect TCP performance. As a result, deploying ultra-lean design can help mitigate the amount of traffic in networks, reduce congestion events, and improve TCP functionality.

Moreover, energy consumption is another negative aspect of these signals, and they can create interferences too. The ultra-lean design in 5G networks strives to reduce the use of always-on signals. As a result, it can enhance energy consumption, prevent bandwidth wastage, and mitigate signal interferences in the network by turning on the signals when needed and turning them off when they are not.

Furthermore, reducing channel interference and bandwidth usage can enhance user experience as the transport layer protocols will encounter fewer sudden changes in the network [4].

2.2.9 Fairness

When several flows coexist with different RTTs, severe fairness issues can arise depending on the deployed TCP and the dropping strategy's aggressiveness. One of the principal achievements of TCP is reliable connections through fair networks. It means that when there

are several flows in a network, they get the same proportions. However, retaining fairness in 5G mmWave networks is a challenging issue. The reason is that this feature is directly connected to RTT values. If a flow has a shorter RTT, it can ramp up to higher throughput quickly and gets more shares of the available bandwidth, as a result, forcing unfairness to the network. This unfairness, which is due to increased RTTs caused by NLoS states, can be intense in scenarios with many hurdles, such as urban deployments. The reason is that when the number of LoS to NLoS transitions increases, it leads to increments in the RTT value. It can be more severe while a UE can see the gNB, and another one cannot establish a proper connection because of being behind an obstacle. This NLoS state will increase the value of the RTT for the corresponding flows; consequently, the user's share from the bandwidth will decline dramatically, and it damages the fairness intensely [56].

2.3 TCP mechanisms and parameters involved in the performance of 5G networks

The transport layer has a significant role in determining end-to-end performance in a network [23]. Although the new mobile generation provides high bandwidth, without an effective transport layer, which is able to utilize the available bandwidth of mmWave in 5G networks and deal with the existing issues such as blockage and misalignment, this bandwidth will be wasted, and reaching high data rates will be challenging [22]. As we know, TCP is the most widely used protocol in the transport layer and is the key player in specifying end-to-end functionality. Various TCP mechanisms, such as congestion control and loss detection, can significantly affect the delivered performance to the final user. This section aims to give an overview of the different mechanisms, parameters, and analyze their effects on the performance of 5G networks.

2.3.1 TCP packet size

Adapting MSS to MTU (Maximum Transmission Unit) and optimizing its values for 5G networks is a challenge. The default value of MTU has been used for a long time and has been performing properly in the previous generations because the moderate bandwidth of them did not need a big MSS to deliver high throughput. On the other hand, MSS's conventional size has a couple of adverse effects on the performance of TCP over 5G networks. As an example, the small size of MSS degrades the performance because TCP cannot utilize the high capacity of the network in some circumstances.

An investigation on the impact of the size of MSS was done in [22], and the results showed that loss-based TCPs such as NewReno, adds up to their congestion window sizes slowly when the standard value for MSS is used, makes protocols underutilize the high bandwidth of 5G mmWave networks. If the size of MSS increases, it leads to faster growth of the sending rate, so a higher performance will be achieved in a shorter time, and it can also help when recovering from congestion states. When RTO triggers, the sending rate is initialized, and TCP enters the slow start phase. If the MSS is small, reaching high sending rates can take more time. However, having a large one can help to recover faster. In some TCPs' congestion avoidance phase, cwnd is added linearly, so the increased value of MSS can help the protocol ramp up quickly and utilize the available bandwidth. As a result, having a larger MSS in a network with a high data rate can assist the protocol in attaining higher throughput. We should notice that increasing MSS can compensate for the throughput degradation issue to some levels, and it is not a perfect solution, and the main focus should be on adapting the congestion control mechanism to 5G networks.

Moreover, when small MSS is exploited, more overhead is forced to the network because of the need for a large number of headers. As a result, using a large size for MSS reduces the number of overheads in the network. Finally, small MSS means transmitting more segments, leading to a higher number of ACKs in the networks, which can exhaust the network. A detailed analysis of MSS's impact on TCP's performance over 5G mmWave networks has been brought in [22], [62].

Our conducted simulations showed that loss-based TCPs could benefit from increased MSS. Comparing Figure 12 to Figure 8 reveals that loss-based TCPs can get considerable interest from a larger MSS size, and in some cases, the value of throughput is larger than the unlimited RLC buffer scenario. The reason is that when they detect a loss in a network and reduces their sending rate, it takes some time to ramp up to the possible sending rate, and this can even be worse when consecutive packet drops occur during a congested situation. However, by increasing the MSS size, these protocols can overcome this flaw.

From the RTT point of view, TCP experiences an increment compared to the 1400 byte scenario, as the acknowledging time can be longer. However, the achieved high throughputs for loss-based TCPs in all conditions can compensate for high RTTS. RTTs can be found in Figure 13.

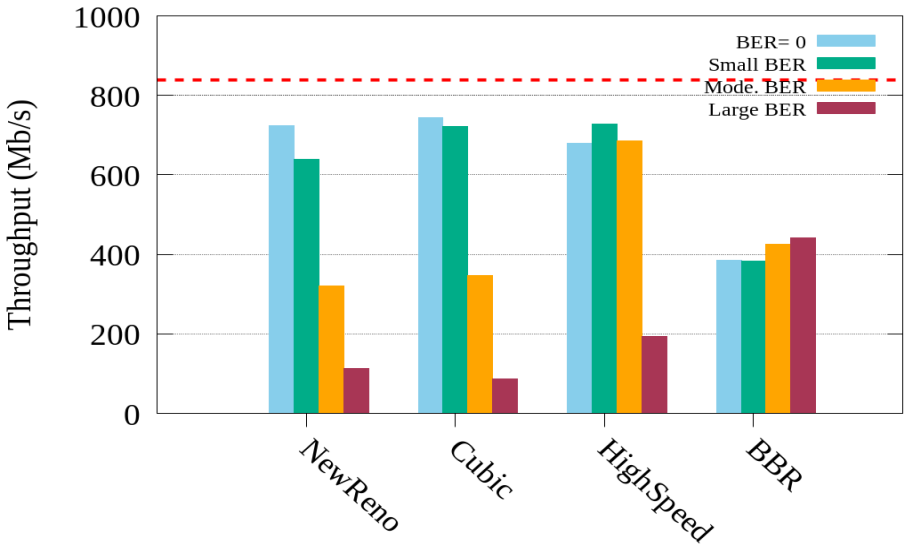

Figure 12. Average throughput for different TCPs, MSS =14000 bytes

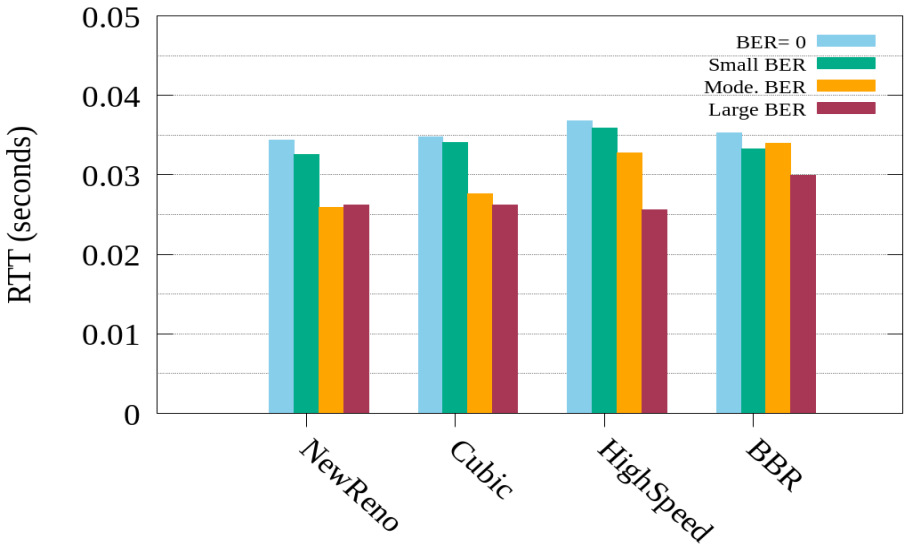

Figure 13. Average RTT for different TCPs, MSS=14000 bytes

2.3.2 Initial Congestion Windows Size

When TCP starts sending data, the first phase is the slow start. In this phase, congestion windows size starts from the minimum value, which can be one, two, or four [70], and then by receiving every ACK, TCP adds up cwnd size by one. Although this mechanism aims to probe the link and can be efficient in networks before 5G, it seems not suitable for the new generation. The first reason is that the sending data rate can be tremendously huge in 5G networks because

of the high bandwidth. However, the small starting number for cwnd can take a long time to utilize the full potential of 5G networks. Secondly, when an RTO happens and TCP enters the slow start again, initializing the cwnd and starting from one in a network that can support high data rates is astonishingly wrong. The first step could be using higher values for the initial congestion window, so TCP can benefit from getting higher sending rates by doubling it in the slow start phase. In this case, we should be careful about a premature transition from the slow start to the congestion avoidance. As a result, it may be necessary to modify the slow start threshold by considering the initial congestion window. Generally, a high value for the slow start threshold is used in 5G networks.

In addition to increasing the initial values of the congestion window, new approaches are essential to be proposed to deal with this issue. These approaches can be as simple as testing new values or proposing intelligent solutions to set the initial value of the cwnd based on the network parameters such as the loss probability, available bandwidth, cwnd size in the last packet drop, and time intervals between drops. It should be said that because TCP performs in the congestion avoidance phase most of the time, modifying the slow start phase and the initial congestion window has a low priority.

2.3.3 Exponential backoff Retransmission Time-Out

RTO effects can be severe in long-time disconnections, especially when buffers are small. When extended failures occur, the probability of triggering RTO is high, which can negatively affect 5G networks' performance. The cause of this degradation is triggering RTO when the reason is not congestion. There are circumstances where the network is not congested and performing well, but having an RTO triggered by an obstacle can also lead to initializing the cwnd size and entering the slow start, causing a dramatic reduction in the performance. These issues can be severe when static situations or a long distance between a UE and a gNB exist in the network because the chance of triggering RTO in each blockage will be high, and techniques such as handover seem useless in these situations. As a result, some solutions need to be proposed to prevent the performance degradation caused by RTOs during the blockage.

Link-layer retransmission [4], [71] is a method that can help to reduce the number of TCP retransmissions by hiding some of the losses from the transport layer. In this case, other layers of the 5G protocol stack, such as the MAC and RLC layers, try to mask some losses from the

upper layers, such as HARQ, which is one of the deployed methods. When the physical layer detects some losses in the received packets, it asks for a resend. In such a case, the responsibility of sending some redundancies is on the HARQ, so the physical layer can correct the error by exploiting these redundancies. In addition to the MAC layer, the RLC layer, which resides on top of the MAC layer, can do retransmission to some levels when the AM (Acknowledged Mode) is enabled. By considering the limited number of attempts in the MAC layer in recovering the lost packets, RLC can compensate for it and help in retransmitting more lost packets. Based on the received information from the receiver, RLC can detect which packets are lost in order to retransmit them. If the UM (Unacknowledged Mode) is activated in RLC, this procedure will be halted. Being timeliness is the advantage of the mentioned methods. However, some limitations are the downside of the link-layer retransmission compared to the TCP one. For example, the number of retransmissions in the MAC layer is usually limited to three attempts [71]. These retransmissions can aid TCP, especially in NLoS mode, in which the probability of losing packets is a high number.

The most crucial reason for link-layer retransmission and hiding losses from the upper layers is to give some guarantees in delivering packets. However, this can force shuffling in TCP's packets order and can lead to a reordering problem. Moreover, these retransmissions increase delay, so TCP RTO in some cases can also expire. In conclusion, tuning these parameters is also a challenge over 5G networks.

There is a comparison of the TCP functionality with and without link-layer retransmission in [71]. The results showed that in LoS mode, the distance between a UE and a gNB has a significant impact on the throughput of TCP. In this case, when the distance is low, deploying link-layer retransmissions does not have a significant effect, however, by distance increment, the throughput of TCP declines in the absence of link-layer retransmissions. In a nutshell, the principal reason for the throughput decrement without link-layer retransmission is that TCP cannot handle all of the retransmissions efficiently by itself, especially in NLoS modes. Furthermore, when packet losses are not hidden from TCP, the sending rate could be reduced frequently. We should notice that it is beneficial to have a trade-off between throughput and latency as the link-layer retransmissions can harm the latency value. As the results in [71] show, the best value for latency is for the time that only TCP retransmits the lost packets, i.e., no HARQ

plus RLC UM, however, it can decline the throughput. To sum up, deploying link-layer retransmission can help increase TCP's throughput, especially in higher distances at the cost of increased latency.

2.3.4 TCP congestion control mechanisms

When the first TCP was proposed, wired networks existed, and the Internet was connected through them worldwide. Furthermore, most of the networks were operating at moderate speeds, and there was no necessity for high-speed protocols. Thus, at first, TCPs did not have any mechanisms to adapt themselves to high-speed networks. As time passed, with the emergence of these networks and ubiquitous wireless technologies, new TCP variants emerged to adapt to these networks. As a result, TCPs such as HighSpeed [30], CUBIC [34], and BBR [57] appeared.

One of the most well-known congestion control mechanisms serving for a long time is AIMD, the default congestion control strategy in some TCPs, such as NewReno. This mechanism can perform adequately in networks with moderate congestion status but is not suitable for networks that need TCPs with aggressive approaches in increasing and decreasing the sending data rate. Moreover, existing random packet losses in wireless communication can mislead this mechanism and prevent it from functioning properly.

Most TCPs, especially loss-based ones, such as CUBIC [34] and NewReno [55], cannot perform appropriately in 5G networks due to the intermittent nature of the wireless channels and the blockage problem, primarily when servers are located remotely, which affects their performance negatively.

There is a technique that can enhance the congestion control functionality of TCP, called proxying [72]. When deploying proxying, a connection can be split into two separate connections, and then different congestion control mechanisms can be performed on the individual connections. In general, a proxy is a mediator between a client and a server and can reside on gateways or gNBs in 5G networks. Furthermore, performing different congestion control mechanisms, protocol mapping in gateways, and data caching can be the advantages of exploiting a proxy.

This section indicates that the previous congestion control mechanism can suffer from 5G mmWave channels' unique characteristics, and we need to design new ones in order to deploy the full potential of the coming generations.

2.3.5 TCP loss detection

As mentioned, conventionally, there are two ways for TCP to detect a loss in a network, three duplicate ACKs for indicating moderate congestion or triggered RTOs for heavy congestion.

Moreover, by deploying the TCP Selective Acknowledgements (SACK) [73] option, the sender will be informed of the successfully transmitted segments and retransmit only the lost ones. This mechanism prevents the sender from resending the correct ones. The use of the TCP SACK option increases the amount of packet overhead by improving the retransmission mechanism. However, being restricted to 40 bytes for the TCP option field forced by TCP specification is a hurdle in the way of implementing SACK in large BDP networks.

To sum up, the unique characteristics of 5G mmWave networks are barriers on the way of implementing TCP and having reliable end-to-end communication. As a result, making some modifications to TCP to make it suitable for 5G mmWave is necessary.

2.4 Related work

As it was mentioned in the previous sections, using TCP over 5G networks can be challenging. One of the most promising approaches to mitigate or eliminate the adverse effects of 5G mmWave networks such as blockages is modifying or adapting some mechanisms of TCP. Another alternative can be designing a new protocol from scratch and replacing the existing protocols. Both of the mentioned techniques can improve the performance of 5G networks and have their advantages and disadvantages. When designing a new protocol, existing issues can be addressed in detail, solved more efficiently, and the chance of resulting in an improved performance becomes high. However, there is no guarantee that it will work with other protocols, and there is a probability of having some problems such as fairness when coexisting with other ones. Moreover, testing environments may need to be modified in order to be compatible to evaluate new protocols accurately.

In addition, the main hurdle on the way of creating a new protocol is that it is almost impossible to replace a protocol in the Internet Stack because the existing ones are widespread on the Internet and have been around for a long time. For that, as an example, QUIC (Quick UDP Internet Connections) [74], the protocol developed by Google, is based on UDP, which intends to reduce end-to-end latency.

There exist several investigations on TCP over 5G, especially 5G mmWave networks [20], [22], [24], [64], [71], [75], [76]. The most significant motivation in modifying or optimizing TCP and making it capable of being deployed in high-speed networks, especially 5G mmWave, is its end-to-end reliability. The proposals of TCP over 5G need to overcome a variety of constraints like throughput degradation, latency increment, fluctuation in adjusting congestion window, and fairness issue when several flows co-exist [22], [77]. However, the first step to address these issues is to detect them and then set the goals. This section first addresses a more in-depth investigation of the general TCP proposals, then, TCP-based throughput enhancements are addressed. Other important investigations groups are explained as they are focused on latency, fairness, and multi flows versus a single flow.

2.4.1 A deeper investigation

In order to approach a problem, the first step is providing a clear insight into it. As a result, for finding the characteristics of different TCP variants, a thorough investigation of TCP over 5G mmWave was done in [22]. There, different TCPs in various situations were analyzed to have a more in-depth view of the protocol's functionality. The aspects that they evaluated were, deploying edge servers (with minimum RTTs on the order of 4 ms) versus remote servers (with minimum RTTs on the order of 40 ms), handover and mobility effects, different congestion control algorithms and their impacts, TCP packet size, and RLC buffer size effects. They were analyzed in two different scenarios, including a high-speed scenario where a UE is inside a moving train and a dense urban environment, and the main focus was on the first scenario. Four different versions of TCPs (NewReno, HighSpeed, CUBIC, and BBR) have been analyzed throughout the simulations. The results for the highspeed scenario revealed that when edge servers are deployed, it can improve loss-based TCPs because of the short control loop feature. However, there are some exceptions to this conclusion, and by using small buffers, the goodput for CUBIC and HighSpeed in remote server mode is higher than the edge server one. Among

the four analyzed TCPs, BBR shows the best performance along with using big buffers. However, it cannot reach the saturated achievable goodput in the urban deployment, which is 2 Gbps for the 28 GHz spectrum.

We should consider that high goodput can be attained at the cost of higher latency; however, by deploying edge servers with small buffers, this negative impact can be compensated for up to some levels. Among these loss-based TCPs, the best goodput is for HighSpeed, for it increases the size of cwnd aggressively in high BDP areas. Among NewReno and CUBIC, in remote server mode, CUBIC can perform better, however, in the edge server one, the opposite is correct. In the urban deployment scenario, all TCPs can attain the same average cell goodput, but the RTT values for each one can be significantly different. Especially when we have NLoS or inside building UEs, loss-based TCPs suffer from higher latencies. Evaluations exhibit that for satisfying 5G requirements (i.e., goodput larger than 100 Mb/s and latency lower than 10 ms) in an urban deployment, only BBR can perform well in accompanying an edge server deployment and under desirable channel conditions [22].

Simulation results revealed that TCP generally could benefit from edge servers due to the shorter response time. However, CUBIC has the lowest throughput in the edge server mode, as this value is for NewReno when remote servers are deployed. In addition to the location of servers, MSS's size can affect the functionality of TCP, especially the loss-based ones. For example, by increasing the size of MSS, CUBIC gets more benefits. In contrast, it does not have any positive effect on the performance of BBR. Moreover, if big buffers are exploited, HighSpeed can reach higher performance at the cost of latency. As a result, Implementing HighSpeed needs using some techniques of AQM to reduce latency. In contrast, BBR prefers small buffers where the performance of HighSpeed will experience a reduction, but its latency will be improved.

In addition to urban and high-speed scenarios, we should find out how TCP performs in an indoor environment, such as a train station. The authors in [64] have tried to answer this question, evaluate the effect of the human body as a blocker of 5G mmWave communication, and how using TCP-FSO (Free-Space Optical), which is one of the candidates for long-distance high-speed wireless communications can affect TCP performance. To attain this goal, an indoor train station scenario was simulated by using MATLAB 5G library. We should notice that, although

TCP-FSO has some similarities to CUBIC, some modifications have been adapted, such as the retransmission has been improved, the congestion control mechanism has become delay-based ACK, and improved ACK retransmission control has been used. Thorough information about TCP-FSO can be found in [78]. In the first step, the effect of the human-body blockage was evaluated. It was assumed that passengers in a train station act as blockers in low, medium, and high-density environments. Some other obstacles, such as pillars and walls, could block communications too. Results showed that when the UE is close to the gNB, the number of obstacles, which indicates the number of blockages, had a minor negative impact on the performance. In the second step, the value of SNR (Signal to Noise Ratio) was calculated at 28 GHz carrier frequency. For this, the actual channel at Haneda International Airport Terminal was observed. Results indicated that SNR could be degraded drastically by the blockages caused by human bodies.

Moreover, the distance between a UE and gNB is another factor that can play a significant role in the quality of the received signal. The third step was the 5G network bandwidth calculation. The MATLAB 5G library was used to estimate the bandwidth of the 5G network downlink. Finally, the evaluation of TCP throughput was done by considering the 5G network bandwidth simulation results. To sum up, an overall look on the results reveals that TCP-FSO can reach higher throughput compared to CUBIC when exploited over 5G networks. Moreover, the number of blockers and the distance between a UE and a base station have essential impacts on TCP performance. When there are a few obstacles and the distance is low, TCP can function more efficiently.

In addition to simulations, practical testing is paramount in achieving a clear view of a problem, which could happen after implementing 5G networks. As a result, one of the first practical evaluations of the commercial 5G mmWave networks was done in [60]. The test was conducted through the first world's commercial 5G in Chicago and Minneapolis provided by Verizon since April 2019. It is operating at a 28 GHz carrier with 400 MHz subcarriers. A Samsung Galaxy S105G has been deployed as the UE in this experiment. This evaluation was done at four different locations, including downtown, at the border of 5G coverage, inside a hotel, and near the U.S. Bank Stadium in a huge open environment. The aim of deploying these four cases was to emulate different deployment scenarios of 5G.

In order to evaluate TCP performance, a different number of TCP connections, including 1, 2, 4, 8, and 16, were tested to obtain throughput and RTT values. To utilize the high capacity of the 5G channels to their full potential, a TCP bulk download was performed. The results showed considerable enhancements for 5G compared to 4G in terms of throughput, which in some cases, it was ten times more than 4G throughput. Although 5G has a much higher throughput, it showed many fluctuations, even in LoS connections. The reason behind these fluctuations is that different layers, such as the transport layer, were not ready to be deployed in 5G networks, and they need some modifications. In terms of RTT, 5G could not exhibit significant improvements, and it shows slight enhancements compared to 4G. It is because of the NSA mode that makes most of the used infrastructure in 5G borrowed from 4G. It will be improved dramatically if SA mode is implemented in the coming future. Experiments in the presence of blockages such as human bodies, pillars, and trains showed that except for some thin materials like backpacks, cardboard boxes, or clear glass, most of them cause a drastic reduction in the performance of 5G. The reason behind it is that 5G signals cannot penetrate most of the materials, and when a blockage happens, a handover from 5G to 4G is initiated, which reduces the functionality of the network. This can somewhat be relieved in dense deployments where the signal reflection caused by obstructions such as buildings is helpful.

The simulation results in [22], [64] and the practical testing output in [60] revealed that TCP's functionality could be impaired in several aspects including, throughput, latency, and congestion windows adjusting. As a consequence, for benefiting 5G mmWave full potential, some efforts should be made. These efforts can include wide ranges from non-intelligence-based schemes to complex intelligence-based algorithms in improving different aspects of TCP.

2.4.2 Throughput enhancement

With the advent of 5G networks, the backbone traffic will increase intensely, and a need for a protocol to handle it efficiently is inevitable. One of the important attempts of designing a novel TCP for 5G networks was TCP Ohrid [75], which aims to improve throughput by attaining 400 Gbps data rates in the core network. The main purpose of TCP Ohrid is to manage the backhaul traffic to prevent collapse due to heavy congestion. The design of TCP Ohrid is based on HighSpeed TCP, with this main difference that it strives to have different responses to different speeds. As a consequence, the behavior of TCP Ohrid is up to the current speed of the network.

The goal of TCP Ohrid is to achieve at least 5 Gbps for mobile users under heavy mobility and a 400 Gbps data rate for the backhaul. Because of the different mechanisms of TCP Ohrid in approaching congestion in a network, it can reach a larger congestion window size than NewReno. However, the results revealed that it could not outperform HighSpeed TCP in terms of the congestion window, so it means that HighSpeed TCP can reach larger cwnd sizes compared to TCP Ohrid. The principal advantage of TCP Ohrid over HighSpeed is being more friendly to existing protocols and achieving comparable data rates to HighSpeed TCP, which makes this protocol suitable for being deployed in mobile communication and backhaul transmission [75].

When a catastrophic circumstance occurs, the necessity for establishing an instant wireless communication to transmit current videos to evaluate the site's conditions is inevitable. The best candidate to be deployed in these situations is 5G mmWave due to its high bandwidth and extremely low latency. However, collapsed buildings, broken trees, and other obstacles prevent 5G mmWave from functioning adequately. For that, another attempt to modify TCP to have a new scheme suitable for disastrous situations called DL-TCP (Deep-Learning TCP) was proposed in [20]. The main effort of TCP Ohrid was improving 5G network performance by mainly increasing the backhaul data rates, then the fronthaul by a less priority. However, DL-TCP aims to improve the fronthaul functionality by efficiently adjusting the congestion window size during disconnection occurrences in the network caused by blockages or misalignments and prevents the sending rate from being initialized wrongly in disastrous situations. In DL-TCP, an ML (Machine Learning) framework was developed to put a threshold between RTOs caused by congestion and the ones created by the blockage and misalignment. DL-TCP employs some parameters to divide the network into three parts, long-time failure, short-time failure, and congestion. When long-time failures happen, the network is going through a long-time disconnection. In such a case, DL-TCP uninitializes the cwnd size and prevents the resetting process of the sending rate. When short-time failures occur in the network, they are indications for short interruptions, so the algorithm intends to maintain the cwnd size and retransmits the most recent transmitted packets. The goal of this process is retransmitting recently lost packets caused by short disconnections. When congestion is detected, the algorithm decreases the cwnd size and enters the steady-state to work in its normal way. The used parameters in the deep neural network for DL-TCP for estimating the mentioned states are: "time" that is the time of the used SNRs, "location" which is the location information of the TCP sender, "velocity" which is the

current speed of the TCP sender, and "SNR" which is the SNR value received by the TCP sender and shows the signal quality. Authors in [20] evaluate the performance utilizing simulations in two different scenarios (small and big obstacles) and two different mobility modes.

The simulation results showed that the proposed TCP could outperform other TCPs. In terms of RTT, DL-TCP, NewReno, and CUBIC are similar, but BBR has high RTT values compared to the others. Comparing different cwnd sizes indicates that all of the protocols are experiencing intense fluctuations because of the intermittent nature of the channels. However, DL-TCP prevents cwnd initializing during the interruptions and lowers it in case of a congestion event, and can help adjust cwnd size more efficiently [20]. DL-TCP could reach high throughput for UAVs (Unmanned Aerial Vehicles) in disastrous situations; however, it lacks a well-designed evaluation method as the protocol was trained and tested in the same topology. Furthermore, it has not been compared to HighSpeed, one of the best candidates for the 5G mmWave networks.

Another approach called D-TCL (Dynamic-TCP) was proposed in [79]. This protocol's prime aim is to handle the adverse impact of random packet drops in 5G mmWave networks by appraising 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 could 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 [62], and not having a discussion on the obtained average RTTs can be mentioned as the downsides of D-TCP.

2.4.3 Latency and fairness

Besides throughput, latency and fairness are two other KPIs that need to be improved to adapt TCP to 5G mmWave networks. Latency is one of the critical features in 3GPP specifications for 5G networks, which pursues considerably low values close to zero. By improving latency, fairness will automatically be enhanced due to its direct correlation to latency because shorter latencies lead to faster paces in increasing the sending rate, so senders with shorter latencies can reach larger sending rates compared to the ones with higher values. As a consequence, having a fair network could be challenging in situations that latencies differ drastically.

A simulation analysis of TCP over 5G networks to see the impact of parameters such as RLC buffer size and RTO on TCP functionality has been made in [24]. RCL buffer size can significantly impact the latency and throughput by having the capability of masking losses to higher levels, especially the transport layer. At first, the effect of the RLC buffer size on the performance of higher layer protocols was analyzed, and the results indicated that exploiting buffers at the size of 1 MB, which was enough in the previous 3GPP mobile networks, is not good enough for 5G networks. As a result, they suggest deploying a seven MB buffer size with an RTO of 200 ms to replace the conventional one second to improve the functionality of TCP. However, the author did not generalize their findings.

In order to analyze multiple flows, different UEs using various applications were investigated. In their scenario, one of the UEs generates the heaviest traffic and moves around the environment to trigger the handover and affects the other UEs' performances. In this case, both blockage and long flows can exist at the same time. Results showed that most UEs have fewer retransmitted packets in YeAH than CUBIC, and generally, multiple flows can perform better when YeAH is deployed. However, when a UE is not affected by a heavy flow and is served by one gNB in the entire period (i.e., no handover is triggered), the retransmissions number is much less for CUBIC compared to YeAH. On the other hand, when a UE is served by several gNBs, it can affect CUBIC more than YeAH. For static users, the performances are similar, but the number of packets they need for retransmissions is different. When different flows exist together, CUBIC will retransmit more, but it can reach higher performance.

From the buffer using point, when medium-size flows are not affected by long ones, both protocols use the same buffer size, and ARQ (Automatic Repeat reQuest) at the MAC layer can mask packet losses caused by wireless errors to the transport layer. However, when a long flow exists, CUBIC deploys more buffers than YeAH and can attain a higher rate. Protocols like CUBIC that try to utilize the link’s capacity and have a quick recovery mode perform well during NLoS disconnections but not very well when long NLoS ones exist. On the other hand, protocols with a hybrid mechanism like YeAH (which uses packet losses and RTTs) have fewer performance variations. Moreover, A comparison between throughput and RTT for different TCPs can be found in [56], which indicates different reactions of each TCP to various delays.

As the authors in [24] suggested some simple mechanisms such as modifying RLC buffer size and RTO value, then analyzing TCP functionality, the authors in [56] strived to enhance latency and fairness by leveraging sophisticated and straightforward schemes.

They sought the root of the problem in quick paces of buffers fillings in NLoS states when queue sizes are large. In contrast, deploying a small buffer leads to an underutilization of TCP performance. The base suggested solution to handle the problem is deploying AQM techniques such as CoDel [68] and Fq-CoDel [69], which drop packets before the queue is full. However, these techniques need some modifications in order to work appropriately in 5G networks. The first choice for tackling the fairness issue is exploiting Fq-Codel, which behaves each flow differently in queuing, and tries to maintain fairness among them. However, this technique is not able to perform appropriately in 5G mmWave networks. This malfunction of Fq-CoDel in 5G networks is due to the harsh effects of NLoS disconnections, especially long failures during static conditions. Moreover, dropping many packets during a NLoS period by AQM techniques forces TCP to enter the fast retransmit or the slow start phase, then after switching to LoS, it takes a long time for TCP to gain the possible high performance.

The first proposed solution in [56] is called on-off. In this case, when the network goes through a NLoS situation, CoDel and Fq-Codel will be disabled and are not able to drop packets. This approach prevents massive packet drops throughout the NLoS period. This can be achieved by setting the target parameter to five seconds to mimic a disabled state. The second scheme, which is more complicated than the first one, can perform even better. In this case, the RTT for each flow needs to be estimated, then based on the estimated values, the target parameter for each flow is calculated and used. Results show that better fairness can be achieved in the second approach compared to the on-off one. Using CoDel + on-off, Fq-CoDel + on-off, and tuning (i.e., the RTT estimation approach) in accompanying NewReno and CUBIC could lead to almost constant fairness during different LoS/NLoS conditions even when the distance between the UE and gNB increases. Especially, Fq-CoDel + on-off exhibits nearly the same fairness independent of NLoS time.

In addition to fairness, exploiting these approaches can affect the value of the delay parameter, and when CoDel + on-off mode is deployed, the average delays for different flows can be almost the same. However, the delay values for the three approaches are different, and in most cases,

the best value is for Fq-CoDel + tuning. By deploying the three suggested schemes in [56], fairness can be improved at the cost of 10 ms of more delay. However, when Fq-CoDel + tuning is used, this number can be reduced to 5 ms by negatively affecting the fairness.

The principal reason for improving fairness is that it is one of the ultimate goals of TCP congestion algorithms that is desired to be obtained along with high throughput while preventing congestion in the network. In order to increase the performance of networks, buffers are used to avoid dropping the packets that are experiencing short-lived traffic peaks. In general, buffers suffer from two drawbacks: a weakness in managing the queues and TCP congestion control failure, which can lead to higher latencies and underutilize the network's available bandwidth. Moreover, full buffers in a network, which are reasons for higher latencies, end up in bufferbloating problems, one of the most significant issues in deploying buffers. This problem can be intense when we know a tremendous number of buffers have been installed throughout the Internet without having efficient strategies in controlling the queues. These buffers can degrade TCP's performance and be hurdles in the way of this protocol in accomplishing its aims [56], [80]. The existing techniques encounter some challenges in 5G networks and need to be renovated to adapt.

On the one hand, the proposed solutions in [24], [56] can enhance latency and fairness to some levels. On the other hand, they are not adequate enough to meet 5G mmWave networks' desired values. As a consequence, some advanced algorithms should be proposed. One solution can be providing intelligence to AQM techniques by using ML approaches to make the dropping mechanism more effective so that they can handle the issues caused by buffer size and packet dropping in the queues more accurately. In such a case, the AQM techniques static mechanism will be modified and will replace with smart schemes, so they will look at the existing parameters, and based on them, decide to drop a packet or not. The ultimate goal of new algorithms can be predicting the behavior of a network and drop beforehand in order to provide a tradeoff between throughput and latency. One of the most promising ML techniques that can be convenient in redesigning AQM techniques can be RNN (Recurrent Neural Network) because this technique provides feedback from the previous states, which are helpful in improving AQM techniques.

Another technique can be bringing the cloud close to UEs, which is called fog networking [81], [82]. In this case, a node such as APs (Access Points), small cells, or routers can be a fog node

that provides services to other UEs. One of the most critical questions in fog networking is the location of the fog nodes (i.e., which nodes are the best candidates to be selected as the fog nodes), especially in heterogeneous networks, in the combination of HPNs (High Power Nodes) with LPNs (Low Power Nodes), where some LPNs are selected to be upgraded to become fog nodes in order to improve the performance of the network. One way could be to divide the nodes into some clusters and then choose a node in each group as the leader. ML techniques can be beneficial to be exploited in order to reach this purpose. As a result, an unsupervised ML approach was proposed in [19] to answer the central question in fog networking about which LPNs should be upgraded to become fog nodes. This algorithm is based on unsupervised soft clustering machine learning. In this case, all of the LPNs are divided into separate groups, and then in each group, a node is selected as the head of the group. After that, all of the heads turn into fog nodes. One of the ultimates of this approach is improving the k-means hard clustering, in which each node chooses the corresponding fog node based on the closest Euclidean distance. This approach, executed by deploying the Voronoi Tessellation model, can lead to a poor channel connection because there is no guarantee that the channel between the node and the closest fog node has the best quality. As a result, performance and latency will be degraded. By using an unsupervised ML approach, the proposed algorithm is able to enhance the latency, which is one of the most critical issues on the way of deploying TCP over 5G networks.

2.4.4 Multi-flows versus a single flow

Because new cellular devices intend to use several interfaces deploying MP-TCP (Multi-Path TCP) [83] can have some advantages compared to other TCPs. The key feature of MP-TCP is its capability with multipath communication, which means that when a socket establishes TCP connections, it can handle more than one flow related to different applications. These interfaces can incorporate various types of communications, such as Wi-Fi, a cellular network, and an Ethernet connection. The significant feature of MP-TCP that makes it different from conventional TCPs is how it handles the cwnd size in different subflows, which can be coupled or uncoupled. When deploying the latter one, each subflow is treated independently, and cwnd sizes for them are adjusted separately.

In contrast, in the coupled mode, all of the cwnds are adjusted in a correlated way. As a result, the congestion control algorithm of MP-TCP includes two different approaches. The ultimate

goal of this separation is an attempt to achieve the main purposes of MP-TCP, which are: 1) the minimum performance of MP-TCP should be as good as a single-path TCP; 2) the deployed resources by MP-TCP should not be more than conventional TCPs; 3) it should be capable of navigating more packets to uncongested paths. By considering the aims, deploying MP-TCP can be one of the solutions for having a trade-off between throughput and latency.

The analysis of MP-TCP over 5G and LTE networks was done in [76]. The simulation results showed that MP-TCP could outperform SP-TCP (Single-Path TCP) about 30-40 percent in the LoS conditions when deploying in 5G networks. However, it is not true when 5G and LTE coexist together. We should consider that in contrast to 5G mmWave, in LTE, the distance between a UE and a gNB is not a key factor. Thus, in higher distances, MP-TCP can perform better in LTE and mmWave than when deployed only in 5G mmWave networks. Moreover, Simulation results revealed that the latency value gets higher by increasing the distance between a UE and gNB.

It is worth mentioning that MP-TCP with a coupled congestion control mechanism shows a poor performance compared to CUBIC and cannot fulfill the first goal of the MP-TCP designing. The cause behind it is that in the congestion control process, MP-TCP assumes mmWave as a congested path due to its high loss probability and tries to transmit packets through the LTE links. In contrast, this issue does not exist in the uncoupled mode. Another problem is that when the uncoupled MP-TCP coexists with SP-TCP due to their unfriendly nature, it leads to an unfair network. All the mentioned problems indicate that more efforts need to be made to design an MP-TCP to fulfill all the goals.

2.5 Methodology for designing new protocols

To evaluate the design of a new protocol or improve a specific part of a network, we need to perform detailed analysis, which can be achieved in real scenario tests, analytics models, or simulation environments. In this thesis, the most convenient way of evaluating a new protocol is by exploiting simulation tools. These tools have evolved in a way that can mime real scenarios.

The first step is selecting a powerful simulation tool to evaluate TCP's behavior in various situations. The chosen tool should be selected based on the needs and the targets that are

followed, and to achieve this goal, extensive simulations must be conducted. The following aims to analyze different simulation tools, their advantages and disadvantages, and select one that can help us accomplish our goals.

One of the popular simulating tools is called LENA [84], an LTE-EPC network simulator. LENA is an open-source product-oriented LTE/EPC network simulator that supports LTE small/macrocell vendors to create and test Self-Organized Network (SON) algorithms and explications. Target applications for LENA include the design and performance evaluation of downlink and uplink Schedulers, Radio Resource Management Algorithms, Inter-cell Interference Coordination solutions, Load Balancing, Mobility Management, HetNets (Heterogeneous Network) solutions, end-to-end QoE (Quality of Experience) provisioning, Multi-RAT network solutions, and Cognitive LTE systems. LENA is based on the well-known ns-3 network simulator, and the expansion of LENA is open to the community to promote early adoption and contributions by industrial and academic partners [85]. The suitability of LENA for simulating LTE networks, i.e., LTE and its core, makes the process of simulation more convenient. It is also capable of simulating different layers such as RRC (Radio Resource Control), PDCP (Packet Data Convergence Protocol), RLC, physical layer, MAC layer, different channels, and antenna models.

Another powerful tool for simulating 5G mmWave networks is ns3-mmWave [86]. The ns-3 [87], [88] is a robust network simulator with discrete-event simulations. The ns3-mmWave module is an extension of ns-3 that is capable of simulating a tremendous number of different networks. The first introduction of the ns-3 mmWave was in [86], [89], the channel model implementation was proposed in [90], and the dual connectivity was explained in [91], [92]. A comprehensive description of ns3-mmWave is available in [93]. This module is a powerful and accessible tool that can simulate various aspects of 5G mmWave, such as the corresponding layers, channels defined in 3GPP specifications, and many more. Besides its powerful features, it is an open-source simulator, and the architecture of ns3-mmWave builds up on the ns-3 LTE module (LENA) [84], [91]. One of the main aspects of this module is its availability of connecting it to a Direct Code Execution [94], so the Linux stack TCP/IP can be run as the TCP/IP stack of ns-3 nodes. Moreover, the wide range of selection from 6-100 GHz channels, which is the official 3GGP channel model [95] and is described in [90], is another powerful simulating tool in ns-3

mmWave. To sum up, ns-3 mmWave is the one with numerous features that can help researchers have a strong and robust testing environment.

Another tool for simulating 5G mmWave networks is using a library supported by MATLAB [96], which can provide the main features of 5G networks. It includes standard-compliance functions and reference examples that can be used in order to model, simulate, and verify 5G communication systems. By using this toolbox, configuration, simulation, measurement, and analysis of an end-to-end connection is available.

There is another tool for simulating 5G networks called K-SimNet, proposed by Seoul National University [97]. It is an extension of ns-3 that can support 5G NR, 5G core, multi-RAT protocols, traffic management on multi-connectivity, SDN/NFV, and other features of 5G, which make it capable of simulating 5G end-to-end networks.

To sum up, ns3-mmWave is one of the primary candidates for a researcher who wants to evaluate a 5G network. This ns-3 based software includes the majority of the 3GPP features such as channel modeling, supports dual connectivity, and more. Moreover, being around for a long time is a powerful aspect of the NS series.

After selecting an appropriate simulation tool, the second step is defining scenarios and choosing parameters that can satisfy the most desired conditions. In our way to enhance the performance of reliable end-to-end communication, we have designed various simulation topologies and evaluated the network's performance to ensure that our data is valid. The first part incorporated an evaluation of the behavior of TCP and 5G mmWave in detail. The second part included the design and evaluation of new protocols to improve the networks' functionalities.

After designing proper topologies and protocols, the third step involves conducting extensive simulations and collecting the results. As a consequence, numerous simulations should be run in order to draw the results, and then the final step is comparing the new outcomes with the legacy ones in order to ensure that the new schemes are functioning efficiently.

Regarding software usage, C++ is being used in the ns3-mmWave module in order to implement the network topologies and new protocols. Moreover, Python accompanying

Tensorflow [98] along with the customary libraries, including NumPy [99], Pandas [100], and Matplotlib [101], are the tools for developing and training the machine learning proposals.

By combining the mentioned tools, we could develop a structure for simulating 5G mmWave networks' functionality to evaluate and enhance the performance. That is the framework we have chosen for the Ph.D.

2.6 Conclusions

The ultimate aim of 5G is to provide 1GHz frequency channels with the help of the broad available spectrum of mmWave. However, higher spectra have unique characteristics that create new barriers in front of having reliable end-to-end communication throughout 5G mmWave networks. These challenges did not exist in the previous generations, or their impacts were negligible, which could be ignored. One of the most critical issues is the blockage, which could affect the performance of the transport layers protocol in functioning properly, and if TCP cannot work correctly, having reliable end-to-end communication will be tough to be reached. Moreover, user experience performance relies mainly on the functionality of TCP; as a result, it will be degraded drastically. Some of the issues that need to be addressed are:

- The delivered performance to end-users will be degraded dramatically.
- The sending rate adjustment will be a real challenge as the channels fluctuate abundantly.
- RTT will be affected, and it will be risen due to the blockage issue.
- Attaining the saturated throughput in the network will be challenging.
- High achieved values for cwnd in conventional TCPs can exhaust the buffers.
- Having a well-performed network in an Urban deployment will be strenuous, as users move desultorily and could be behind or inside an obstacle almost all the time.
- As conventional TCPs are not able to work properly, the main aim of solving the issues could be designing intelligence-based protocols by deploying techniques such as Fuzzy and Deep learning.

All these issues indicate that efforts should be made to design and create a new transport protocol, which is able to tackle all the mentioned issues by establishing well-performed reliable end-to-end communications.

3 FB-TCP: A 5G MMWAVE FRIENDLY TCP FOR URBAN DEPLOYMENTS

This section 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 avoid impacts of blockage. 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.

3.1 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 (Fuzzy-Based TCP) protocol proposed in this thesis. Fuzzy logic [102] or Fuzzy sets, a subset of AI (Artificial Intelligence), is for indicating the 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 offers full membership, and the values between them are for partial membership [102].

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 humans 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. 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 [103].

As a result, Fuzzy logic is a suitable paradigm for decision-making and clustering problems, which can be used in complicated systems. In our work, 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 is to divide the network into different clusters and then adjust the sending rate based on the current cluster. Because of the nature of the Fuzzy, it can address various aspects in 5G mmWave networks. One of these features is handover, 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 design. SDN/NFV 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 signals; as a result, it can assist in reaching an ultra-lean design [59]. All of the mentioned aspects can be refined with the help of Fuzzy logic. However, in this thesis, we have decided to concentrate on the protocol side of the communication and improve throughput, latency, and sending rate adjustment.

3.2 FB-TCP: Fuzzy-Based TCP

As 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 long-lasting 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 desirable ranges is essential. In this case, LoS and NLoS states can be distinguished, and proper functionality can be achieved. The critical aspect of the protocol is the time that the UE is in a NLoS state. In these cases, the cwnd should be adjusted carefully to prevent the buffer overflow and keep the RTT as low as possible along with high throughput.

FB-TCP strives to handle the issues 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 states 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 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 14 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.

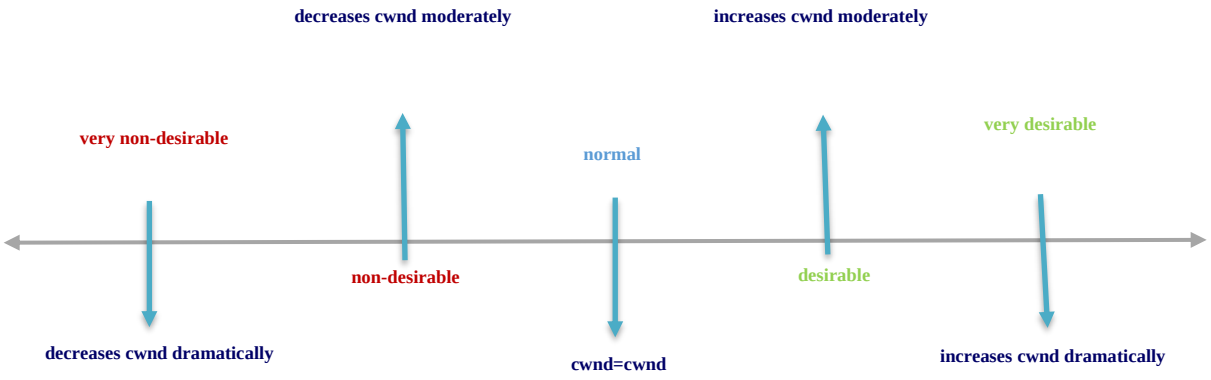

Figure 14. How clustering works

After defining the states, 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 (2), the maximum value for cwnd is figured:

$$maxCwnd = ((DP * minRtt) / 8)/MSS * \rho \qquad (2)$$

Where maxCwnd is the maximum value needed for cwnd to reach the highest available throughput in the network, the DP (Delivered Packets) is the number of delivered bits every 100 ms in our experiments (a tradeoff of simulation time and performance), dividing into eight is for converting the bits into bytes, and minRtt is the minimum RTT in the network for a congestion window. MSS 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 because we want to omit the adverse impact of packet drops, especially the random ones. 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 the 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 for dividing the network into two major sections is to determine the upper bound of the network and make 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.

In contrast, being above this spot indicates that the cwnd is forcing more packets than the network's current capacity, and FB-TPP should decrease the cwnd. As seen in Figure 14, 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 14. 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 that for the Divergence.

As the network moves toward non-desirable states, FB-TCP employs a conservative approach to relieve the intense 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, one of the employed parameters, is calculated by deploying (3):

$$Diff = currentCwnd - targetedCwnd \quad (3)$$

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 minRTT and desired throughput, which is calculated based on (4):

$$targetedCwnd = DesiredThourghput/minRTT \quad (4)$$

Where DesiredThourghput is calculated based on (5):

$$DesiredThourghput = currentCwnd * baseRtt \quad (5)$$

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 distinct 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, i.e., cwnd is lower than its possible maximum value. 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 issues.

Moreover, while the protocol sets the sending rate around the highest estimated available data rate, it does not blindly increase the cwnd because it has a clear insight into 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 feedback from the network and is a model-based TCP, it is immune to random packet drops.

3.3 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 in desirable states or control the sending rate in undesirable ones. 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 some cases. However, this mechanism helps FB-TCP to probe the bandwidth more appropriately and calculate the parameters precisely. 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 (6) is correct, the Convergence phase starts:

$$currentCwnd <= maxCwnd \quad (6)$$

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 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 (7):

$$CAD = targetedCwnd / currentCwnd \quad (7)$$

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 shown in Figure 15, 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 not 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.

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

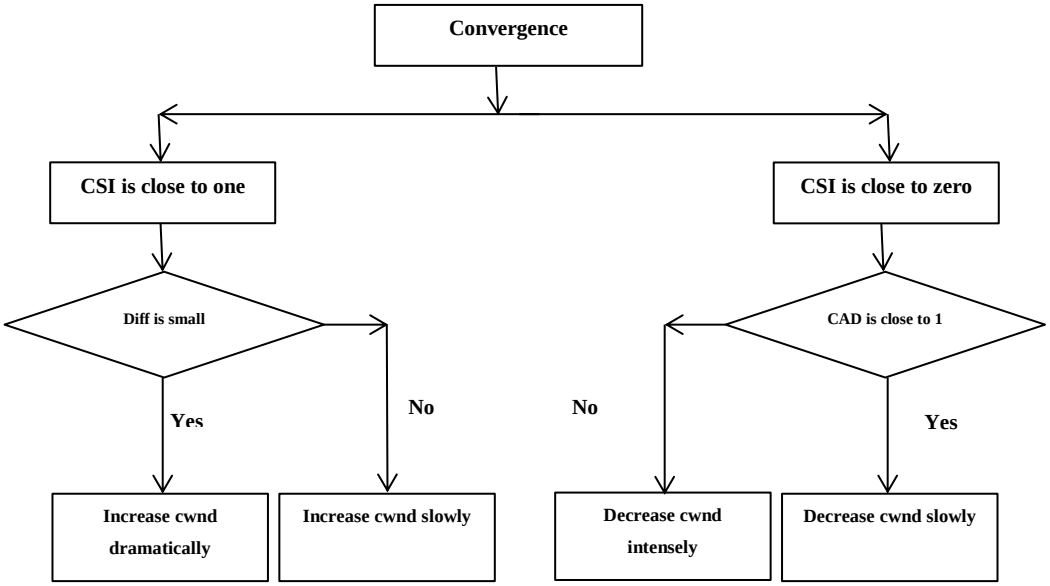

Figure 15. How the Convergence phase functions

One of the Fuzzy primary memberships that FB-TCP employs in its initial steps to divide the network into different sections is CSI. 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 IV, Table V, Table VI, and Table VII. 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.

TABLE IV

HOW THE CONVERGENCE PHASE FUNCTIONS- SUB-PHASE 1 (INCREASING SUB-PHASE)

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

As the value of Diff is close to zero, it shows ideal conditions; thus, FB-TCP can increase 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 are 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 V. However, we believe that the protocol is flexible, and values can

be changed in order to adapt to different scenarios. For example, if the RTT is the most important KPI, the minimum RTT can be attained at the cost of throughput by tuning the values.

The increasing sub-phases 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 IV, (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.

TABLE V

DEPLOYED PARAMETERS IN THE INCREASING SUB-PHASE OF THE CONVERGENCE

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

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 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. The rules for the other sub-phase can be found in Table VI.

TABLE VI

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

If FB-TCP remains for thirty consecutive RTTs in C10, (8) will replace currentCwnd=currentCwnd, i.e., the sending rate will not be kept fixed and instead of currentCwnd=currentCwnd equation (8) will be deployed and adjust the cwnd. 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 tuning the parameters for various scenarios and layouts has been kept open in a way that the protocol has the capability of being altered.

$$currentCwnd = \alpha * currentCwnd \quad (8)$$

Where α is for keeping a tradeoff between throughput and RTT, as we increase α, the throughput value will be improved. 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 (9) to lowers the upper bound and reduces the aggressiveness of the protocol when it remains in C10 for more than thirty consecutive RTTs:

$$maxCwnd = \beta * maxCwnd \quad (9)$$

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" in Table VI identify the time that the protocol remains in the same sub-phase, i.e., the same state, for more than two consecutive RTTs. It means that in normal conditions "b" sub-phases are not employed by the protocol, however, when some new situations are appeared (explained before), "b" sub-phases are exploited instead of the main ones. 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, not C12. 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 b sub-phases, as the parameter values indicated in Table VII.

TABLE VII

DEPLOYED PARAMETERS IN THE DECREASING SUB-PHASES OF THE CONVERGENCE

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

We have selected 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 [28], [104], [105]. Considering this fact, our protocol's targeted scenario is the urban deployment; nevertheless, it can show sufficiency in other ones.

For choosing the values that can cover a vast range of urban deployments, shown in Table V and Table VII, 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 preventing cwnd fluctuations. Moreover, the protocol can have stable functionalities through various conditions and is immune to the network's changes.

3.4 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 16 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 there could be more capacity. In contrast, when it is not close to one, the combination of CSI and CAD shows that the network's functionality is neither very desirable nor non-desirable; thus, the sending rate can be kept fixed.

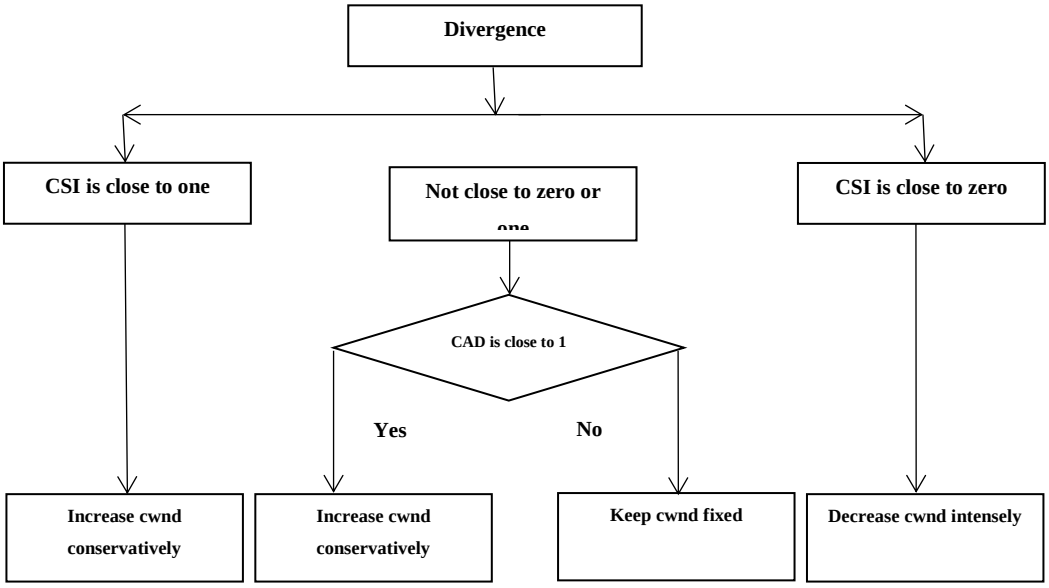

Figure 16. How the Divergence phase functions

The main constructed framework in this phase aims to create some suitable clusters for the time when 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 VIII indicates how the Divergence phase functions. When FB-TCP remains in D3 and D4 for thirty consecutive RTTs, (10) 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.

$$currentCwnd = \gamma * currentCwnd \qquad (10)$$

TABLE VIII

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

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 will be set by using (11) to drain the buffers, where θ equals 0.9 in our simulations:

$$maxCwnd = \theta * maxCwnd \quad (11)$$

D5.b and D6.b indicate that the status of being in D5 has D6, respectively, 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 IX. All the factors in "b" sub-phases can be selected from zero to one. However, the aggressiveness of the protocol has a paramount role in choosing the ideal one.

TABLE IX

DEPLOYED PARAMETERS IN THE DIVERGENCE

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

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 the 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 outperform NewReno, CUBIC, HighSpeed, and BBR. The code for the FB-TCP is available online [106].

3.5 Simulation scenarios and 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. we have evaluated the functionality of FB-TCP in four different scenarios to ensure the conclusions' validity.

3.6 Scenario one, short NLoS scenario

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 17 depicts the exploited layout in scenario one.

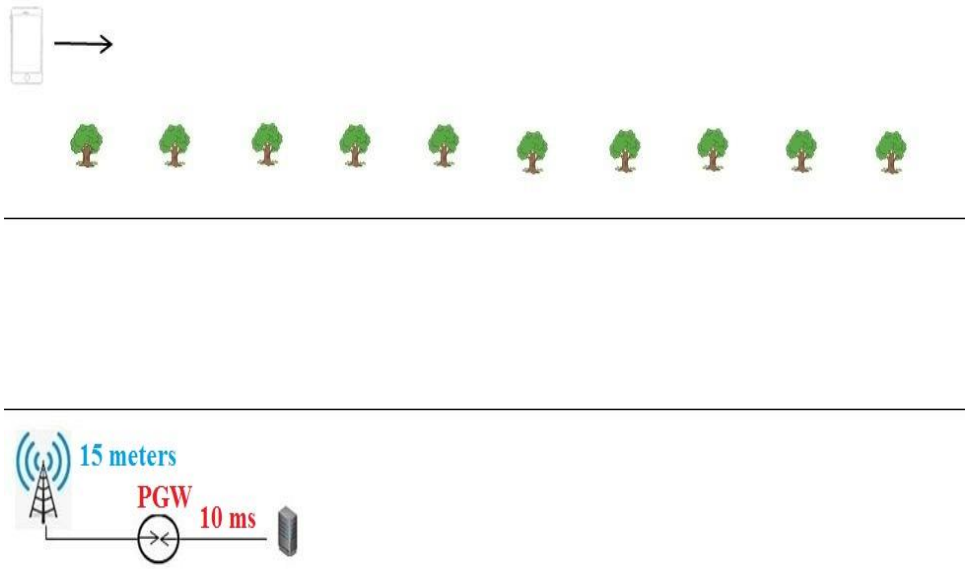

Figure 17. Scenario one

3.6.1 Simulation results for scenario one

The obtained results in the first scenario confirm that obstacles can create blockage states and make the received SINR weaker, which is the main reason for TCP's confusion. The value for SINR is depicted in Figure 18.

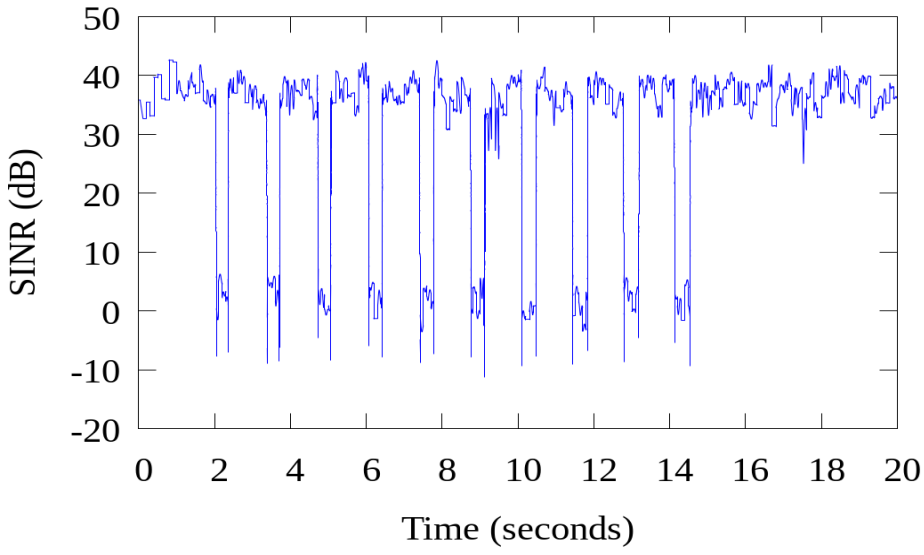

Figure 18. SINR fluctuation

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.

Conventional TCPs can not distinguish different states in a network, which is the main source for their performance reduction.

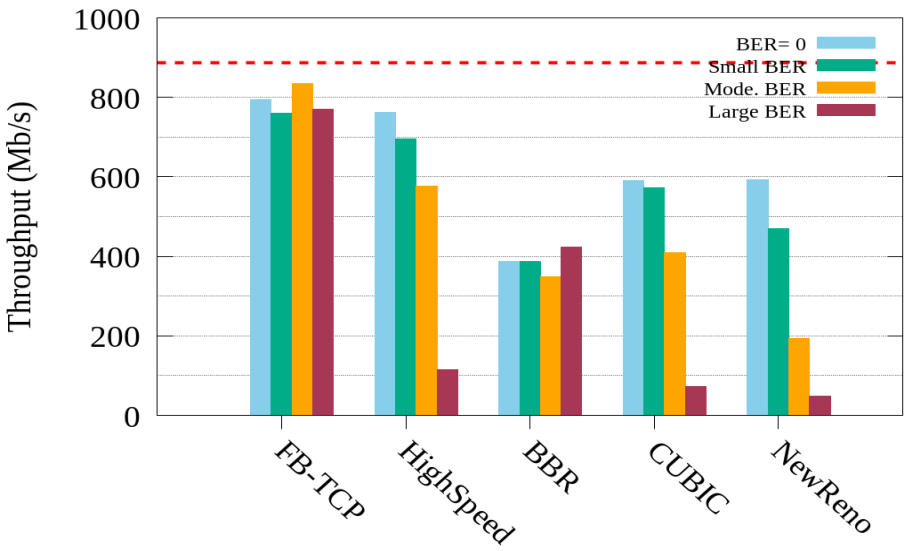

Figure 19. Average throughputs for different TCPs

By looking at Figure 19, 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 dashed line in the figure. 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 constant 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 apparent in Figure 20, when the throughput degrades dramatically.

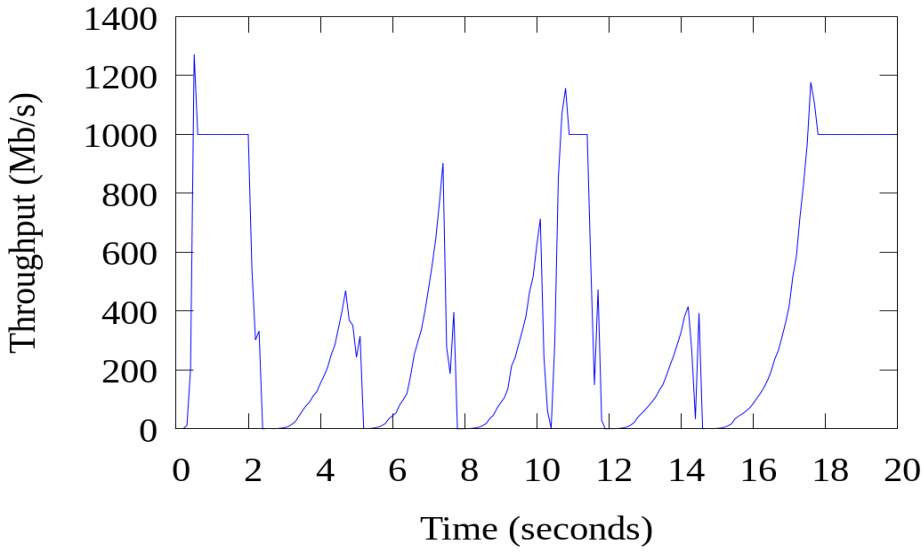

Figure 20. Throughput for BBR, BER=0

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.

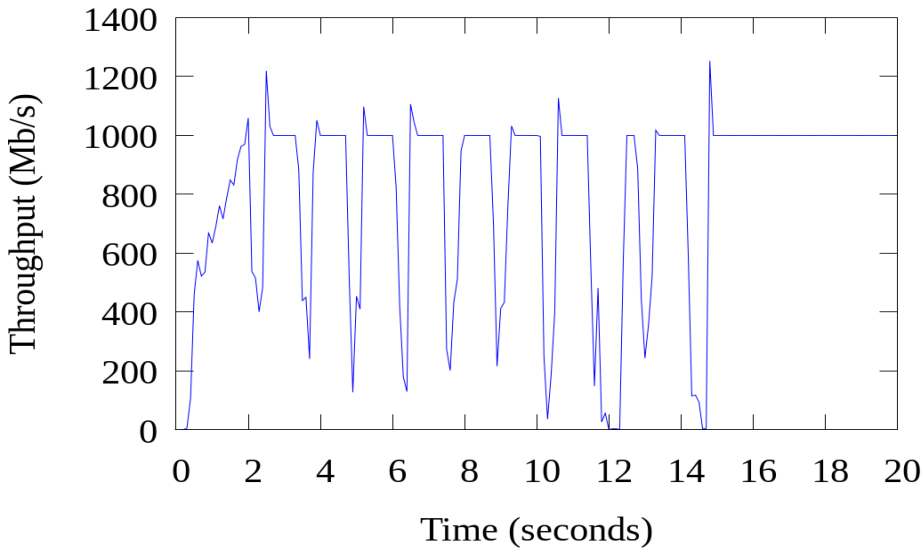

Figure 21. Throughput for FB-TCP, BER=0

Figure 21 shows the instantaneous throughput for FB-TCP when there are no random packet drops in the network. If we compare this figure to Figure 22, it is shown that there are intense drops when BER is zero. The fewer drops of the protocol when BER is moderate is 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. The tops after each blockage are because of the RLC buffer that can store some packets and after transitions to LoS states, these packets can force a sudden increment in the throughput as the TCP continuing with its former sending rate without any knowledge from the current situations.

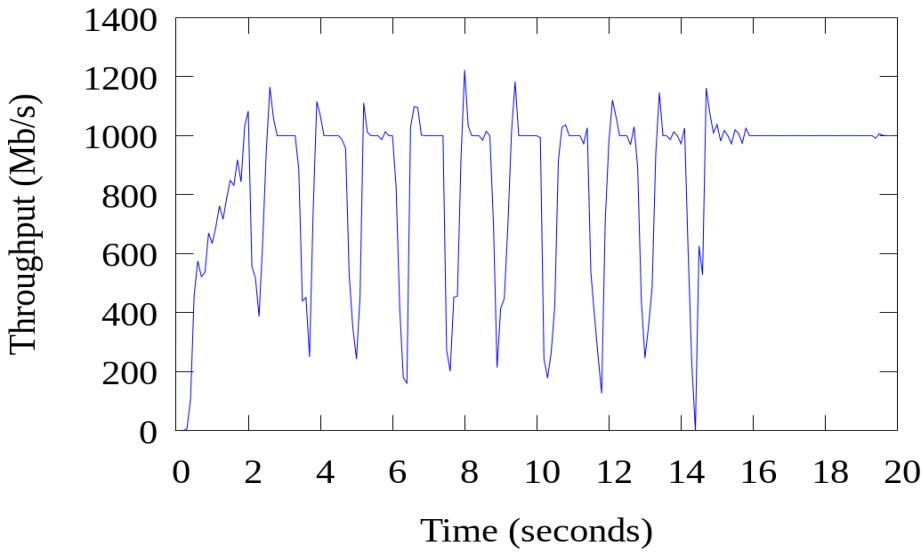

Figure 22. Throughput for FB-TCP, Moderate BER

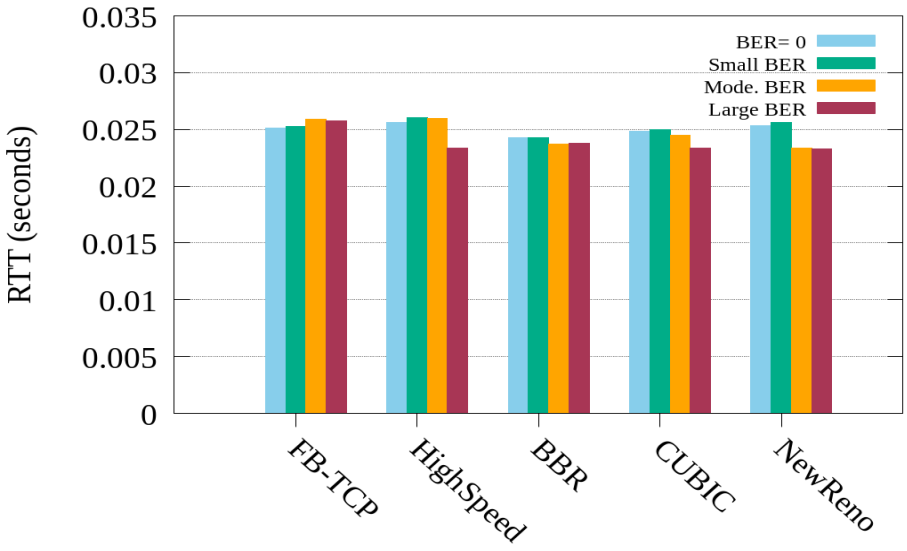

Figure 23. Average RTTs for different TCPs

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 23 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. Another important result is that RTT seems not to be affected by BER oscillations compared to other TCPs, because of its non-loss-based approach.

Based on the average throughputs and RTTs, we can compare FB-TCP and HighSpeed as the best candidate of the conventional TCPs to see the differences between them and having a more precise insight for the following scenarios. Figure 24 indicates that FB-TCP has a stable functionality, can react adequately to different situations, and 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 small delay in utilizing the full potential but gives a clear insight to the protocol from the network.

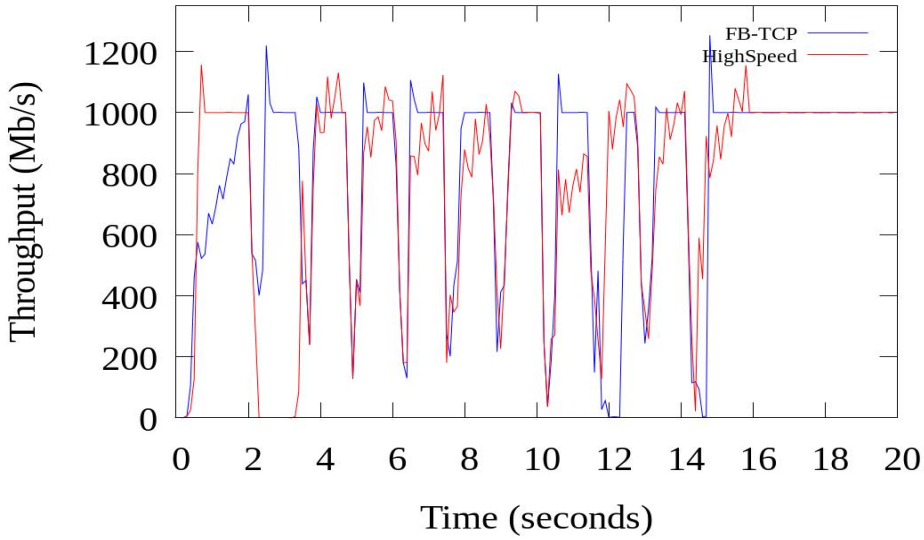

Figure 24. FB-TCP and HighSpeed throughput comparison, BER=0

The RTT comparison indicates that both protocols have similar functionality, as shown in Figure 25. 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

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.

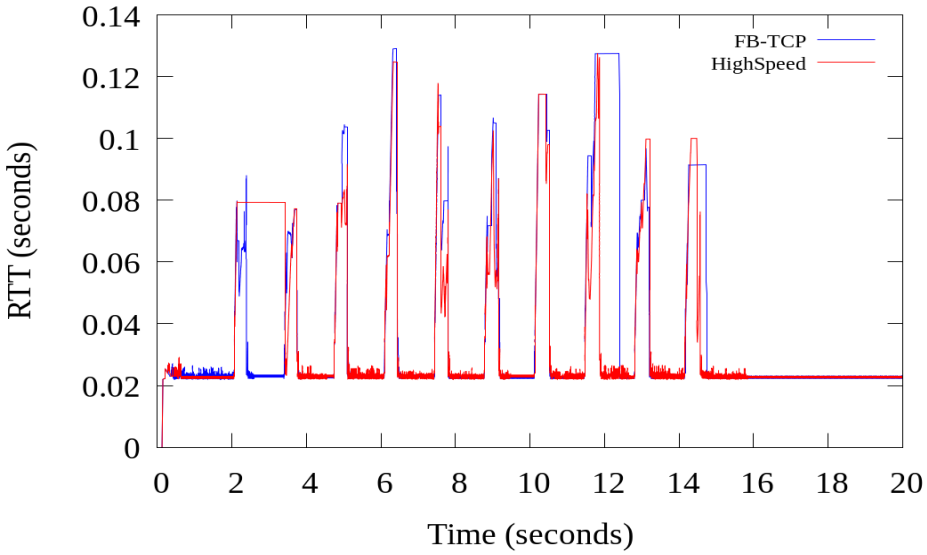

Figure 25. FB-TCP and HighSpeed RTT comparison, BER=0

Figure 26 shows how HighSpeed controls the sending rate. The slow start threshold should be a very high number [107], 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.

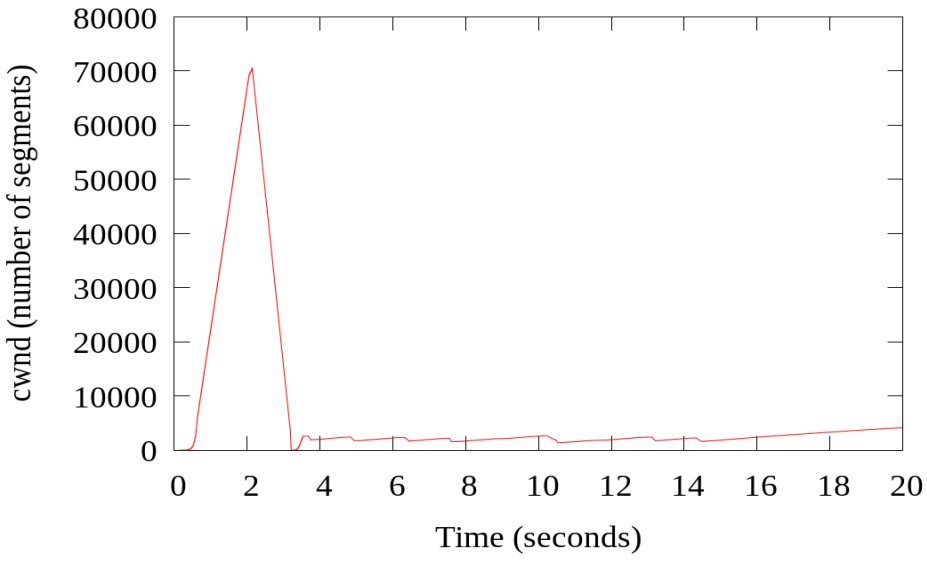

Figure 26. cwnd adjustment for HighSpeed, BER=0

As a result, a large slow start threshold is exploited for conventional TCPs in 5G mmWave networks, and the window scaling option is enabled [59]. 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, CUBIC from 591.39 Mbps to 532.18 Mbps, and 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 27.

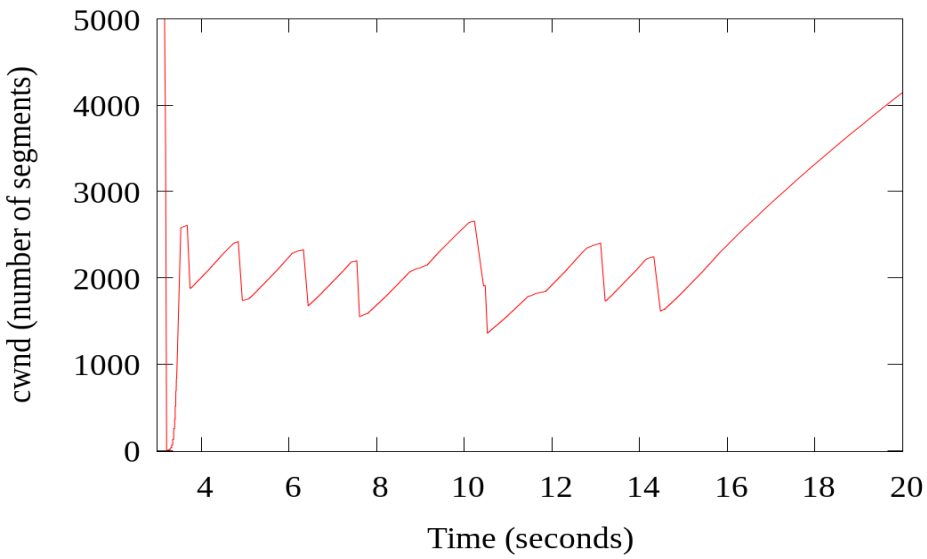

Figure 27. How HighSpeed adjust the cwnd in the congestion avoidance phase, BER=0

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 28. 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.

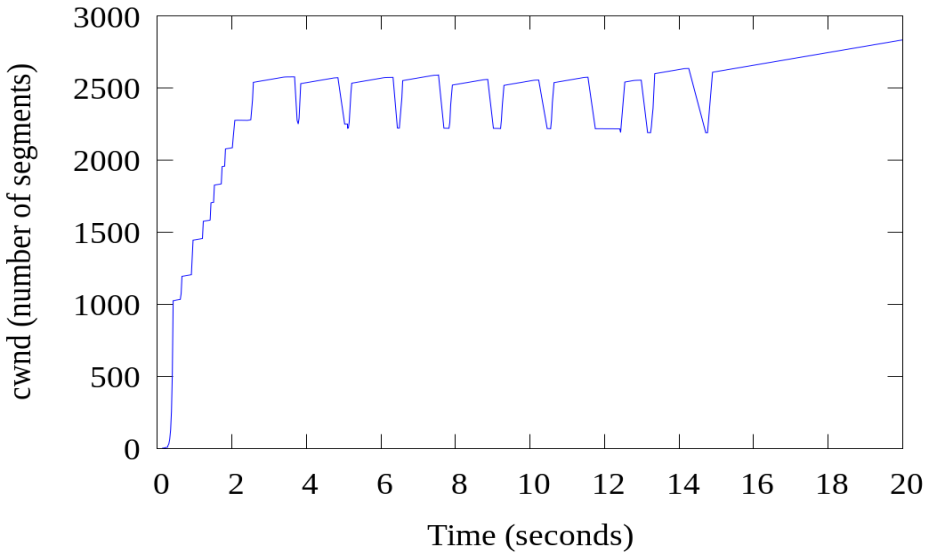

Figure 28. How FB-TCP adjust the cwnd, BER=0

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 exhausting senders' buffers and deploys the available space in intermediate buffers more efficiently. For more clarity, we can have a look on Table X to see the average values for cwnd in different BERs.

TABLE X

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

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 different BER values.

Figure 29 indicates the two protocols' throughput when BER is a small value. In contrast to HighSpeed, FB-TCP is not affected by high BERs.

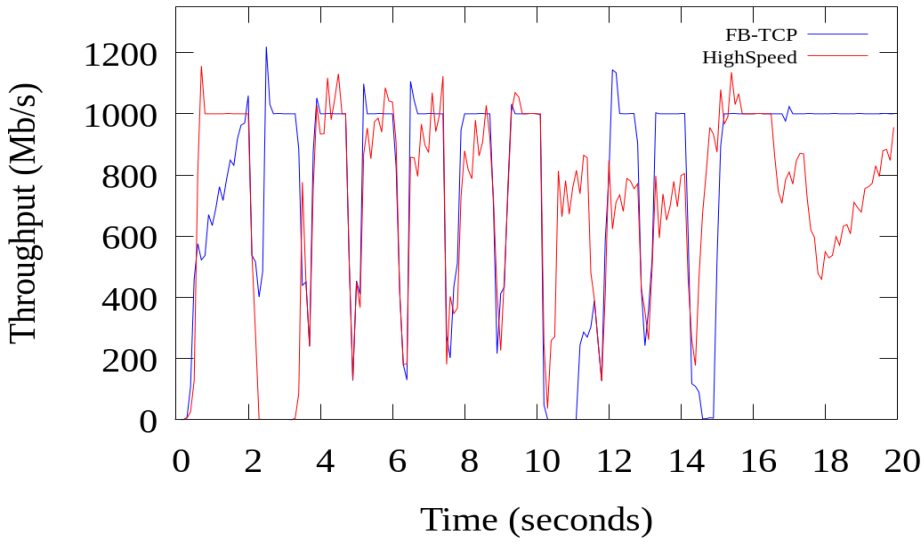

Figure 29. FB-TCP and HighSpeed throughput comparison, small BER

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 30.

This figure shows the deficiency of HighSpeed in having proper reactions to random packet drops, which leads to the underutilization of the wide available bandwidth in the network. If we increase the BER value to a large number, this insufficiency can be more obvious, as seen in Figure 31.

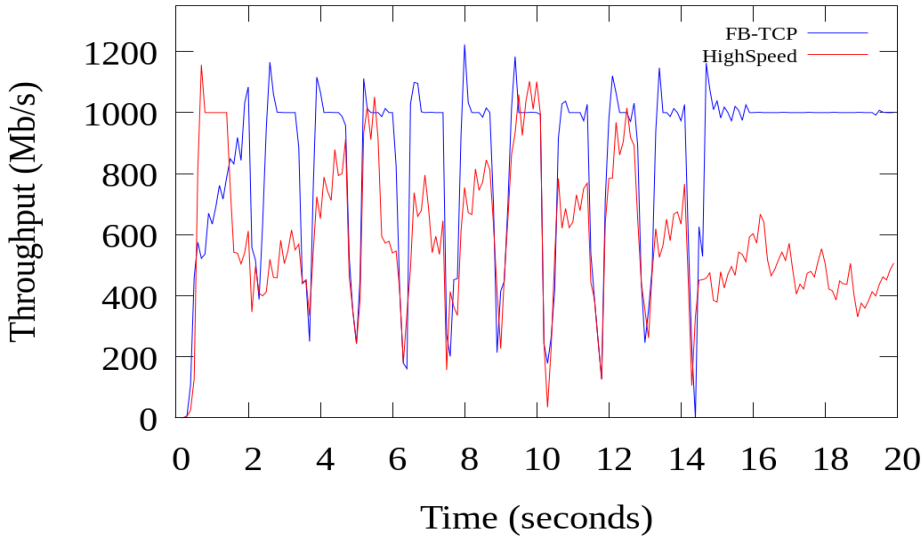

Figure 30. FB-TCP and HighSpeed throughput comparison, moderate BER

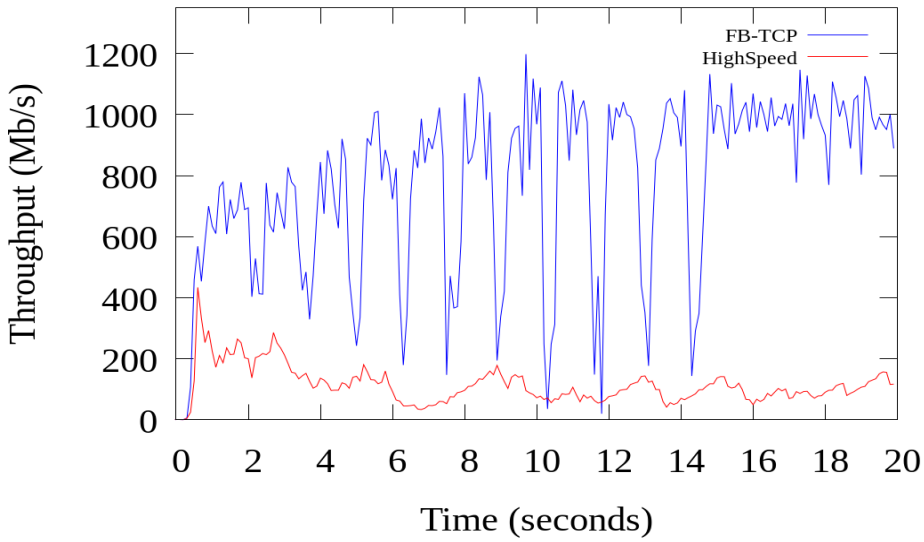

Figure 31. FB-TCP and HighSpeed throughput comparison, large BER

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

3.7 Scenario two, large obstacles

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 32 indicates the deployed layout in the second scenario.

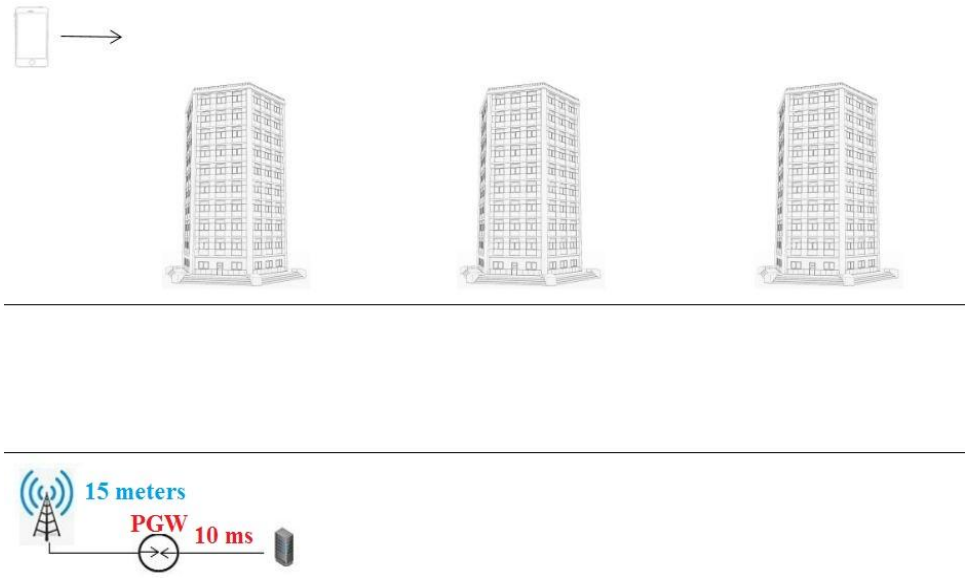

Figure 32. Scenario two

3.7.1 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 33, it can be seen that FB-TCP is the only protocol that can operate near the saturated UDP value, which equals 636.62 Mbps.

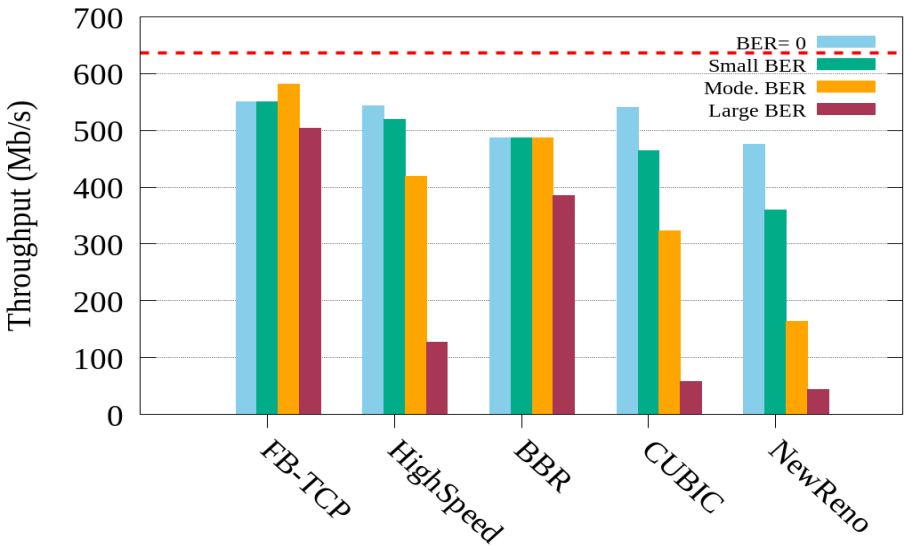

Figure 33. Average throughputs for different TCPs

Moreover, FB-TCP can retain this high functionality throughout different BERs relying on accurately analyzing the current condition of the network. Similar to scenario one, BBR shows 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 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.

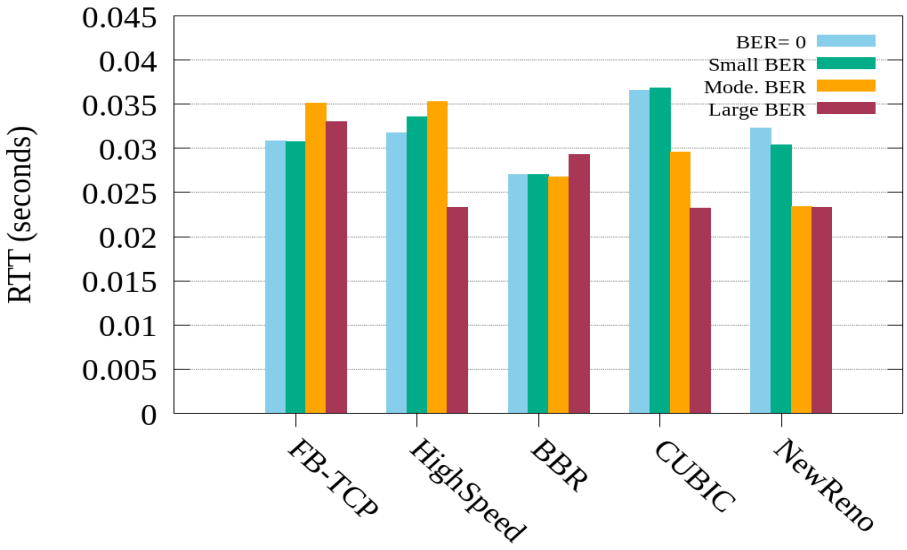

Figure 34. Average RTTs for different TCPs

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 fewer packets to the network; as a result, long queues are not established in the buffers. Figure 34 indicates average RTTs 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 35 indicates the cwnd adjustment for this protocol when there are no packet drops in the network.

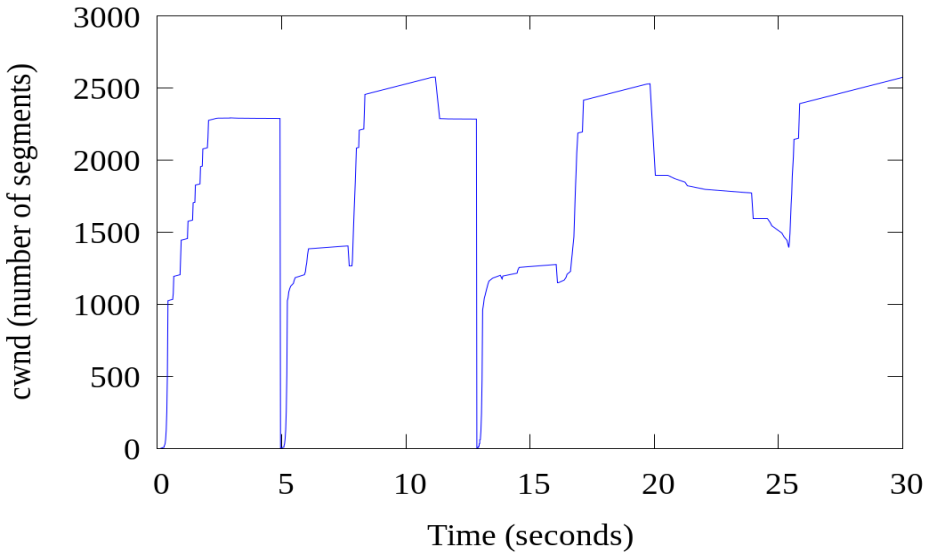

Figure 35. FB-TCP cwnd adjustment, BER=0

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, and it can recover quickly after finishing these states, then 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 35 is about the last building.

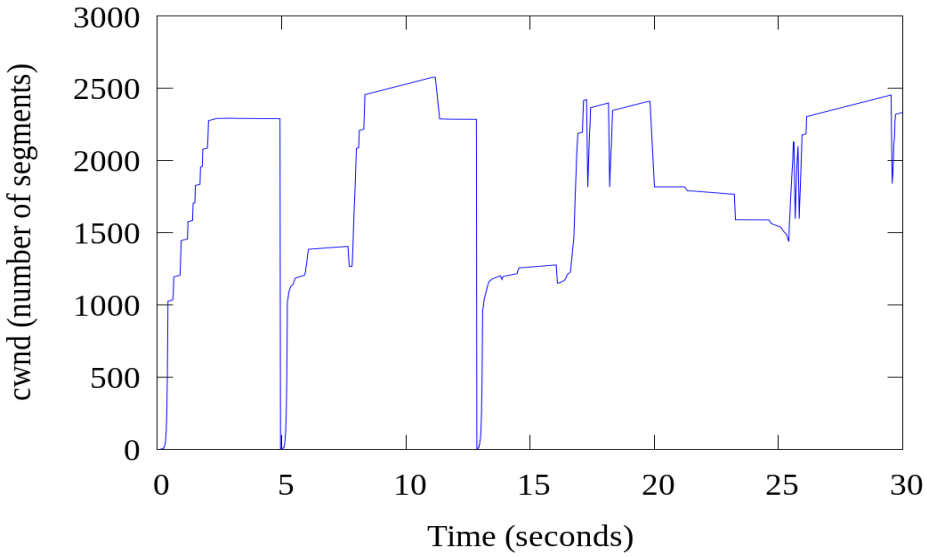

Figure 36. FB-TCP cwnd adjustment, small BER

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 36 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 37 and Figure 38.

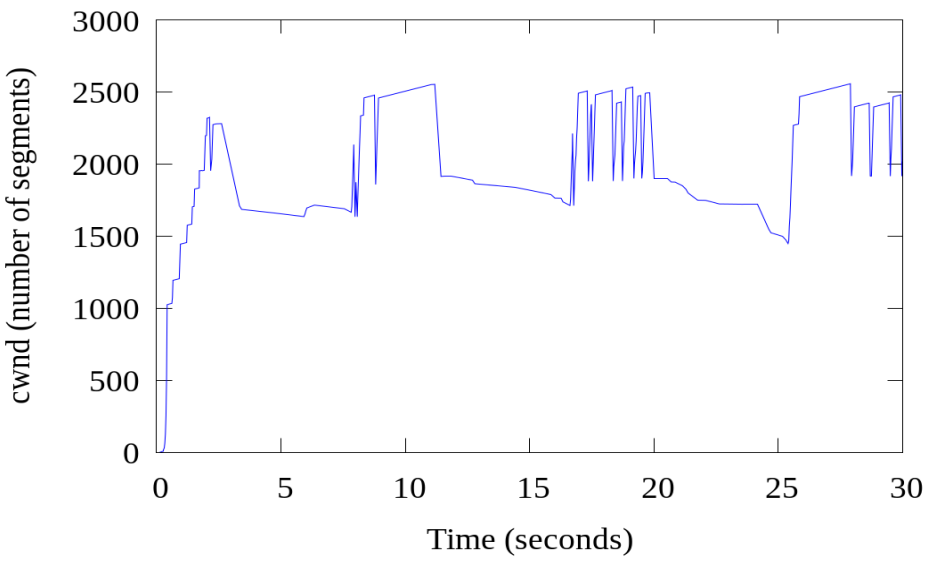

Figure 37. FB-TCP cwnd adjustment, moderate BER

The first figure shows the cwnd adjustment when BER is a moderate value. The following 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 these two figures. Figure 38 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.

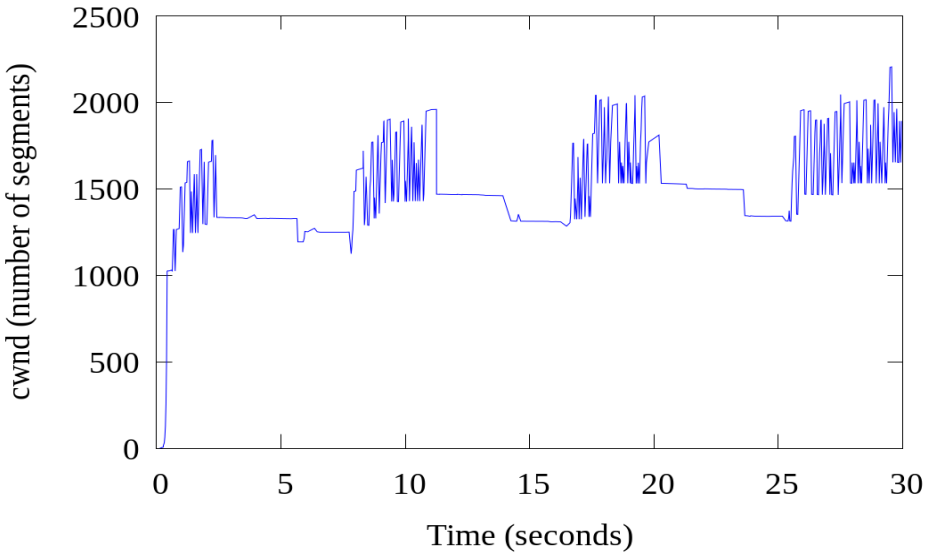

Figure 38. FB-TCP cwnd adjustment, large BER

3.8 Scenario three, statistic NLoS states added

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 drastically impair TCP functionality over 5G mmWave networks. The primary purpose is to emulate a realistic situation inside a city. The simulation time is fifty seconds.

3.8.1 Simulation results for scenario three

The obtained results showed that, like the previous scenarios, FB-TCP could outperform other TCPs. By looking at Figure 39, we can figure out that the new protocol has more efficient performance in terms of throughput and works 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 attaining higher throughputs. The average RTT for different TCPs can be seen in Figure 40.

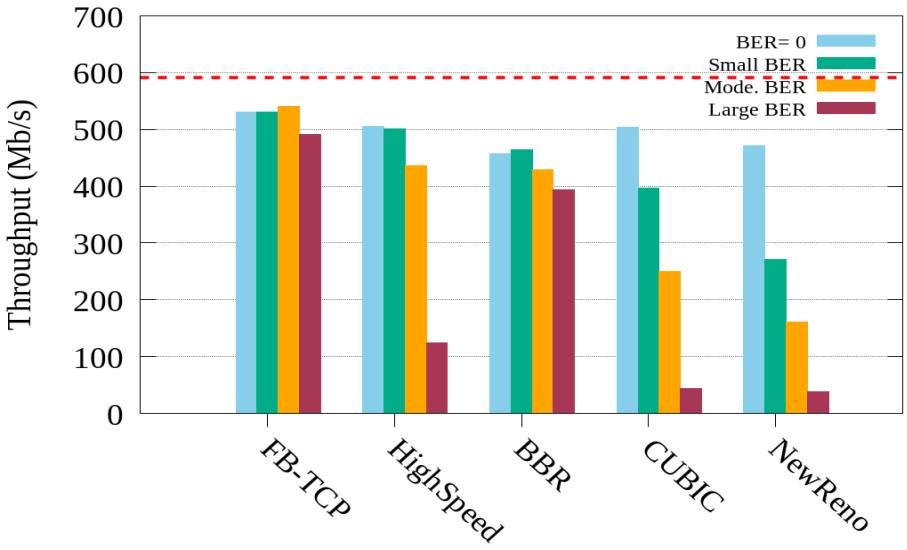

Figure 39. Average throughputs for different TCPs

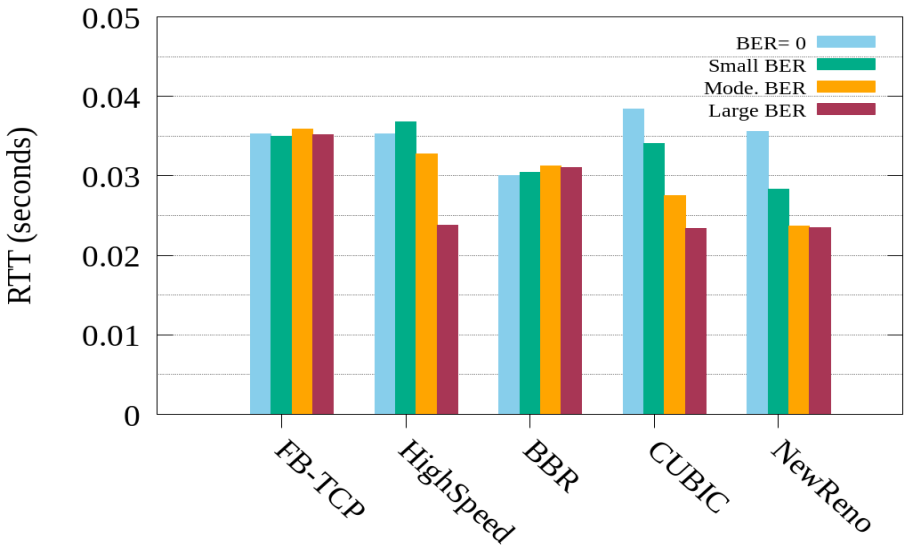

Figure 40. Average RTTs for different TCPs

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 41.

By increasing BER, FB-TCP can have 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.

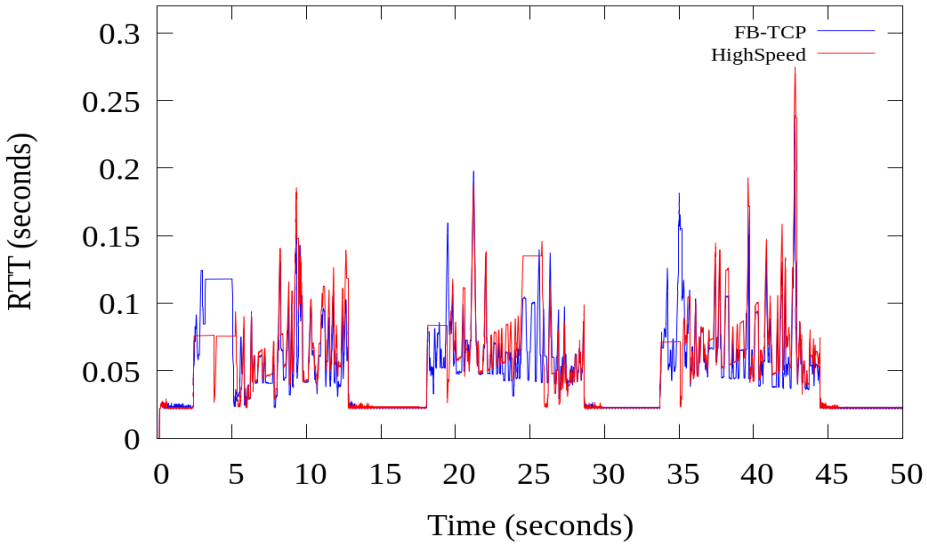

Figure 41. FB-TCP and HighSpeed RTT comparison, BER=0

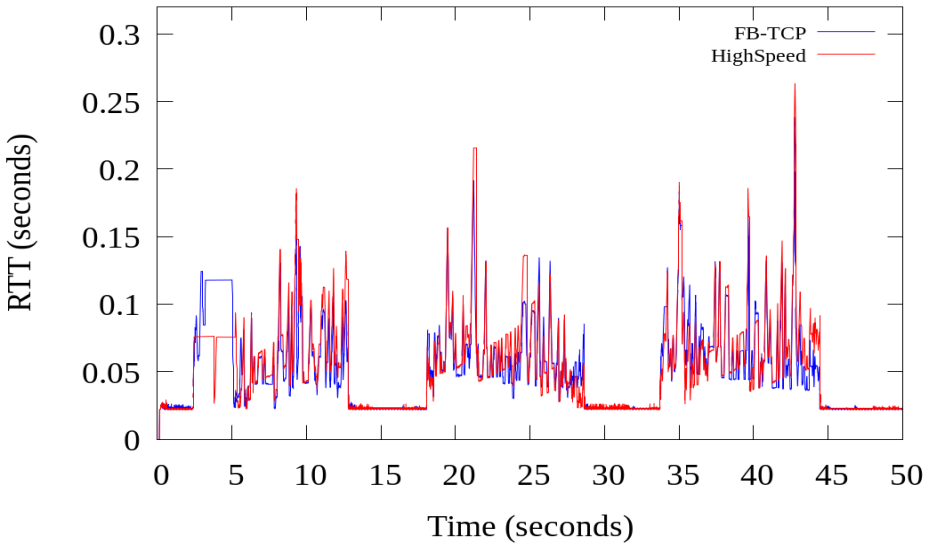

Figure 42. FB-TCP and HighSpeed RTT comparison, small BER

Figure 42 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 worth comparing.

3.9 Scenario four, all in one

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 function under very lossy conditions.

3.9.1 Simulation results for scenario four

Like the previous scenarios, FB-TCP outperforms other TCPs in terms of throughput, as shown in Figure 43. FB-TCP is the only protocol that can function close to the saturated values in all conditions.

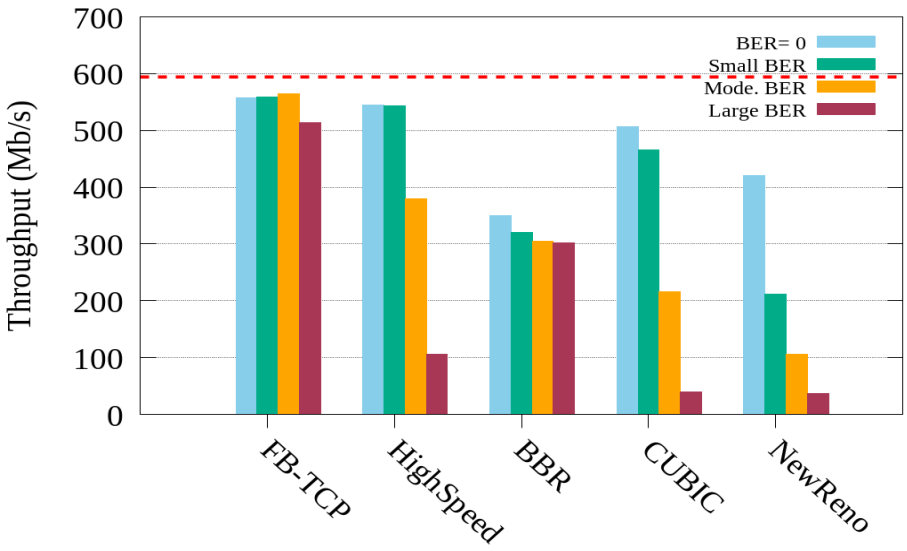

Figure 43. Average throughputs for different TCPs

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 44. 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.

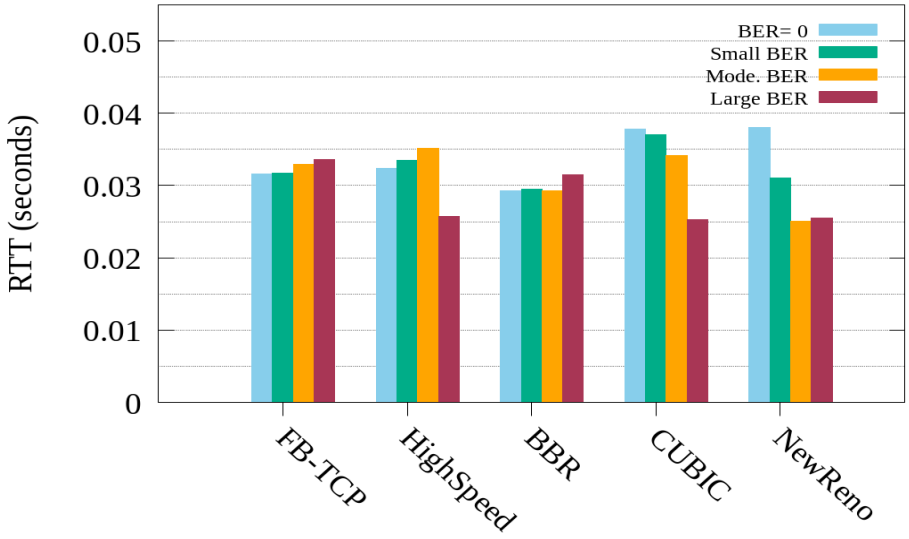

Figure 44. Average RTTs for different TCPs

Moreover, we analyzed all TCPs in very lossy environments, i.e., BER 1.25e-7 and 1.25e-6, to see how different protocols function in very large and extremely lossy environments. As Figure 45 indicates, the only TCP that can have 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 other TCPs performance in lossy environments was also proved in [79].

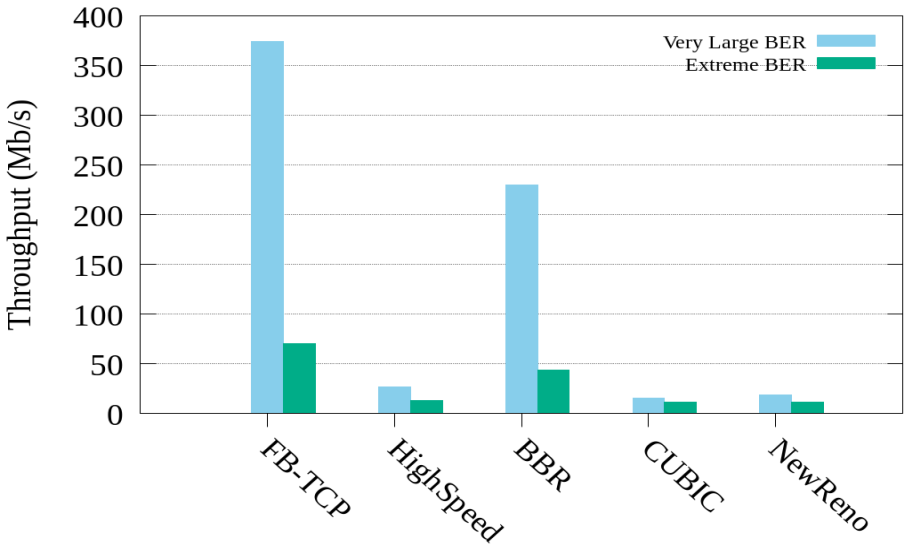

Figure 45. Average throughputs for different TCPs

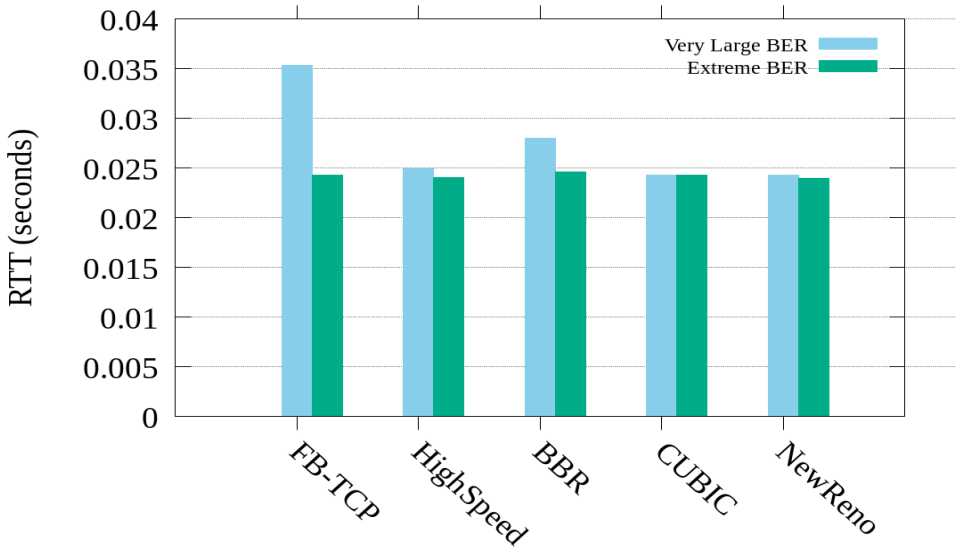

Figure 46. Average RTTs for different TCPs

The interesting part of higher throughput for FB-TCP is that it can achieve this value through acceptable RTTs, as seen in Figure 46.

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.

3.10 Conclusions

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 adequately adjusting its sending rate, leading to throughput degradation, RTT increment, and cwnd fluctuation. This section 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 urban deployments.

FB-TCP is a novel protocol that could enhance the functionality of TCP over 5G mmWave networks in urban deployments. It could function properly in all aspects. However, we can improve it one more level by adding more smartness to the protocol by exploiting the deep learning technique. The next section presents a new protocol that is using machine learning techniques and has some superiorities to FB-TCP.

4 DEEP LEARNING TCP FOR IMPROVING MMWAVE 5G PERFORMANCE DUE TO NLOS IMPAIRMENTS

As it was discussed, the new millimeter-wave frequency band can provide high data rates for the new generations. However, it suffers from a shortcoming called non-line of sight, which occurs when an obstacle between a user and an antenna makes it hard to communicate properly. The intense impact of this flaw is on TCP performance by forcing some undesirable packet drops as the protocol cannot differentiate various states in a network. This section presents another novel TCP based on deep learning that can overcome the existing defects. FB-TCP, which was proposed in the previous section, is a sufficient protocol that can enhance the performance of 5G networks. However, if we could add dynamicity to the nature of the protocol and train it based on the data from urban deployment scenarios, it would be able to work more accurately. As a result, in this section, we intend to deploy deep learning to foster the protocol's features.

4.1 DEEP learning-based TCP

Deep learning is one of the state-of-the-art techniques that can be employed to solve complicated problems. A DNN (Deep Neural Network) is a network established using neurons as inputs, outputs, and hidden layers, i.e., the neurons between inputs and outputs. The network mechanism is to solve problems through forward and backward propagation with the help of the neurons and calculations done in these neurons. DNNs are the evolved and improved version of ANN (Artificial Neural Networks), so they can employ more layers to achieve higher accuracy. The neurons in individual layers receive the previous layer's output, then, using nonlinear functions, i.e., activation functions, calculate the new values and feed them to the next layer, this procedure is called forward propagation. Different nonlinear functions such as Sigmoid, Relu, and Tanh

can be deployed based on various factors such as the training set [108]. Figure 47 shows a DNN architecture and employed parameters in our training. Later in the document, we will explain the architecture in detail.

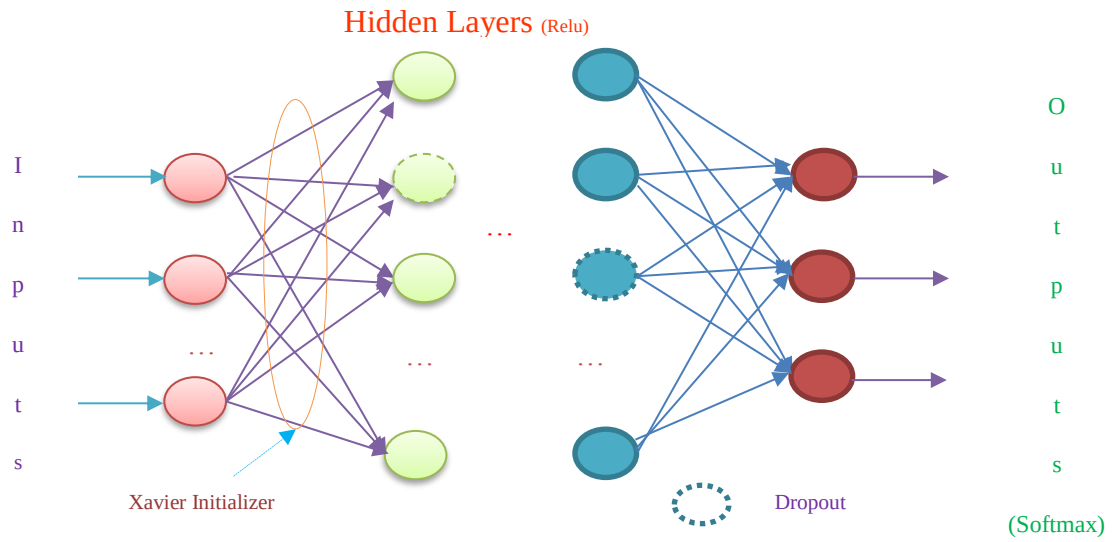

Figure 47. The architecture of a Deep Neural Network

4.2 DB-TCP architecture

DB-TCP (Deep learning-Based TCP) employs a network of five inputs as the features, three hidden layers, and three outputs. The first hidden layer consists of twenty neurons, the second one twenty-five, and the third one twenty. The number of neurons in hidden layers has been selected based on trial and error mechanism, so when the accuracy was high enough and loss became low, the procedure was stopped. The first input is the current RTT, which is the RTT at the moment. The subsequent inputs are CSI, CAD, Diff, and Average. We have borrowed CSI, CAD, and Diff from our previous protocol, FB-TCP [109], where a thorough explanation for them can be found in the previous chapter. Finally, the last input is the average of the most recent three RTTs. The reason for choosing these inputs is that they can provide a clear insight from the network and help the protocol distinguish NLoS from LoS states. The three outputs of the DNN for specifying the current state of the network are LoS, DNLoS (Dynamic NLoS), and SNLoS (Static NLoS). LoS is when there are no obstacles between the user and the antenna. DNLoS and SNLoS refer to the times that some obstacles act as hurdles on the way of establishing a proper connection. In the former one, the user is moving, but in the latter one, it is still.

The inputs for training the model have been taken from the simulation results of the training scenario, seen in Figure 48. It means that we collected the values for the simulation when it was running. Then we employed these values in order to feed the deep neural network. In the future, this training set can be tremendously increased in order to have a generalized protocol. Moreover, other training algorithms can be exploited to change the behavior of the protocol as the protocol dynamicity let changing different parts of it.

There are five trees and three buildings in this scenario that act as obstacles to impair communication and create NLoS states. The primary reason for choosing this scenario is that it includes most of the conditions in urban deployments, so the protocol can function well in cases with fewer flaws. During training the model, we have used callbacks to stop the training when the accuracy is above 0.9985; as a result, the training has stopped in 987 epochs, and the corresponding loss was 0.0045. Both accuracy and loss are close to the optimal values, which are one for the accuracy and zero for the loss.

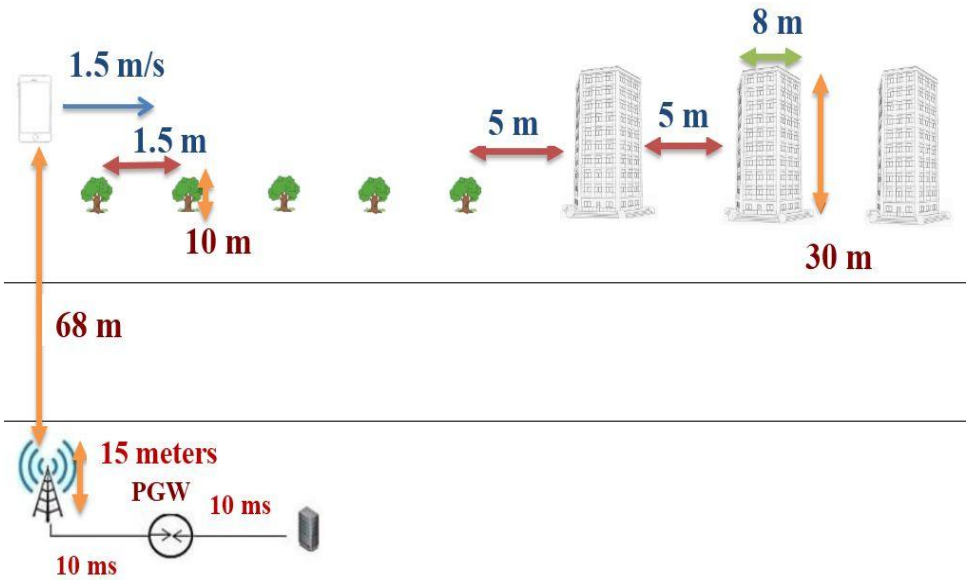

Figure 48. The training topology

In the next step, for the evaluation process, we have tested the trained model on the created data from another scenario called evaluation. The main aim of this procedure is to figure out how the trained engine functions on inputs that it has not seen before. The result for accuracy and loss was 0.9940 and 0.0367, which are desirable for a trained model.

The employed activation function in the DNN is Relu (Rectified Linear Activation Function), one of the popular activation functions. To prevent overfitting, Dropout [110] has been used. For establishing the initial weights Xavier Initializer [111] has been chosen. Finally, for the calculation of the backward propagation, Adam optimizer [112] has been exploited to lower the loss function. Eventually, Softmax was employed at the output layer to classify the network into three different clusters.

TABLE XI

HOW DB-TCP ADJUSTS THE SENDING RATE

DB-TCP	CWND ADJUSTMENT
(LOS) && (CSI >= 0.99) && (DIFF == 0)	CWND=CWND + CWND/10
(LOS) && (CSI >= 0.99) && (DIFF > 0) && (DIFF <= 2)	CWND=CWND + CWND/100
(LOS) && (CSI >= 0.99) && (DIFF > 2)	CWND=CWND + CWND/1000
(LOS) && (CSI < 0.99)&& (CSI > 0.8) && (DIFF < 2)	CWND=CWND + CWND/200
(LOS) && (CSI < 0.99)&& (CSI > 0.8) && (DIFF > 2)	CWND=CWND
(LOS) && (CSI < 0.8) && (DIFF < 2)	CWND=CWND-CWND/100
(LOS) && (CSI < 0.8) && (DIFF > 2)	CWND=CWND-CWND/50
(DNLOS) && (CSI >= 0.7)	CWND=CWND- CWND/20
(DNLOS) && (CSI < 0.7) && (CSI >= 0.3)	CWND=CWND - CWND/10
(DNLOS) && (CSI < 0.3)	CWND=CWND - CWND/5
(SNLOS) && (CSI >= 0.6)	CWND=CWND - CWND/5

(SNLoS) && (CSI < 0.6)	CWND=CWND/2

After determining the state of the network by the output of the DNN, DB-TCP adjusts the cwnd based on CSI and Diff, where the first one specifying the aggressive of the protocol and the second one is for controlling the steps and figuring out the upper bound of the network. In the next step, for the evaluation process, we have tested the trained model on the created data from another scenario called evaluation. The main aim of this procedure is to figure out how the trained engine functions on inputs that it has not seen before. The result for accuracy and loss was 0.9940 and 0.0367, which are desirable for a trained model.

The employed activation function in the DNN is Relu (Rectified Linear Activation Function), one of the popular activation functions. To prevent overfitting, Dropout [110] has been used. For establishing the initial weights Xavier Initializer [111] has been chosen. Finally, for the calculation of the backward propagation, Adam optimizer [112] has been exploited to lower the loss function. Eventually, Softmax was employed at the output layer to classify the network into three different clusters.

Table XI indicates how DB-TCP adjusts the cwnd. These parameters are tunable, and other values can be tested in future researches to adapt the protocol to different scenarios and circumstances. In the next step, for the evaluation process, we have tested the trained model on the created data from another scenario called evaluation. The main aim of this procedure is to figure out how the trained engine functions on inputs that it has not seen before. The result for accuracy and loss was 0.9940 and 0.0367, which are desirable for a trained model.

The employed activation function in the DNN is Relu (Rectified Linear Activation Function), one of the popular activation functions. To prevent overfitting, Dropout [110] has been used. For establishing the initial weights Xavier Initializer [111] has been chosen. Finally, for the calculation of the backward propagation, Adam optimizer [112] has been exploited to lower the loss function. Eventually, Softmax was employed at the output layer to classify the network into three different clusters.

Table XI has been obtained after numerous simulations to be best fitted for the urban deployment. The main reason behind the selection of the parameters is to achieve high throughput

through low RTTs, which can happen by aggressive approaches in LoS states and non-aggressive ones in NLoS states.

4.3 Simulation results

In this section, we compare DB-TCP with the most used TCPs (NewReno, Cubic, HighSpeed, and BBR). Later in the document, we will compare it with the previously designed and evaluated protocol (FB-TCP). As shown in Figure 49, DB-TCP can operate close to the UDP saturated value, which is 604.99 Mb/s, and has better functionality than other TCPs, especially when there are random packet drops in the network.

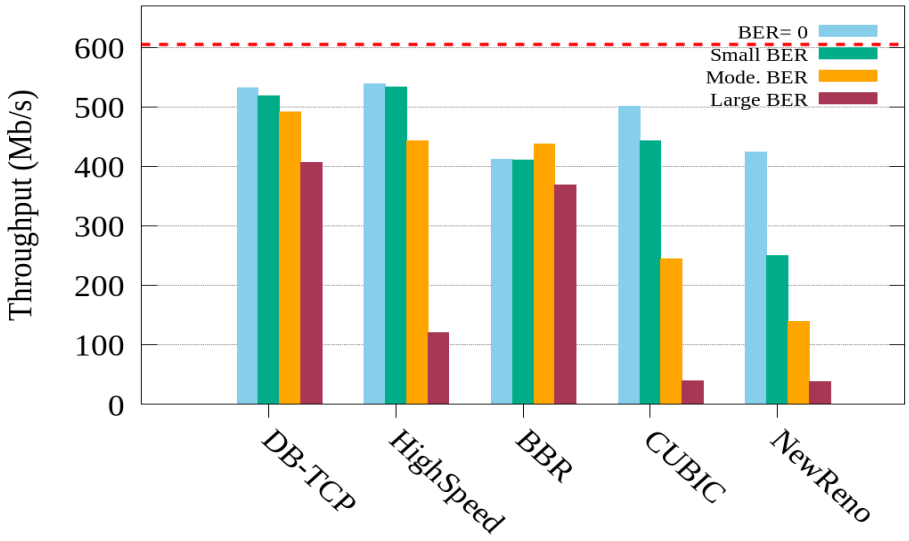

Figure 49. Average throughputs for different TCPs in the training scenario

A similar conclusion can be drawn from the RTT point of view, as shown in Figure 50. We can see that DB-TCP has better performance while maintaining good RTT values.

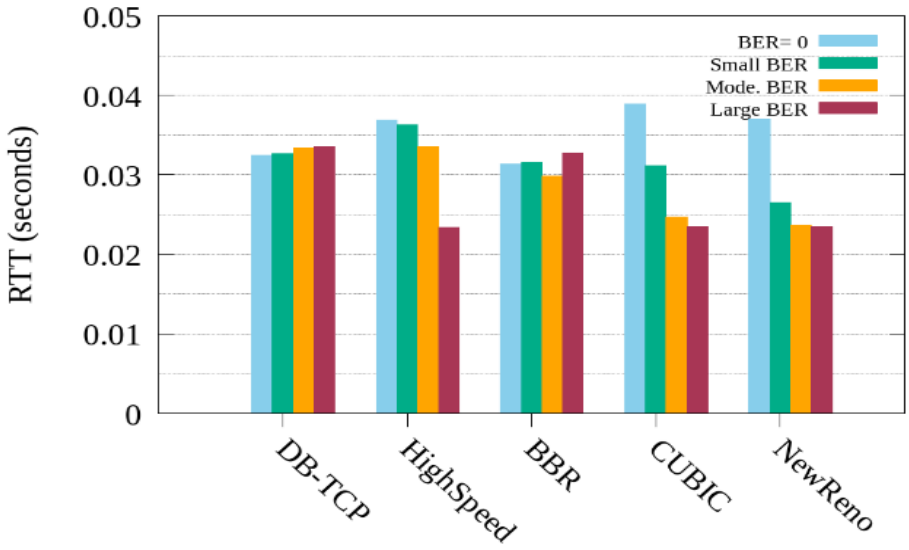

Figure 50. Average RTTs for different TCPs in the training scenario

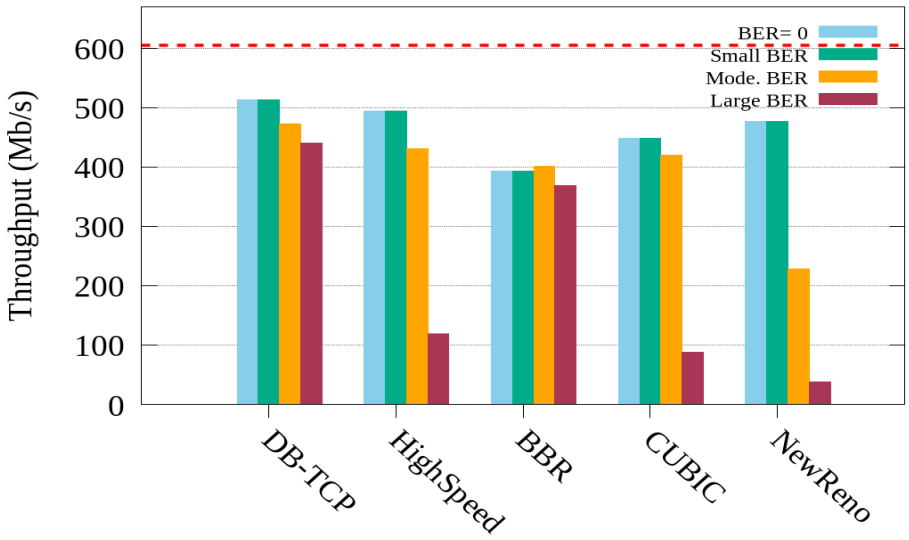

Figure 51. Average throughputs for different TCPs in the evaluation scenario

We compared the protocol with other TCPs in the evaluation topology to analyze DB-TCP on data that it has not seen before.

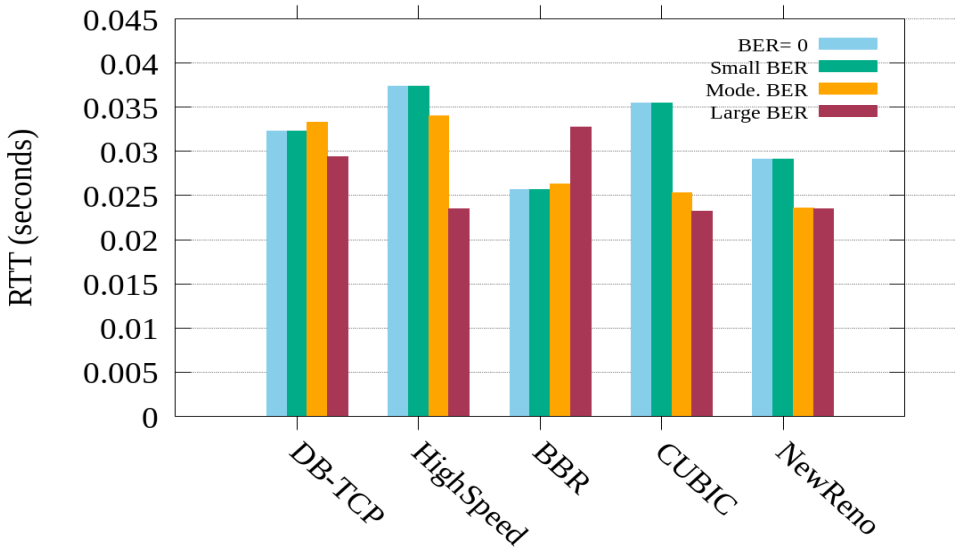

Figure 52. Average RTTs for different TCPs in the evaluation scenario

As shown in Figure 51 and Figure 52, DB-TCP has better functionality than other TCPs. This superiority is because of its vision from the network's conditions and precisely adjusting the cwnd, as shown in Figure 53.

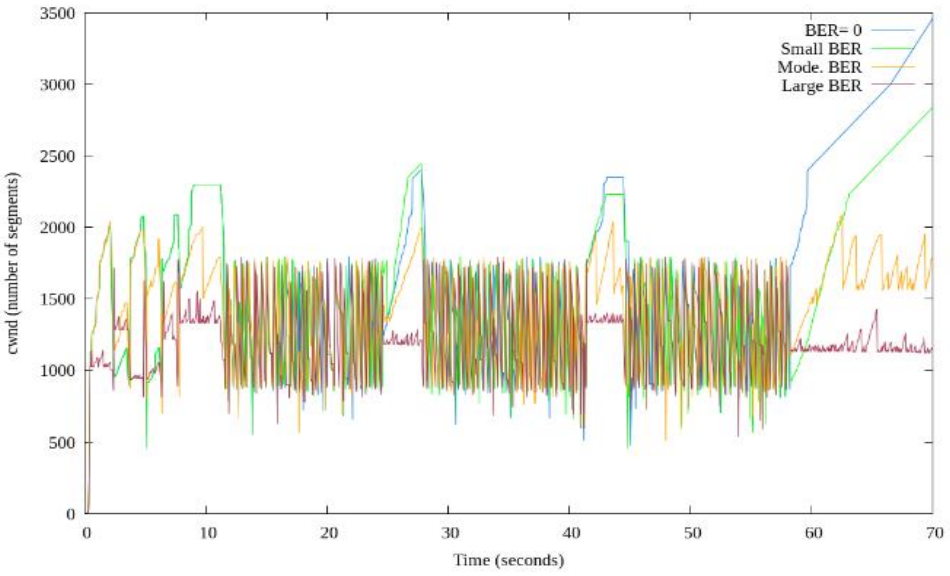

Figure 53. DB-TCP cwnd adjustment in the training scenario

Besides elaborate adjustment, DB-TCP can achieve high performance through a small average cwnd size, which shows its high accuracy in controlling the sending rate and not exhausting the buffers. Table XII is a comparison of DB-TCP and HighSpeed in adjusting the cwnd.

TABLE XII

DB-TCP AVERAGE CWND SIZE COMPARED TO HIGHSPEED

BER	DB-TCP	HighSpeed
zero	1800	27091
Small	1629	27819
Moderate	1481	14378
High	1195	303

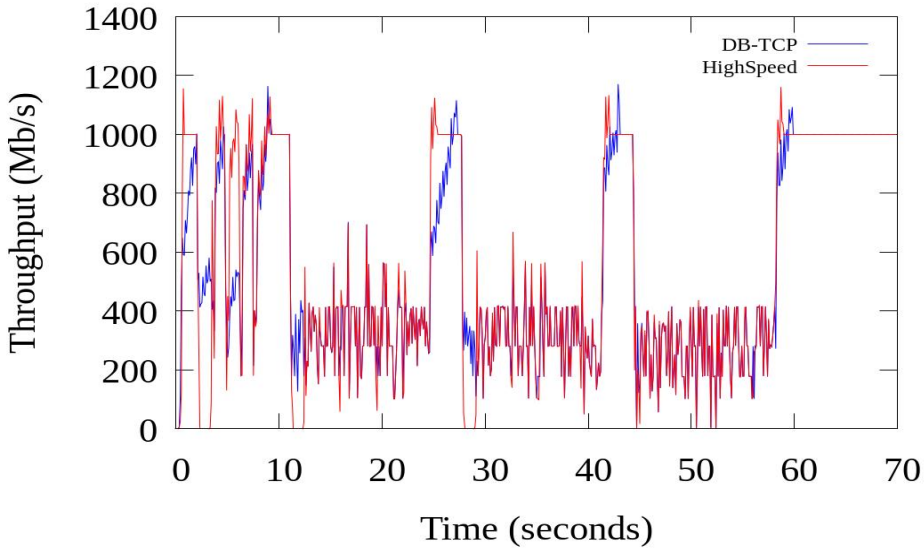

Figure 54. Throughput comparison of DB-TCP and HighSpeed, BER=0

In order to have a comprehensive insight, we can look at the instantaneous values for DB-TCP and HighSpeed as the best candidate for conventional TCPs. Figure 54 indicates the comparison of the throughputs in the absence of random packet drops. The figure reveals robust responsiveness for the new protocol. This superiority is more evident when there are random packet drops in the network, as seen in Figure 55. DB-TCP can react to different situations more promptly.

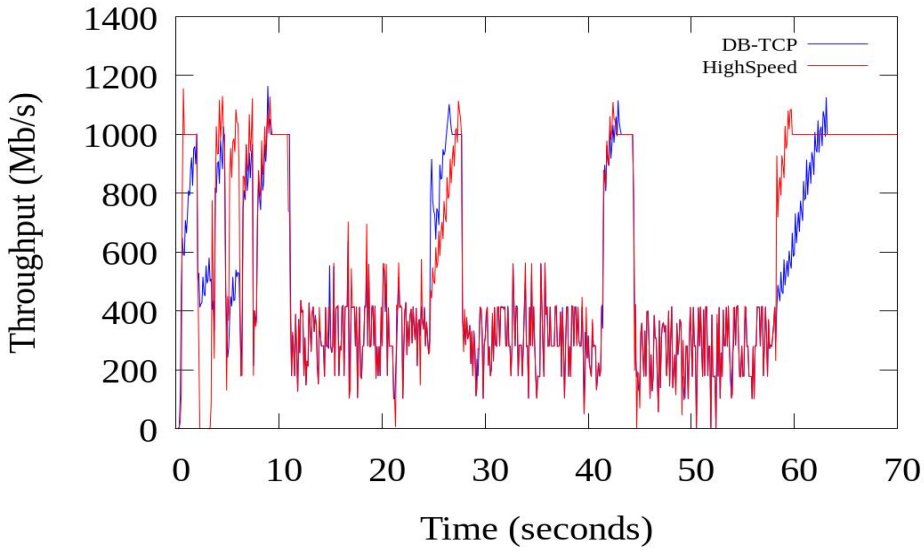

Figure 55. Throughput comparison of DB-TCP and HighSpeed, small BER

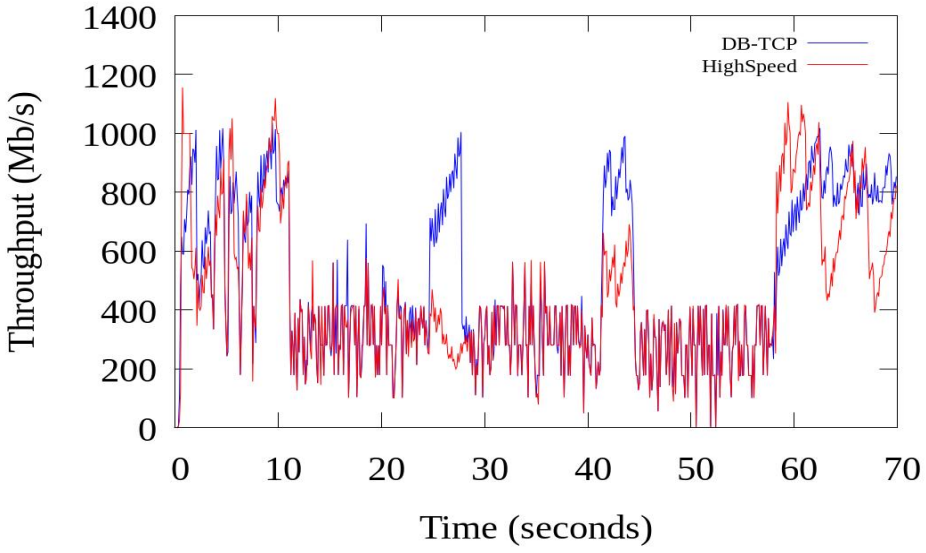

Figure 56. Throughput comparison of DB-TCP and HighSpeed, moderate BER

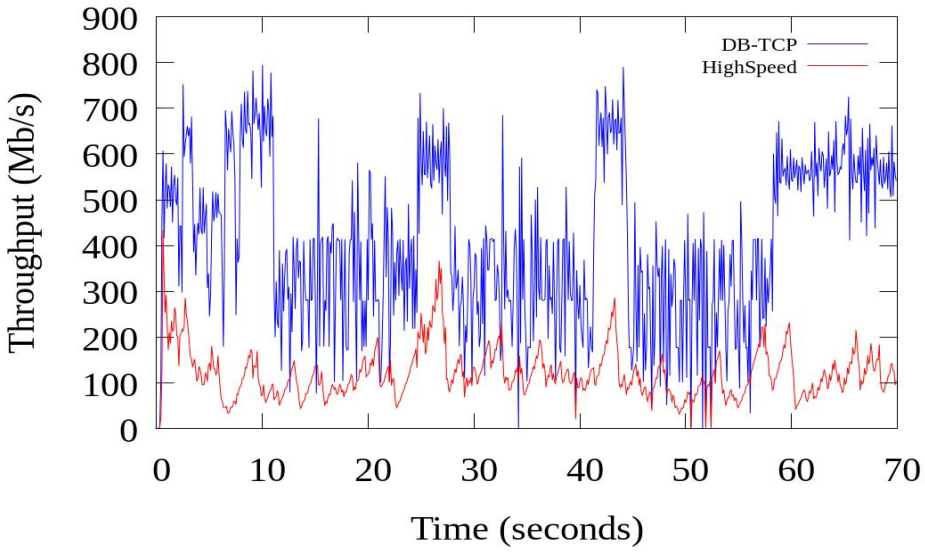

Figure 57. Throughput comparison of DB-TCP and HighSpeed, large BER

If the BER is increased to medium or high values, this supremacy is more straightforward, as seen in Figure 56 and Figure 57. The figures reveal that FB-TCP can react appropriately to different circumstances, which makes this protocol an excellent choice for urban deployments.

4.4 FB-TCP or DB-TCP, which one is the best choice

In this thesis, we have introduced two protocols, one based on fuzzy and the other based on deep learning. Now it is time to get to the conclusion that which of the protocols has superiorities compared to the other one; as a result, we are going to compare some KPIs of them. For this, we have deployed FB-TCP in both training and evaluation scenarios of DB-TCP, i.e., scenario one and scenario two, respectively. As it is shown in Figure 58, both protocols have close functionalities, especially when BER is increasing. Looking at the low BERs reveals that DB-TCP has a better performance compared to FB-TCP. This is because this protocol can better understand the network based on its deep learning approach and reacts to different situations precisely.

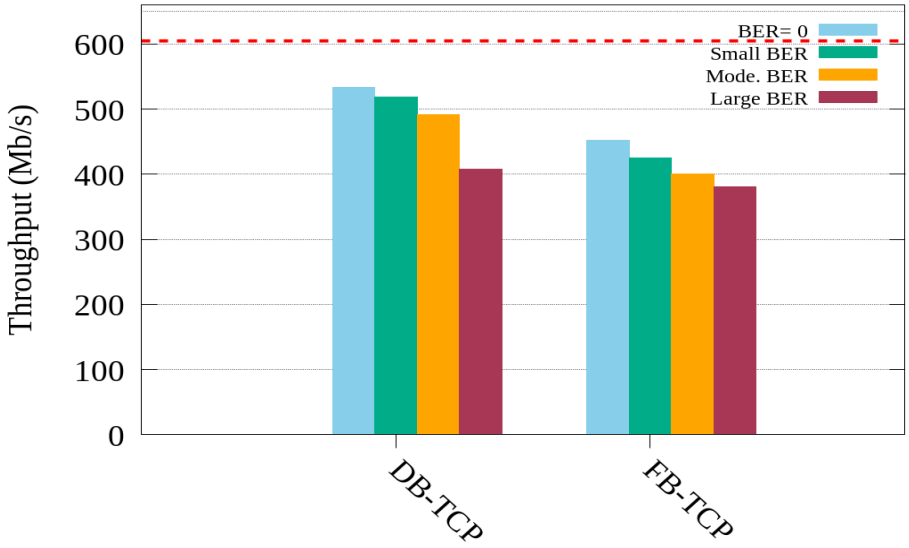

Figure 58. Average throughputs for DB-TCP and FB-TCP in scenario one

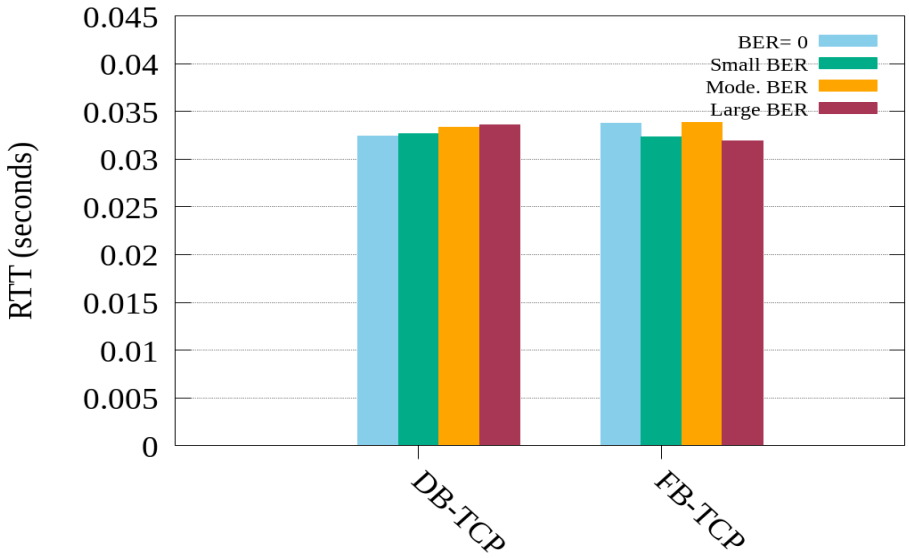

Figure 59. Average RTTs for DB-TCP and FB-TCP in scenario one

This superiority can also be gained in terms of RTT, as seen in Figure 59. Except for the lossy environment, in which FB-TCP has a better RTT, DB-TCP can have negligible improvements in other circumstances. However, in the case of RTT, both protocols can function close to each other.

Comparing cwnd adjustment for both protocols can give us a clear understanding of how they react to various situations. As a result, we have analyzed their behavior in all BERs. Figure 60

shows the cwnd adjustment for both protocols with no random packet drop in the network. In this case, FB-TCP can have stable functionality all the time; however, DB-TCP reacts properly and recovers quickly from adverse situations, which is its principal superiority. These quick reactions can be seen when NLoS states are finished. Each NLoS state happens when the UE is behind an obstacle. These states are more clear when the UE is behind a big obstacle like a building, as we have three of them in the figure causing degradation on the performance of the protocols. The impacts of the trees can be seen at the beginning of the figure too.

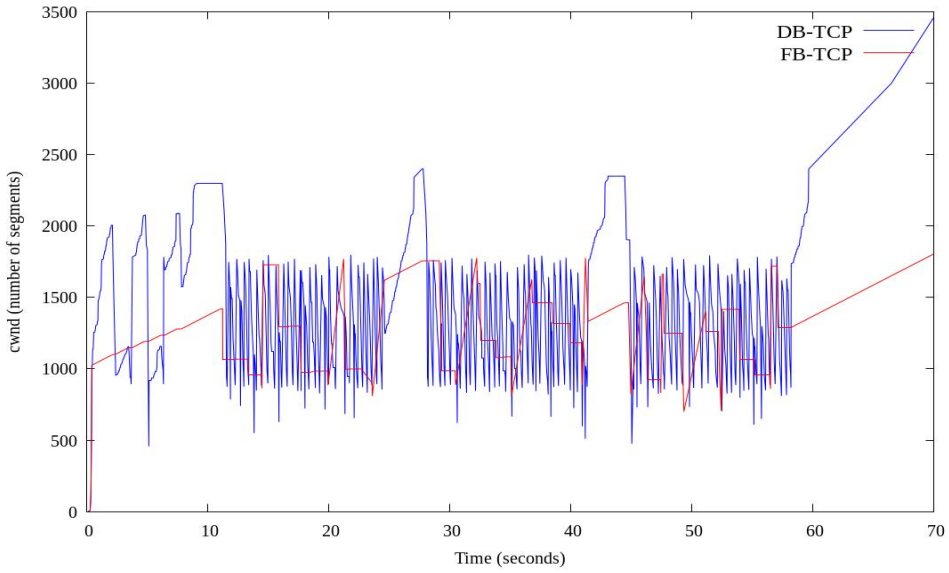

Figure 60. cwnd adjustment comparison of DB-TCP and FB-TCP, BER=0

Looking at Figure 61 and Figure 62 indicate that small and moderate BERs cannot affect the functionality of protocols because of their non-loss-based nature. However, by having moderate random packet drops, cwnd adjustments are affected minorly, which because of that, both protocols experience a paltry reduction in their performances. Figure 63 justifies this claim and shows that even large BERs cannot mislead these protocols in adjusting the cwnd in contrast to conventional TCPs.

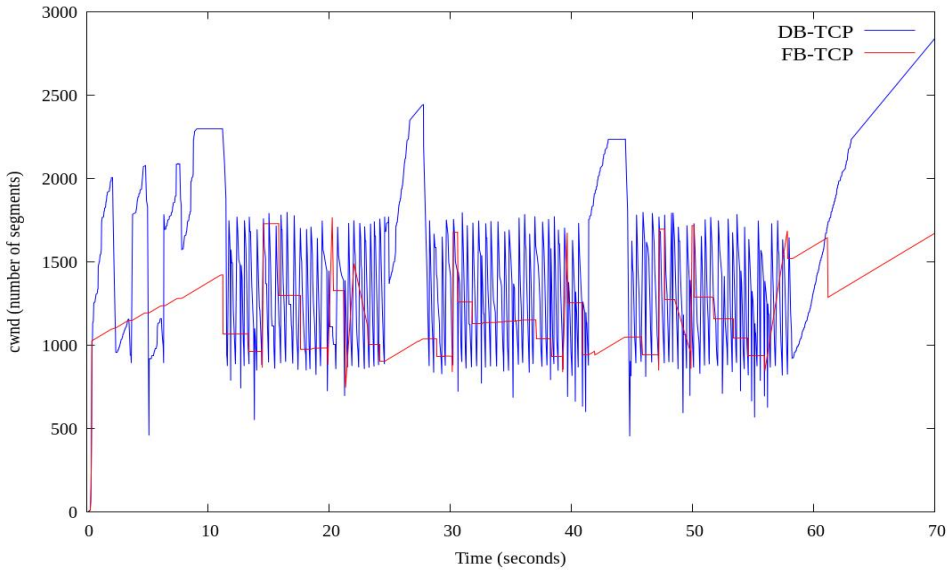

Figure 61. cwnd adjustment comparison of DB-TCP and FB-TCP, small BER

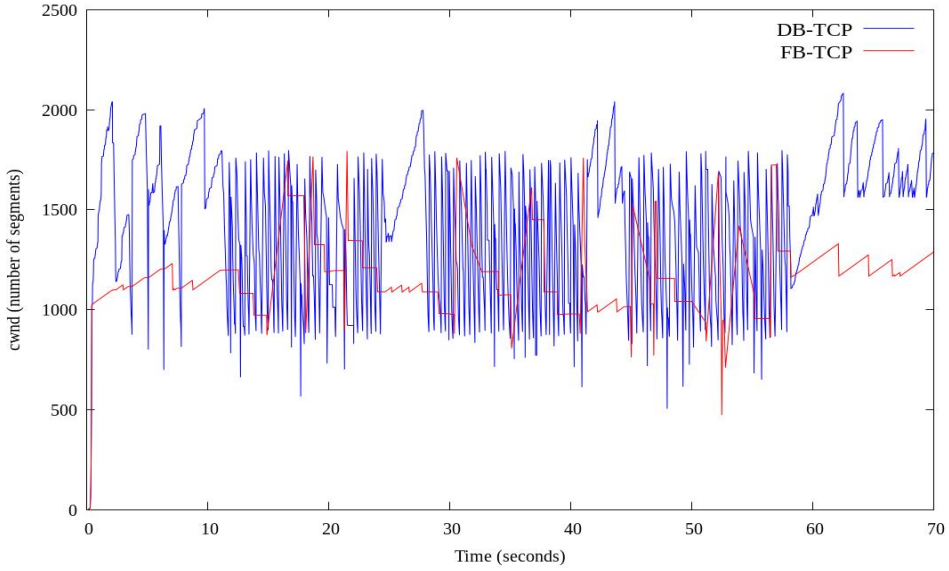

Figure 62. cwnd adjustment comparison of DB-TCP and FB-TCP, moderate BER

As discussed in the previous chapters, long NLoS states can affect the functionality of the protocols severely. However, DB-TCP and FB-TCP can distinguish different situations more clearly and reveal the adverse impacts. The reason behind this superiority is that these protocols rely on their intelligent congestion control mechanism and try to avoid blind decisions that unwanted situations may force.

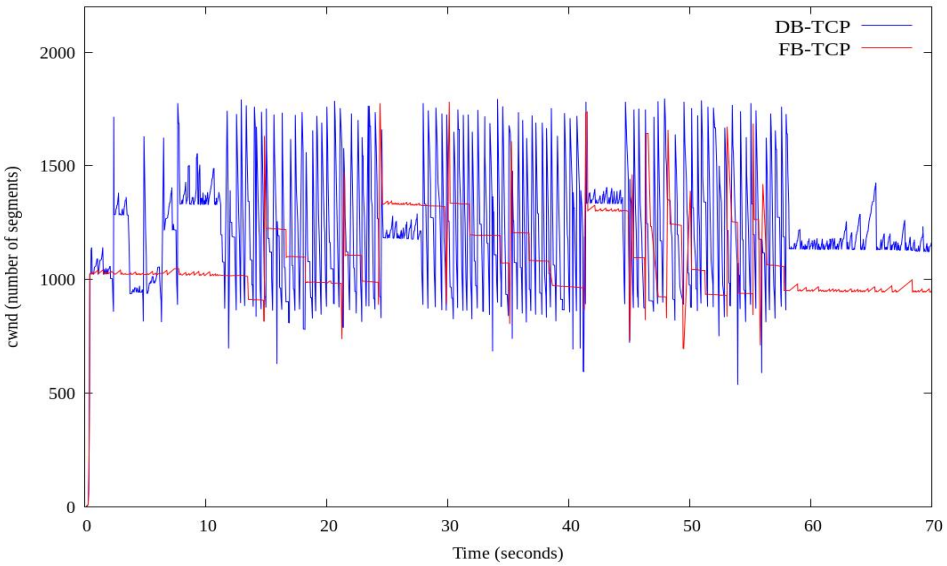

Figure 63. cwnd adjustment comparison of DB-TCP and FB-TCP, large BER

For more clarity, we can compare average cwnd sizes for these protocols as seen in Table XIII. Both protocols can achieve high performances through low values, which is a significant upside in preventing buffer exhaustion. DB-TCP can attain higher averages cwnd, which is its key capability in having larger throughputs than FB-TCP.

TABLE XIII

DB-TCP AVERAGE CWND SIZE COMPARED TO FB-TCP

BER	DB-TCP	FB-TCP
zero	1800	1401
Small	1629	1264
Moderate	1481	1144
High	1195	1061

From the RTT point of view, both protocols have the same functionality with some slight changes in various situations, as seen in Table XIV.

TABLE XIV

AVERAGE RTTS FOR DB-TCP AND FB-TCP IN SECONDS

BER	DB-TCP	FB-TCP
zero	0.032422	0.033753
Small	0.032685	0.032342
Moderate	0.033330	0.033807
High	0.033607	0.031941

After comparing the protocols in the first scenario, i.e., training, we have employed FB-TCP in the second scenario, i.e., evaluation, to have in-depth information from both protocols' functionalities. Table XV summarizes the obtained results for the most important KPIs. In all BERs, DB-TCP has higher throughputs than FB-TCP. However, FB-TCP, because of its reduced throughputs compared to DB-TCP, can achieve lower RTTs. Furthermore, both protocols can function around small average cwnd sizes, which is a positive feature.

TABLE XV

DB-TCP AND FB-TCP AVERAGE VALUES COMPARISON IN THE SECOND SCENARIO

BER	DB-TCP	FB-TCP
zero	Throughput:512/76 Mb/s	Throughput:450/67 Mb/s
	RTT: 0.032340 s	RTT: 0.028273 s
	cwnd: 1466	cwnd: 1150
Small	Throughput:512/76 Mb/s	Throughput: 450/67 Mb/s
	RTT: 0.032340 s	RTT: 0.028273 s
	cwnd: 1466	cwnd: 1150
Moderate	Throughput:473.10 Mb/s	Throughput:444/13 Mb/s
	RTT: 0.033321 s	RTT: 0.028788 s
	cwnd: 1481	cwnd: 1133
High	Throughput: 439.76 Mb/s	Throughput: 402/25 Mb/s
	RTT: 0.029359 s	RTT: 0.027870
	cwnd: 1136	cwnd: 993

To sum up, both DB-TCP and FB-TCP can function adequately in urban deployments by having a clear view from the network's different conditions, such as LoS, NLoS, or random packet drops. Both can achieve high throughputs; however, DB-TCP always has the higher ones. In terms of RTT, the protocols can achieve acceptable RTTs; nonetheless, the lower RTTs for

FB-TCP in some cases can be compensated for higher throughputs of DB-TCP. Finally, Both protocols can perform by attaining a small average cwnd size, which can prevent bufferbloating. In a nutshell, it is true that both protocols are excellent choices, but DB-TCP is the most suitable protocol for urban deployments that can satisfy all the expected features of a well-performed protocol.

4.5 Conclusions

A new 5G mmWave protocol called DB-TCP was proposed in this section to overcome flaws caused by NLoS states in urban deployments. The novel protocol relies on deep learning to have a manifest insight from the network and adjust the sending rate accurately. As a result, it can achieve higher performance than other TCP variants in terms of throughput, RTT, and cwnd fluctuation. In non-lossy environments, the throughput enhancement can be negligible. However, in lossy ones, it can reach large orders. The main reason for this superiority is that DB-TCP has a tangible view from the network and can react quickly to different states.

5 CONCLUSIONS AND FUTURE WORK

This is the final chapter of the thesis. In this chapter, we present a summary of the conclusions of the thesis and future work that can be done.

5.1 Conclusions

The different layers of the protocol stack, mainly the widely used transport protocol TCP, encounter new issues when deployed in 5G networks, especially along with higher frequencies such as mmWave. The main challenge of TCP is due to the intermittent nature of mmWave channels, which are sensitive to blockage and misalignment, especially when a UE and a gNB cannot establish a LoS connection. These problems cause fluctuations in the functionality of the congestion control mechanisms of different TCP variants by having LoS<->NLoS transitions, which lead to degradation of the measuring factors, including throughput, latency, cwnd adjustment, and fairness. All these KPIs are degraded due to the mentioned issues and need to be tackled in order to utilize 5G mmWave to its full potential. The main challenge on the way of attaining the highest possible performance is the misleading impacts of the blockage and the non-intelligent mechanisms of congestion control algorithms in various TCPs, which are not able to distinguish different statuses such as LoS, NLoS, or congestion in a network. We have done a thorough review of 5G technology and its various aspects, a full investigation of TCP and its functionality over 5G networks, and the analysis of if it is better to replace this protocol with novel ones in the coming future or adapting it. Moreover, we analyzed the behavior of TCP and 5G mmWave when exploited in urban deployments in-depth. The results indicated that the existing issues, such as blockage and random packet losses, could degrade the network's performance as TCP encounters difficulties reaching acceptable functionality.

To sum up, the 5G network is a promising telecommunication infrastructure that will revolute various aspects of communication. However, different parts of the Internet, such as its regulations and protocol stack, will face new challenges, which need to be solved in order to exploit 5G capacity, and without intelligent rules and protocols, the high bandwidth of 5G, especially 5G

mmWave will be wasted. Two novel schemes to solve the issues have been proposed based on an Artificial Intelligence subset technique called fuzzy and a machine learning-based approach called Deep learning to enhance the performance of 5G mmWave by improving the functionality of the transport layer. The obtained results indicated that the new schemes could improve the functionality of TCP by giving intelligence to the protocol. As the protocol works more smartly, it can make sufficient decisions on different conditions.

In this thesis, we have contributed with a comprehensive analysis of TCP and its well-known variants. Also, 5G features and capabilities were discussed in detail to discover the advantages and disadvantages of the new generations. Afterward, TCP mechanisms and parameters, which can be affected in 5G networks, were inspected. Following, the researches that are involved in improving and analyzing TCP over 5G networks were presented. Finally, the methodology has been brought to lighten the path for the new protocols.

As a way to propose an improvement of TCP over mmWave 5G, our novel protocol based on fuzzy called FB-TCP has been brought. FB-TCP strives to enhance the functionality of the transport layer in 5G mmWave networks over urban deployments. 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, including throughput, RTT, and cwnd adjustment. This protocol can utilize the 5G mmWave bandwidth to its full potential by reaching an acceptable latency.

After the first proposal of the protocol, we identified a machine learning approach to enhance TCP performance; for that, another innovative TCP based on deep learning called DB-TCP was proposed. This protocol tries to determine the current state of the network, i.e., LoS, SNLoS, and DNLoS, then decides based on the ongoing conditions. The simulation results revealed that the new protocol could achieve high performance comparing to conventional TCPs.

The principal conclusions and contributions of the thesis are as follows:

- The mmWave band is the leading candidate to be employed in new cellular communications due to the new features and capabilities that it can provide. Compared

to the previous generations, mmWave can equip the upcoming cellular generations with much higher performance.

- The transport layer of the protocol stack with its widely deployed protocol, TCP, is the primary tool in providing reliable end-to-end connection over the cellular networks. However, because of the susceptible characteristic of the mmWave to obstacles and distance, it can impair the functionality of TCP.

- The downside of the mmWave in not being able to penetrate most of the materials forces unwanted packet drops in the network, which can be misleading to TCP because most of the conventional TCP assumes every packet drop is an indicator of congestion in the network. In addition to the conventional TCPs, cutting-edge TCPs such as BBR can not also function sufficiently in the new cellular networks.

- The malfunctionality of TCP wastes the wide bandwidth of the mmWave. Not having a proper TCP that can utilize the 5G mmWave to its full potential is one of the main issues that new generations will encounter. Without a well-designed TCP, 5G aims such as high throughput and low latency cannot be accomplished easily.

- Novel TCPs based on artificial intelligence solutions have been proposed to enhance the functionality of TCP. FB-TCP and DB-TCP are based on fuzzy and deep learning, respectively, which could improve the functionality of TCP over 5G networks, especially in the 3GPP's urban deployment scenario. The protocols could achieve throughput near the UDP's saturated value and minute latency through a low average sending rate.

The output of the thesis is promising as high throughput and low latency are pillar features of 5G networks. The novel TCPs can enhance the functionality of the protocol stack over the new cellular networks, as a result, improving the overall performance.

5.2 Future work

Some of the results in this thesis have been proven through a tremendous number of simulations utilizing one of the famous network simulators, i.e., NS-3 [87], [88]. Furthermore, Some other

analyses have been done over neural networks to identify the accuracy of the new protocol, i.e., DB-TCP. However, some real-time tests can be done in the future by implementing the protocols in the Linux kernel.

Moreover, the following open aspects can be considered in future work.

- Novel well-suited protocols for 5G mmWave can be designed in order to fulfill the other 3GPP's deployment scenarios.

- SDN/NFV-based protocols can be designed to gain broad insight from the network and control the flows intelligently and sufficiently.

- New queue control algorithms can be designed to control buffers intelligently.

- Novel link-layer mechanisms can be proposed to enhance the functionality of the protocol stack.

- The issues could be tougher by moving to 6G (Sixth Generation) and employing the Terahertz spectrum [113]. As a result, some elaborate efforts should be done to relieve the impediments on the way to accomplishing high performance. These efforts can be employing a wide range of techniques, from exploiting artificial intelligence to Non-Terrestrial Networks [114]. In [115] the initial requirements for network 2030, i.e., 6G, were presented. These features are going to improve most aspects of 5G networks, such as the following: 1 Gbs/m^2 area capacity, 10^7 device/km^2, approximately 100% reliability, better energy consumption, up to 1000 km/h mobility, 25µs to 1 ms, 1 Gbps user experience, and more than 1 Tbps peak data rate. 5G features are not capable of fulfilling the aforementioned goals. For example, because of some limitations, 5G can not virtualize any skills from one part of the world to another part below 1 ms latency. As a consequence, 6G along with its order-of-magnitude spectrum, employing frequencies more than 1 Thz will appear. These frequency ranges would indeed provide many new use cases, however, the analyzed issues in 5G mmWave could be tougher in the coming generation [116].

REFERENCES

[1] Several Authors, “Ericsson Mobility Report,” Publisher Fredrik Jejdling, Ericsson, June 2021. [Online]. Available: https://www.ericsson.com/4a03c2/assets/local/mobility-report/documents/2021/june-2021-ericsson-mobility-report.pdf.

[2] Several Authors, “5G PPP use cases and performance evaluation models,” Version 1.0, April 2016. [Online]. Available: https://5g-ppp.eu/wp-content/uploads/2014/02/5G-PPP-use-cases-and-performance-evaluation-modeling_v1.0.pdf.

[3] Several Authors, “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.

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

[5] A. Ghosh, A. Maeder, M. Baker, and D. Chandramouli, “5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15,” *IEEE Access*, vol. 7, pp. 127639-127651, Sep. 2019. DOI: 10.1109/ACCESS.2019.2939938.

[6] Several Authors, “NB-IoT deployment guide to basic feature set requirements,” GSMA, London, United Kingdom, June 2019. [Online]. Available: https://www.gsma.com/iot/wp-content/uploads/2019/07/201906-GSMA-NB-IoT-Deployment-Guide-v3.pdf.

[7] "Study on Scenarios and Requirements for Next Generation Access Technologies, V14.2.0,” 3GPP, Sophia Antipolis, France, Rep. TR 38.913, 2017. [Online]. Available: https://www.etsi.org/deliver/etsi_tr/138900_138999/138913/14.02.00_60/tr_138913v140200p.pdf.

[8] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, and T. Miyazawa, “Consideration on automation of 5G network slicing with machine learning,” *2018 ITU Kaleidoscope: Machine Learning 5G Future (ITU K)*, Santa Fe, Argentina, Nov. 2018 pp. 1-8. DOI: 10.23919/ITU-WT.2018.8597639.

[9] M. Series, ‘‘IMT vision—Framework and overall objectives of the future development of IMT for 2020 and beyond,’’ Tech. Rep. Recommendation ITU-R M.2083-0, 2015.

[10] "User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone, V15.5.0,” 3GPP, Sophia Antipolis, France, Rep. TS 38.101-1, 2019.

[11] "User Equipment (UE) radio transmission and reception; Part 2: Range 2 Standalone, V15.5.0,” 3GPP, Sophia Antipolis, France, Rep. TS 38.101-2, 2019.

[12] "Road to 5G: Introduction and Migration," GSMA, London, United Kingdom, April 2018. [Online]. Available: https://www.gsma.com/futurenetworks/wp-content/uploads/2018/04/Road-to-5G-Introduction-and-Migration_FINAL.pdf.

[13] D. Sattar and A. Matrawy, "Optimal Slice Allocation in 5G Core Networks," *IEEE Netw. Lett.*, vol. 1, no. 2, pp. 48-51, June 2019. DOI: 10.1109/LNET.2019.2908351.

[14] G. V. Murudkar and R. D. Gitlin, "Optimal-Capacity, Shortest Path Routing in Self-Organizing 5G Networks using Machine Learning," i*n Proc. IEEE 20th Wireless Microwave Technol. Conf. (WAMICON)*, Cocoa Beach, FL, USA, July 2019, pp. 1-5. DOI: 10.1109/WAMICON.2019.8765434.

[15] K. Katsaros and M. Dianati, "Evolution of Vehicular Communications within the Context of 5G Systems," *in Enabling 5G Communication Systems to Support Vertical Industries*, Hoboken, NJ, USA , 2019, pp.103-126, DOI: 10.1002/9781119515579.ch5.

[16] A. Tzanakaki, M. Anastasopoulos, and D. Simeonidou, "Converged Optical, Wireless, and Data Center Network Infrastructures for 5G Services," *J. Opt. Commun. Netw.*, vol. 11, no. 2, pp. A111-A122, Feb. 2019. DOI: 10.1364/JOCN.11.00A111.

[17] Q. Liu, R. Liu, Z. Wang, and Y. Zhang, "Simulation and Analysis of Device Positioning in 5G Ultra-Dense Network," *2019 15th Int. Wireless Commun. Mobile Compu. Conf. (IWCMC)*, Tangier, Morocco, Morocco, July 2019, pp. 1529-1533. DOI: 10.1109/IWCMC.2019.8766743.

[18] S. Martiradonna, A. Grassi, G. Piro, L. Grieco, and G. Boggia, "An Open Source Platform for Exploring NB-IoT System Performance," *In Proc. IEEE European Wireless Conf. (EW)*, Catania, Italy, May 2018, pp. 174–179.

[19] E. Balevi and R. Gitlin, "Unsupervised Machine Learning in 5G Networks for Low Latency Communications," *in Proc. IEEE Int. Perform. Comput. Commun. Conf. (IPCCC)*, San Diego, CA, USA, Dec. 2017, pp. 1-2. DOI: 10.1109/PCCC.2017.8280492.

[20] 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.

[21] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 173-196, Firstquarter, 2019. DOI: 10.1109/COMST.2018.2869411.

[22] M. Zhang et al., "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.

[23] 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, Third Quarter 2010. DOI: 10.1109/SURV.2010.042710.00114.

[24] 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.

[25] Postel, J., “Transmission Control Protocol,” STD 7, RFC 793, DOI 10.17487/RFC0793 RFC 793, September 1981, Updated by: RFC1122, RFC 3168, RFC 6093, RFC 6528 [Online]. Available: https://tools.ietf.org/html/rfc793.

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

[27] V. Paxson and M. Allman, "Computing TCP's retransmission timer," RFC 6298, June 2011. [Online]. Available: https://tools.ietf.org/html/rfc6298.

[28] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Comput. Commu. Rev.*, vol. 18, no.4, pp. 314-329, Aug. 1988. DOI: 10.1145/52325.52356.

[29] A. Mondal and A. Kuzmanovic, "Removing Exponential Backoff from TCP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no.5, pp. 19- 28, Oct. 2008. DOI: 10.1145/1452335.1452338.

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

[31] T. Kelly, “Scalable TCP: improving performance in highspeed wide area networks,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 83-9, April 2003. DOI: 10.1145/956981.956989.

[32] L. Xu, K. Harfoush, and I. Rhee, “Binary increase congestion control for fast, long distance networks,” *in Proc. IEEE INFOCOM*, Hong Kong, China, March 2004, pp. 2514–2524. DOI: 10.1109/INFCOM.2004.1354672.

[33] D. Leith, “H-TCP: TCP congestion control for high bandwidth delay product paths,” IETF Internet Draft, Oct. 2008. [Online] Available: https://tools.ietf.org/html/draft-leith-tcp-htcp-06, Accessed 25 Oct. 2017.

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

[35] G. Marfia et al., “TCP Libra: Exploring RTT-Fairness for TCP,” UCLA Computer Science Department, Tech. Rep. UCLA-CSD TR-050037, 2005.

[36] C. Caini and R. Firrincieli, “TCP Hybla: a TCP enhancement for heterogeneous networks,” *Int. J. Satellite Commun. Net.*, vol. 22, no. 5, pp. 547–566, Sep. 2004. [Online]. Available: https://dl.acm.org/doi/10.1002/sat.799.

[37] A. Baiocchi, A. P. Castellani, and F. Vacirca, “YeAH-TCP: yet another highspeed TCP,” *in Proc. PFLDnet, ISI*, Marina Del Rey (Los Angeles), California, USA, Feb. 2007, pp. 1-6.

[38] R. King, R. Baraniuk, and R. Riedi, “TCP-Africa: an adaptive and fair rapid increase rule for scalable TCP,” *in Proc. IEEE INFOCOM*, Miami, FL, USA March 2005, pp. 1838–1848, vol. 3. DOI: 10.1109/INFCOM.2005.1498463.

[39] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “A compound TCP approach for high-speed and long distance networks,” *in Proc. IEEE INFOCOM 2006*, Barcelona, Spain, April 2006, pp. 1-12. DOI: 10.1109/INFOCOM.2006.188.

[40] S. Liu, T. Basar, and R. Srikant, “TCP-Illinois: A loss and delay-based congestion control algorithm for high-speed networks,” *Perfor. Eval.*, vol. 65, no. 6-7, pp. 417-440, June 2008. [Online] Available: https://doi.org/10.1016/j.peva.2007.12.007.

[41] L. S. Brakmo, S.W. O’Malley, and L. Peterson, “TCP Vegas: new techniques for congestion detection and avoidance,” *ACM SIGCOMM Comput. Commun. Rev.*, vol .24, no. 4, August 1994. [Online]. Available: https://doi.org/10.1145/190809.190317.

[42] J. Sing and B. Soh, “TCP New Vegas: Improving the Performance of TCP Vegas Over High Latency Links,” *in Proc. 4th IEEE Int. Sympo. on Net. Comput. and Applicat. (IEEE NCA05)*, Cambridge, MA, USA, 2005, pp. 73–80. DOI: 10.1109/NCA.2005.52.

[43] K. Yamada, R. Wang, M.Y. Sanadidi, and M. Gerla, “TCP Westwood with Agile Probing: Dealing with Dynamic, Large, Leaky Pipes,” *2004 IEEE Int. Conf. on Commun. (IEEE Cat. No.04CH37577)*, Paris, France, June 2004, pp. 1070-1074, vol. 2. DOI: 10.1109/ICC.2004.1312665.

[44] D. Kliazovich, F. Granelli, and D. Miorandi, “Logarithmic window increase for TCP Westwood+ for improvement in high speed, long distance networks,” *Comput. Net.*, vol. 52, no. 12, pp. 2395–2410, Aug. 2008. [Online] Available: https://dl.acm.org/doi/abs/10.1016/j.comnet.2008.04.018.

[45] H. Shimonishi, T. Hama, and T. Murase, “TCP-Adaptive Reno for Improving Efficiency-Friendliness Tradeoffs of TCP Congestion Control Algorithm,” *in Proc. 4th Int. Wksp. on Protocols for Fast Long Distance Net.*, Feb. 2006, pp. 87-91

[46] K. Kaneko, T. Fujikawa, Z. Su, and J. Katto, “TCP-Fusion: a hybrid congestion control algorithm for high-speed networks,” *in Proc. Int. Wksp. PFLDnet, ISI*, Marina Del Rey (Los Angeles), California, USA April 2007, pp. 31-36.

[47] G. Hasegawa, K. Kurata, and M. Murata, “Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the Internet,” *in Proc. IEEE ICNP*, Osaka, Japan, Nov. 2002, pp. 177–186. DOI: 10.1109/ICNP.2000.896302.

[48] V. Paxson, “End-to-end Internet packet dynamics,” *in IEEE/ACM Transactions on Networking*, vol. 7, no. 3, pp. 277-292, June 1999, DOI: 10.1109/90.779192.

[49] R. Ludwig and R. H. Katz, “The Eifel algorithm: making TCP robust against spurious retransmissions,” *SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 1, pp. 30–36, Jan. 2000. [Online]. Available: https://dl.acm.org/doi/10.1145/505688.505692.

[50] R. Ludwig and A. Gurtov, “The Eifel Response Algorithm for TCP,” RFC 4015, Feb. 2005. [Online]. Available: https://tools.ietf.org/html/rfc4015.

[51] F. Wang and Y. Zhang, “Improving TCP performance over mobile adhoc networks with out-of-order detection and response,” *in Proc. 3rd ACM int. symp. mobile ad hoc net. Comput.*, New York, NY, USA, Jan. 2002, pp. 217–225. DOI: 10.1145/513800.513827.

[52] A. Venkataramani, R. Kokku, and M. Dahlin, “TCP Nice: A Mechanism for Background Transfers,” *Operat. Sys. Rev.*, vol. 36, no. SI, pp. 329–344, Dec. 2002. [Online]. Available: https://doi.org/10.1145/844128.844159.

[53] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “TCP Westwood: Bandwidth estimation for enhanced transport over wireless links,” *In Proc. ACM Mobicom*, Rome, Italy, July 2001, pp. 287–297. DOI: 10.1145/381677.381704.

[54] L. A. Grieco and S. Mascolo, “Performance evaluation and comparison of Westwood+, New Reno and Vegas TCP congestion control,” ACM SIGCOMM Comput. Commun. Rev., vol. 342, no. 2, April 2004. [Online]. Available: https://dl.acm.org/doi/10.1145/997150.997155.

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

[56] M. Pieska, A. Kassler, H. Lundqvist, and T. Cai, “Improving TCP Fairness over Latency Controlled 5G mmWave Communication Links,” *in 22nd Int. ITG Wksp. Smart Antennas (WSA ’18)*, Bochum, Germany, June 2018, pp. 1-8.

[57] 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: 10.1145/3009824.

[58] D. Scholz et al., "Towards a Deeper Understanding of TCP BBR Congestion Control," *2018 IFIP Netw. Conf. (IFIP Networking) Wksp.*, Zurich, Switzerland, 2018, pp. 1-9. DOI: 10.23919/IFIPNetworking.2018.8696830.

[59] R. Poorzare and A. Calveras, "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.

[60] . Narayanan, J. Carpenter, and E. Ramadan, "A First Measurement Study of Commercial mmWave 5G Performance on Smartphones," Sep. 2019, arXiv: 1909.07532. [Online]. Available: https://arxiv.org/pdf/1909.07532.pdf.

[61] SH. Hailea, K-J Grinnemoa, S. Ferlinb, P. Hurtiga, and A, Brunstroma, "End-to-End Congestion Control Approaches for High Throughput And Low Delay in 4G/5G Cellular Networks," *Computer Netw.*, vol. 186, Feb. 2021. DOI: 10.1016/j.comnet.2020.107692.

[62] 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.

[63] R. Poorzare and A. Calveras, "Open Trends on TCP Performance over Urban 5G mmWave Networks," *In PE-WASUN'20: 17th ACM Symposium Performance Eval. Wireless Ad Hoc, Sensor, & Ubiquitous Net. Proceedings*, Alicante, Spain, Nov. 2020, pp.85-92. DOI: https://doi.org/10.1145/3416011.3424749.

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

[65] "3rd generation partnership project; technical specification group radio access network; study on scenarios and requirements for next generation access technologies, v14.3.0" 3GPP, Sophia Antipolis, France, Rep. TR 38.913, Jun. 2017. [Online]. Available: www.3gpp.org/dynareport/38913.htm.

[66] "Guidelines for evaluation of radio interface technologies for IMT-2020," ITU, Geneva, Switzerland, ITU-Recommendation M.2412, Oct. 2017. [Online]. Available: https://www.itu.int/dms_pub/itur/opb/rep/R-REP-M.2412-2017-PDF-E.pdf.

[67] 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 & Tut.*, vol. 22, no. 2 pp. 905-929, Second quarter 2020. DOI: 10.1109/COMST.2020.2971781.

[68] K. Nichols, V. Jacobson, A. McGregor, and J. Iyengar, “Controlled Delay Active Queue Management,” RFC 8289, Jan. 2018. [Online]. Available: https://tools.ietf.org/html/rfc8289.

[69] T. Hoeiland-Joergensen, P. McKenney, D. Taht, J. Gettys, and E. Dumazet, ”The Flow Queue CoDel Packet Scheduler and Active Queue Management Algorithm,” RFC 8290, Jan. 2018. [Online]. Available: https://tools.ietf.org/html/rfc8290.

[70] M. Allman, S. Floyd, and C. Partridge, “Increasing TCP's Initial Window,” RFC 3390, Oct. 2002. [Online]. Available: https://tools.ietf.org/html/rfc3390.

[71] M. Polese, R. Jana, and M. Zorzi, “TCP in 5G mmWave Networks: Link Level Retransmissions and MP-TCP,” *in Proc. IEEE Conf. Comput.Commun. Wksp. (INFOCOM WKSHPS)*, Atlanta, GA, USA, Nov. 2017, pp. 343-348. DOI: 10.1109/INFCOMW.2017.8116400.

[72] K. Liu and J. Y. B. Lee, "On Improving TCP Performance over Mobile Data Networks," *IEEE Transactions on Mobile Comp.*, vol. 15, no. 10, pp. 2522-2536, Oct. 2016, DOI: 10.1109/TMC.2015.2500227.

[73] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP Selective Acknowledgment Options,” RFC 2018, Oct. 1996. [Online]. Available: https://tools.ietf.org/html/rfc2018.

[74] J. Iyengar and M. Thomson, “QUIC: A UDP-Based Multiplexed and Secure Transport,” Work in Progress, draft-ietf-quic-transport-33, Dec. 2020. [Online]. Available: https://tools.ietf.org/html/draft-ietf-quic-transport-33.

[75] I. Petrov1 and T. Janevski, “Design of novel 5G transport protocol,” 2016 *Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Fez, Morocco, Oct. 2016, pp. 29-33. DOI: 10.1109/WINCOM.2016.7777186.

[76] 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.

[77] J. Rodriguez, “Fundamentals of 5G Mobile Networks,” 1st ed., Hoboken, New Jersey, United States, John Wiley and Sons, April 2015. DOI: 10.1002/9781118867464.

[78] Y. Hasegawa and J. Katto, “A Transmission Control Protocol for Long Distance High-Speed Wireless Communications,” IEICE Trans. on Comm., vol. E101-B, no. 4, pp. 1045-1054, Apr. 2018. DOI: 10.1587/transcom.2017EBP3229.

[79] M. R. Kanagarathinam et al., "NexGen D-TCP: Next Generation Dynamic TCP Congestion Control Algorithm," IEEE Access, vol. 8, pp. 164482-164496, 2020, DOI: 10.1109/ACCESS.2020.3022284.

[80] J. Gettys and K. Nichols, "Bufferbloat: Dark Buffers in the Internet," *Commun. ACM*, vol. 55, no. 1, pp. 57-65, Jan. 2012. DOI: 10.1145/2063176.2063196.

[81] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog Computing and its Role in the Internet of Things," *in Proc. 1st Edition MCC Wksp. Mobile Cloud Comput.*, New York, NY, USA, Aug. 2012, pp. 13-16. [Online]. Available: https://dl.acm.org/doi/10.1145/2342509.2342513.

[82] M. Chiang and T. Zhang, "Fog and IoT: An overview of research opportunities" *IEEE Internet Things J.*, vol. 3, no. 6, pp. 854-864, June 2016. DOI: 10.1109/JIOT.2016.2584538.

[83] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, "TCP Extensions for Multipath Operation with Multiple Addresses," RFC 8684, March 2002. [Online]. Available: https://tools.ietf.org/html/rfc8684.

[84] 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. Model. Anal. Simulat. Wireless Mobile Syst. (MSWIM-2011)*, Miami Beach, FL, USA, Nov. 2011, pp. 293–298. [Online]. Available: http://doi.acm.org/10.1145/2068897.2068948.

[85] "LTE-EPC Network simulAtor," CTTC [Online]. Available: http://networks.cttc.es/mobile-networks/software-tools/lena/ [Verified: January 2020].

[86] 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. Model., Anal. Simulat. Wireless Mobile Syst.*, Cancun, Mexico, June 2015, pp. 283–290. [Online] Available: https://dl.acm.org/doi/10.1145/2811587.2811619.

[87] Network Simulator 3. Accessed: Feb. 2, 2020. [Online]. Available: https://www.nsnam.org/.

[88] 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, pp. 527, 2008.

[89] 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, Jul. 2017, pp. 1-6. DOI: 10.1109/SPAWC.2017.8227746.

[90] 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," *WNS3 '17 Proc. Wksp. ns-3*, Porto,

Portugal, June 2017, pp. 71–78. [Online]. Available: https://dl.acm.org/doi/10.1145/3067665.3067678.

[91] 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, June 2017. DOI: 10.1109/JSAC.2017.2720338.

[92] 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. Simulat. Tools Techn. (SIMUTOOLS)*, Prague, Czech Republic, Aug. 2016, pp. 118–123. [Online]. Available: https://dl.acm.org/doi/10.5555/3021426.3021445.

[93] M. Mezzavilla et al., “End-to-End Simulation of 5G mmWave Networks,” *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2237-2263, Apr. 2018. DOI: 10.1109/COMST.2018.2828880.

[94] H. Tazaki et al., “Direct code execution: Revisiting library OS architecture for reproducible network experiments,” *in Proc. 9th ACM Conf. Emer. Netw. Exp. Technol. (CoNEXT)*, Santa Barbara, California, USA, Dec. 2013, pp. 217–228. [Online]. Available: https://dl.acm.org/doi/10.1145/2535372.2535374.

[95] "Study on channel model for frequency spectrum above 6 GHz, V14.2.0,” 3GPP, Sophia Antipolis, France, Rep. TR 38.900, 2017.

[96] MATLAB, 5G library for LTE System Toolbox, [Online]. Available: https://www.mathworks.com/products/5g.html.

[97] S. Choi, J. Song, J. Kim, and S. Lim, “5G K-SimNet: End-to-End Performance Evaluation of 5G Cellular Systems,” *2019 16th IEEE Annu. Cons. Commun. Netw. Conf. (CCNC)*, Las Vegas, NV, USA, 2019, pp. 1-6. DOI: 10.1109/CCNC.2019.8651686.

[98] Google. TensorFlow. [Online]. Available: https://www.tensorflow.org.

[99] NumPy Developers. NumPy. [Online]. Available: https://numpy.org.

[100] SciPy. Pandas. [Online]. Available: https://pandas.pydata.org.

[101] The Matplotlib development team. Matplot. [Online]. Available: https://matplotlib.org.

[102] A. Zadeh, ‘‘Fuzzy sets,’’ *Inf. Control*, vol. 8, no. 3, pp. 338–353, Jun. 1965.

[103] R. Czabanski, M. Jezewski, and J. Leski, “Introduction to Fuzzy Systems,” Theory and Applications of Ordered Fuzzy Numbers,” Springer, Cham, vol 356, pp.23-43, Oct. 2017. DOI: https://doi.org/10.1007/978-3-319-59614-3_2.

[104] M. Pieska and A. Kassler, “TCP performance over 5G mmWave links — Tradeoff between capacity and latency,” 2017 IEEE 13th International Conf. on Wireless and Mobile Comput., Netw. and Commun. (WiMob), Rome, Oct. 2017, pp. 385-394, DOI: 10.1109/WiMOB.2017.8115776.

[105] 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.

[106] https://github.com/rezapoorzare1/FB-TCP-a-5G-mmWave-Friendly-TCP-for-Urban-Deployments/tree/main.

[107] Y. Ren, W. Yang, X. Zhou, H. Chen, and B. Liu, “A survey on TCP over mmWave,” Computer Commun., vol. 171, pp. 80-88, Apr. 2021. DOI: 10.1016/j.comcom.2021.01.032.

[108] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Netw.*, vol. 61, pp. 85-117, Jan. 2015. DOI: https://doi.org/10.1016/j.neunet.2014.09.003.

[109] R. Poorzare and A. Calveras, “FB-TCP: A 5G mmWave Friendly TCP for Urban Deployments,” *IEEE Access*, vol. 9, pp. 82812-82832, Jun. 2021, DOI: 10.1109/ACCESS.2021.3087239.

[110] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res*., vol. 15, no. 1, pp. 1929-1958, Jun. 2014.

[111] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proc. Artif. Intell. Statist.*, May 2010, pp. 249-256.

[112] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” in *Proc. ICLR*, Jan. 2014, pp. 1412-6980.

[113] 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.

[114] Giordani and M. Zorzi, “Non-Terrestrial Networks in the 6G Era: Challenges and Opportunities,” *IEEE Netw.*, vol. 35, no. 2, pp. 244-251, Mar./Apr. 2021, DOI: 10.1109/MNET.011.2000493.

[115] Focus Group on Technologies for Network 2030: Representative Use Cases and Key Network Requirements, document ITU-T, Focus Group 2030, Feb. 2020.

[116] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G Wireless Systems: Vision, Requirements, Challenges, Insights, and Opportunities," in *Proceedings of the IEEE*, vol. 109, no. 7, pp. 1166-1199, July 2021, DOI: 10.1109/JPROC.2021.3061701.