Received 19 September 2023; accepted 11 October 2023. Date of publication 18 October 2023; date of current version 2 November 2023.

Digital Object Identifier 10.1109/OJCOMS.2023.3325764

On the Performance of Transport Protocols Over mmWave Links: Empirical Comparison of TCP and QUIC

SIDNA JEDDOU [1], LUIS DIEZ [2], NAJID ABDELLAH [1], AMINE BAINA [1], AND RAMÓN AGÜERO CALVO [2] (Senior Member, IEEE)

[1] Department of Communication Systems, National Institute of Posts and Telecommunications, Rabat 10170, Morocco

[2] Department of Communication Engineering, Universidad de Cantabria, 39005 Santander, Spain

CORRESPONDING AUTHOR: S. JEDDOU (e-mail: jeddou.sidna@inpt.ac.ma)

This work was supported by the Spanish Government (Ministerio de Economía y Competitividad, Fondo Europeo de Desarrollo Regional, MINECO-FEDER) by means of the Project SITED: Semantically-Enabled Interoperable Trustworthy Enriched Data-Spaces under Grant PID2021-125725OB-I00.

ABSTRACT The extensive availability of spectrum resources and the remarkably high data transmission rate of millimeter-wave (mmWave) technology have propelled its significance as a vital component in the advancement of mobile communications, including fifth generation (5G) networks. However, the intermittent nature of mmWave links and their interaction with transport layer protocols pose several challenges, which bring inadequate performance, due to fluctuations in high-frequency channels. Consequently, although these features of mmWave might be advantageous, they can actually hinder the performance. Although these issues have been studied in the literature with TCP, there are few works that have studied how QUIC behaves over this kind of channels. This paper aims to compare the performance of TCP and QUIC over mmWave channels, studying the impact at the application level. We conduct extensive performance evaluations, based on traces that are obtained by means of a detailed simulation of different mmWave scenarios, using the ns-3 simulator. We analyze key performance indicators, such as delay, throughput, and bottleneck buffer. The results evince that QUIC outperforms TCP in highly fluctuating mmWave channels, showing better performance for both throughput and delay.

INDEX TERMS QUIC, mmWave, ns-3, TCP, delay, data rate, buffer size.

I. INTRODUCTION

WE HAVE witnessed, in the last years, a rapid evolution of mobile communication technologies, which has crystallized in the development of 5G/B5G networks, which promise enhancements of $\approx 1000\times$ in channel capacity, $\approx 10\times$ in effectiveness, $\approx 100\times$ in data rates, and $\approx 25\times$ in cell throughput, in comparison to 4G [1]. Moreover, one of the goals of the recent 5G cellular technologies is to fulfill end-user surging needs, which are continuously increasing, while providing support for large-scale networking areas, with a massive number of devices, including Internet of Things (IoT) and the so-called industry 4.0. Features including high availability, extremely large throughput, low latency, and reliability, must be considered to fulfill end-user requirements [2].

Among other technological shifts, 5G technologies exploit the additional capacity brought about by higher frequency bands, such as millimeter-wave (mmWave), which ranges from 30 to 300 GHz. According to [3], [4], 5G mmWave will support critical use cases, such as Enhanced Mobile Broadband (eMBB), in which large wireless network resources should be made available to end-user equipment (UE), for instance to cope with video surveillance and UHD video streaming. Also, massive Machine Type Communications (mMTC) needs to be supported, to enable managing a large number of connected devices, while base stations (BS) and UE should be connected with a latency lower than one millisecond, according to Ultra-Reliable Low-Latency Communication (URLLC), which strives to meet automatic driving and real-time immersive video streaming requirements.

Although the aforementioned use cases were already considered in 4G, the number of connected devices requiring larger capacities and more strict requirements is continuously increasing. Hence, in order to address such challenges, overcome the limited capacity of sub-6 GHz spectrum resources, and fulfill the aforementioned requirements, the use of wider spectrum frequency bands is believed to be a promising solution [3]. Among these novel PHY and link layer technologies, mmWave is expected to be an excellent candidate due to its ultra-high frequency, playing a significant role in fulfilling the performance targets of next mobile network generations [5]. In fact, mmWave is being introduced as a promising and appealing technology that provides multi-gigahertz of unlicensed channels capacity for 5G wireless communications, allowing massive data transmissions with higher data rates, and low latency, which makes it suitable to address the requirements of a wide range of IoT applications including smart city, intelligent transportation, healthcare, logistics, innovative industry, and so on [6].

Nevertheless, mmWave faces several challenges and constraints, which require further research. This includes high penetration loss and poor isotropic propagation, which may hinder exploiting the high transmission capacity in large areas. This makes this technology to be less attractive, as many objects like buildings, cars, and even human bodies can prohibit mmWave from reaching its goal [2]. Also, it is more unstable than conventional wireless technologies, due to its shorter wavelength [7]. As a consequence, channel quality between UEs and BSs drastically drop in Non Line of Sight (NLoS) situations, so that keeping stable connections is not possible due to the presence of communication hurdles [2], [8].

The instability of the physical channel impacts the upper layers of the protocol stack, which perceive congestion or losses upon blockage on the mmWave channel, so that the performance of end-to-end communications is highly jeopardized. As a result, issues including unstable, fluctuating network connections, reduced throughput, increased packet loss rates, and higher Round-Trip Time (RTT) have been detected [9]. The authors of [1] highlight that these extremely varying radio channels bring issues not only on the PHY and MAC layers, but also on higher layers, including transport protocols and end-to-end applications. The congestion mechanisms of Transmission Control Protocol (TCP) consider the intermittence of mmWave as network congestion, so severely limiting the amount of transferred data. Due to limited transmission rates, the enormous bandwidth of mmWave may not be used to its full potential [9]. Furthermore, the interaction with the transport layer protocol plays a key role in determining the end-to-end performance perceived by end-users. The congestion control mechanisms can be affected by blockages, resulting in Retransmission Time Out (RTO) expiration events, and wrong channel state tracking [5]. In addition, since the TCP server waits for one RTT time before receiving an acknowledgment (ACK) packet, it might take a considerable amount of time to use the entire network bandwidth, when the estimated RTT is larger, which could happen due to the effect of the physical channel [10], [11].

Considering the aforementioned issues mentioned, an adaptation between mmWave and transport layers, such as TCP and QUIC, is needed, so that the high potential capacity brought by mmWave channels can be actually exploited. TCP is the most commonly used transport protocol that supports most current Internet services, including those used in IoT environments. It offers reliable end-to-end connectivity, flow control, and congestion management.

However, due to its conservative congestion detection and control mechanisms, TCP frequently wastes capacity. This issue might be particularly severe in mmWave channels [12]. In addition to capacity fluctuations, recent studies [1] show that TCP effectiveness can also be considerably reduced, due to the high error rate of mmWave channels, bringing particular hassle to keep its reliability in mmWave environments. Altogether, drawbacks including channel fluctuations and losses can severely hinder the congestion control in TCP protocol, and so restrict the benefit of additional available capacity when using mmWave technology.

To address some of TCP's limitations and inefficiencies, QUIC protocol was recently developed and standardized. This protocol is built on top of the User Datagram Protocol (UDP) and provides, among other features, flow and congestion control, reliable communications, in-order delivery to upper layers, and multi-stream management. Google [13] pioneered QUIC, which was later standardized by the Internet Engineering Task Force (IETF) [14]. QUIC improves congestion detection mechanisms compared to TCP [15]. Due to expected QUIC's greater performance, which entails lower latency, easier connection initiation, stream multiplexing, and the capacity to avoid head-of-line blocking, many players and stakeholders have already started using it. Although there are various works that assess TCP behavior over mmWave links from different angles, such as TCP adaptation, throughput performance, new TCP variations, and multiple communication channels, very few have actually analyzed QUIC's performance over such links.

This work aims to complement existing studies by comparing the performance of TCP and QUIC over mmWave links in terms of throughput, packet delay, and their impact over the bottleneck buffer link, under a variety of configurations. The contributions of the paper are the following ones:

- Traces generation by simulating multiple mmWave scenarios using ns-3, to carry out the performance assessment.
- Building an evaluation framework and methodology, which comprises a client and server communicating, using either TCP or QUIC, over a channel based on the generated traces, exploiting the Mahimahi link emulator.
- Performance comparison of TCP and QUIC, using different metrics and configurations, over the aforementioned traces.

The rest of the paper is organized as follows. Section II describes the related work, highlighting how this research stands out from existing papers. Then, Section III depicts in detail the different generated traces, as well as the mmWave scenarios that were used to run the experiments, whose results are discussed in Section IV. The paper concludes in Section V, where we provide an outlook of our future work.

II. RELATED WORK

In the following we analyze the existing literature that aims to study, or to improve, the performance of transport layer solutions over mmWave channels. In addition, we will also review the main research trends related to the QUIC protocol.

Most of the existing works aim to analyze the impact of the intermittent behavior of mmWave, paying special attention to how it affects the transport protocol performance. Ren et al. [11] discussed the characteristics of mmWave technology, and assessed the areas where TCP needs to adapt to this technology. They also evaluated alternative solutions to address such adaptation issues, categorizing them into four groups: congestion control algorithms, retransmission proxies, use of multiple paths, and enhancements based on machine learning. Slezak et al. [16] comprehensively studied the impact of blockage over end-to-end performance in mmWave networks. They conducted simulations and measurements based on ns-3 to examine three main features: channel dynamics, 3GPP New Radio beam, and Internet protocol stack. Poorzare and Calveras [4] analyzed TCP performance under various scenarios in an urban setting, to understand the causes that might impact the protocol's performance. Their findings demonstrated that standard TCP implementations could not fully exploit 5G mmWave links. In another work, Poorzare and Augé [17] proposed a new TCP version based on fuzzy logic to mitigate performance degradation in urban deployments. The protocol's congestion detection phase employed fuzzy rules to intelligently adjust the transmitting rate, based on the network state and reducing the impact of blockages for optimal performance.

Kim et al. presented in [18] an alternative TCP design for mmWave communications called mmWave Performance Enhancing Proxy (mmPEP). This design aimed to avoid TCP performance collapse and leverage the characteristics of mmWave channels to improve the end-to-end rate and packet delivery ratio. This solution aimed to treat the mmWave link separately from the other network segments involved in the end-to-end communication, and demonstrated that this approach can indeed increase the application performance. Le et al. [1] proposed an analytical framework to investigate TCP performance over mmWave channels. They considered obstacles, such as buildings, which caused blockages, and fading channels. In this work, the Nakagami-m distribution was used to model the mmWave fading channel. The analysis included the interaction of TCP with protocols like truncated incremental redundancy hybrid automatic repeat request (IR-HARQ) and adaptive modulation and coding (AMC). A transmission loss model based on a finite-state Markov chain (FSMC) was developed, and Monte Carlo simulations were used to validate the analysis.

Yang et al. built-in in [7] a testbed for a hybrid network using 10 Gigabit cables and 60 GHz mmWave. They evaluated the effectiveness of various TCP congestion control algorithms, including NewReno, CUBIC, BBR, and Westwood, characterizing their impact over TCP performance. The study also investigated the impact of a performance-enhancing proxy (PEP), which divided the end-to-end transmission into mmWave and wired portions when the connection state changed.

Interestingly, there is a set of works that have exploited the ns-3 mmWave module to carry out the assessment of TCP features when it interacts with mmWave links. Mateo et al. introduced in [19] an analysis of the end-to-end performance of TCP in mmWave networks in terms of loss, delay and hybrid protocols, using various applications and scenarios. Lee et al. [10] implemented a TCP proxy in the ns-3 module, and evaluated different performance indicators, including delays and TCP proxy buffer sizes, using Multi-connectivity (MC). Hassan and Mowla [20] examined various performance features (pertaining to both the transport protocol and the physical layer), such as SINR, RTT, congestion window, and data rate. They studied the mmWave channel by establishing a framework based on the ns-3 simulator. Abdulrazzak et al. presented in [21] another ns3-based mmWave framework, where they carry out a simulation-based study embracing TCP and mmWave, where they focus on three main performance parameters: RTT, congestion window size, and throughput. Siddiqui and Chau [9] assessed the performance of various TCP congestion control techniques over mmWave communication links, using five alternative TCP versions (New Reno, YeAH, Hybla, Westwood, and Vegas). They analyzed throughput, Round Trip Time (RTT), Signal-to-Interference-plus-Noise Ratio (SINR), and congestion window size.

Polese et al. [5] used the mmWave module in the ns-3 simulator to methodically examine the performance of the Linux kernel's TCP/IP stack implementation over mmWave networks. They analyzed throughput and latency, with and without link-layer re-transmissions, paying attention to the interaction with lower-layer re-transmission protocols. They introduced Multi-path TCP (MP-TCP) with various congestion management algorithms over LTE and mmWave links, and showed how multi-path transmissions enhance the performance of the mmWave network. Poorzare and Augé [8] analyzed the behavior of TCP under an urban deployment scenario from the 3GPP perspective. They covered the effect of various factors, including distant servers, RLC buffer size, congestion management techniques, and maximum segment size, as well as potential benefits of using TCP over 5G mmWave in a city deployment. The findings showed that TCP might benefit from the deployment of edge servers, and that increasing the maximum segment size can further improve the performance. Variable RLC buffer sizes also impact how different TCP variations react.

A cross-layer technique for TCP uplink is proposed by Azzino et al. in [22]. It aims to minimize latency for NLOS conditions and to maintain higher throughput. They compared it with TCP CUBIC and other congestion control techniques in terms of latency and throughput. Kim and Cho [23] proposed an enhancement of the scalable TCP algorithm based on an amendment of the congestion window mechanism to ensure appropriate performance in 5G mmWave networks without significantly affecting TCP fairness. Okano et al. [3] analyzed the effects of a NLOS environment caused by human body obstructions, evaluating the throughput capabilities of 5G inside a train station. They employed a TCP free-space optical (TCP-FSO) to enhance regular TCP transmission performance in such an environment. Zhang et al. discuss in [12] various results of a TCP evaluation over mmWave links, considering TCP BBR congestion control, remote servers, multiple concurrent connections, buffer size, and 3GPP features.

In addition, there exist a number of studies that have analyzed the performance of the QUIC transport protocol. For instance, Polese et al. discussed in [24] the main characteristics of several transport solutions, such as Stream Control Transmission Protocol (SCTP), Datagram Congestion Control Protocol (DCCP), TCP, and QUIC. Nepomuceno et al. [25] compared the benefits of QUIC and TCP in terms of download time for Web traffic from various websites. Jung and An [26] suggested an enhancement for video streaming and Web data services using QUIC in order to improve Quality of Experience (QoE). The paper showed a reduction in delay, based on an evaluation of the congestion window whenever a new connection is opened.

The performance of QUIC has been also assessed in terms of packet loss, delay, and jitter by Bulgarella et al. [27], where the authors proposed new solutions to accurately measure the delay. Similarly, Yu et al. [28] investigated QUIC behavior, with a focus on congestion control and packet pacing techniques. The results evince the benefits of QUIC's capability for multi-stream multiplexing.

Several approaches, including simulation and emulation environments, have been used to examine the performance of QUIC. De Biasio et al. implemented QUIC in the ns-3 simulator, embracing all of its key features [29]. Additionally, Kakhki et al. investigated QUIC's performance in a simulated environment [30].

On the other hand, some studies have examined QUIC's effectiveness as a TCP replacement for IoT or Industrial IoT (IIoT) scenarios. Herrero [31] studied QUIC as a TCP and UDP concurrent solution for IoT services based on CoAP. Analytical models of CoAP performance across UDP, TCP, and QUIC were developed, followed by a test analysis using the Visual Protocol Stack simulator (VPS+). The performance of QUIC in IoT services was analyzed in [32], including comparisons with well-known protocols such as CoAP, MQTT, and MQTT-SN. The analysis was conducted using a combination of real devices and a simulated environment.

Also in the scope of IoT/IIoT, Fernández et al. [33], [34] assessed the performance of MQTT over QUIC, using Linux containers and emulating different wireless technologies and channel characteristics with the ns-3 network simulator. Additionally, Jeddou et al. analyzed the behavior of MQTT, over both TCP and QUIC, in terms of traffic delay and energy consumption, based on a platform that uses real devices [35].

Addressing a different problem, Gärdborn [36] assessed the performance of QUIC and TCP in the context of 5G Core Network Service Based Architecture. Similar to our work, this study compares TCP and QUIC. However, the scenarios they considered and the corresponding methodology are different, and high capacity mmWave channels are not considered at all. On the other hand, Hasselquist et al. [37] present a performance assessment of QUIC and TCP over Dual Connectivity (DC), considering bandwidth, delay, and loss conditions. They illustrate how QUIC is influenced by varying DC parameters and diverse network conditions. Nevertheless, the work does not consider either high capacity mmWave channels.

To the authors' best knowledge, there are very few works that jointly consider QUIC and mmWave channels. In [38] Sinha et al. propose a novel migration mechanism, Cross-layer QUIC (CQUIC), to improve handover procedures. The work focuses on mmWave scenarios, where handovers to umbrella macro cells can be frequent. However, this work does not aim to analyze QUIC performance over mmWave links. Haile et al. [39] proposes a modification of the BBR congestion control algorithm for QUIC and evaluates it over different scenarios, including 4G and 5G connectivity, based on mmWave. Besides, Haile et al. introduce in [40] a mechanism to enhance BBR delay performance in QUIC by utilizing a 5G emulation testbed. Krämer et al. [41] propose a multi-domain PEP with supports both TCP and QUIC connections, whose behavior is analyzed with connections over mmWave channels. As can be seen, the scope of these works differs from ours, since they do not aim to compare the performance of QUIC and TCP over mmWave channels, but to analyze the operation of their proposal.

In a very recent work, Khorov et al. [42] reviewed the issues contributing to suboptimal TCP and QUIC performance over high-frequency environments like mmWave, LightWave, and Terahertz. They evaluated already existing solutions, proposed by IETF, 3GPP, IEEE, and the research community to address these challenges. After conducting this analysis, they provide recommendations regarding the solutions that yield the most significant enhancements in high-frequency bands.

The work closer to ours is the one by Haile et al. [43], where the authors compare the performance of different congestion control algorithms such as BBR, Copa, CUBIC and RBBR (a BBR modification), by using the mvfst QUIC implementation. Different to [43], we aim to compare TCP and QUIC to better understand the expected performance differences at the application level when working over mmWave links.

TABLE 1. Scenarios description. In all cases we adopt 3GPP propagation models, and an operation frequency of 28 GHz.

Scenario	Description
Buildings 1 (B1) Buildings 2 (B2)	One user moves between two buildings in a street canyon from (B1) and towards (B2) a base station at a constant walking speed.
Buildings 3 (B3a & B3b)	We deploy a grid of 3 × 3 buildings uniformly distributed (see Figure 1a) and a base station between two buildings. Over such scenario the user moves following a predefined pattern from and towards different buildings at walking speed. Patterns in the two scenarios (3a & 3b) are different.
Buildings 4 (B4)	Two mmWave eNBs separated by a distance of 100m, and a user moves following a straight line parallel to the one formed by the base stations. Then, 8 buildings of different sizes are randomly deployed between the user and the base stations.
Campus (CA)	Buildings are deployed following a very specific pattern that mimics a university campus (NYU-AD campus), and a base station is located at the main building. User moves around the campus area at walking speed.
Rural Macro (RM)	Users move in an outdoor rural area following a random walk over a square of 30 × 30m, and 25m away from the base station.

All in all, we believe that this work fills an existing gap in the literature, by comparing the potential performance of the most relevant transport protocols over different mmWave scenarios, and under different configurations.

III. EVALUATION SETUP

This section describes the evaluation framework that was built to carry out the performance assessment. In order to mimic the behavior of mmWave links, we have made use of the Mahimahi link emulator [44]. It consists of a set of lightweight tools (UNIX shells) able to model different communication conditions. While it was primarily conceived as a record-and-replay tool, it can be also fed with communication traces, broadening its application scope. When used for link emulation based on traces, Mahimahi is able to define the transmission capacity over time, with a resolution of 1 ms. indicating the number of bytes that can be sent every millisecond. At the edges of the communication we use a pair of client and server applications that send traffic through the aforementioned emulated link. The applications are implemented in go programming language, and in the case of QUIC, the quic-go implementation[1] is used.

The traces used in the performance evaluation have been obtained through a detailed simulation of different scenarios with ns-3. Although the traces generated through simulation may be less realistic than real measurements, they would evenly affect TCP and QUIC, thus yielding a fair comparison. In addition, the evaluation methodology that we have used presents some benefits. First, it allows us to define the environment and the particular configuration of the evaluation scenarios. In addition, using traces allows reproducibility of results for systematic evaluation. In the following we describe the process to generate the traces and the scenarios that we analyze in this work.

1. https://github.com/lucas-clemente/quic-go

A. TRACES GENERATION

We adopt the mmWave module [45] to set up different scenarios. This module is built within the LTE module in ns-3, to support 5G end-to-end network simulations [46], [47]. It implements 3GPP models [48] for frequency spectrum above 6 GHz, as well as other models proposed in the literature [49].

We consider a set of scenarios with one or more mmWave base stations operating at 28 GHz with a channel width of 140 MHz. In all cases, base stations and users have 64 and 16 antenna elements, respectively. In each scenario, users connect to a server sending packets of 1472 bytes at a constant rate of 1 Gbps during 3 minutes. In order to exploit the full link capacity, UDP was used as the transport protocol, so that we avoid the impact of the congestion and flow control algorithms during the generation of traces. Table 2 summarizes the common configuration setup that was used for trace generation in all scenarios. Eventually, the capacity traces are obtained by recording packet receptions at the transport layer. In addition, evolution of metrics of the underlying wireless channel and cellular protocol stack are also gathered, such as Signal o Noise Ratio (SINR) or Radio Link Control (RLC) buffer.

The generation of traces is automated by means of scripts, written in Python language, that permits the rapid configuration of different scenarios parameters. It is worth noting that, depending on the hardware used, generating the trace of a single scenario takes several hours.

B. ANALYZED SCENARIOS

In general we create scenarios embracing users, mmWave base stations and buildings, and we apply the 3GPP channel models [48], [49]. The users move around the scenario generating Line of Sight (LOS) and non-Line of Sight (NLOS) conditions, causing the channel capacity to vary. In Table 1 we summarize the different scenarios that we considered. For each of them, 5 independent simulation of 3 minutes have been conducted, leading to a total of

TABLE 2. Characteristics of the scenario used to generate the traces.

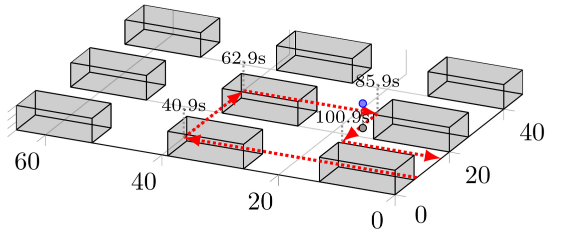
Radio Link	
Operation frequency	28 GHz
Bandwidth	140 MHz
# antennas BS	64
# antennas UE	16
Propagation model	3GPP (LOS & NLOS)
Trace traffic	
Pkt. size	1472 bytes
Traffic shape	UDP on saturation
Scenarios	
Users motion	Walking speed $[1, 3]$ m/s
Scenario details	Table 1

35 different mmWave traces. It is worth noting that the randomness of the mmWave channel gives rise to different traces, even when the considered scenario is the same. In all cases the users move at walking speed between 1 and 3 m/s.

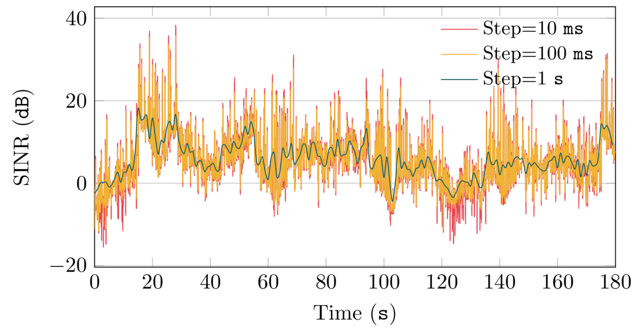
The first two scenarios, namely **B1** and **B2**, consider a street canyon situations where users move between two buildings. The base station is located at one end of the street, so that the channel varies as the users move approaching (**B1**) or receding (**B2**) the base station.

Then, we have generated scenarios comprising several buildings that follow a uniform pattern. Scenarios **B3a** and **B3b** comprise a grid of 3×3 buildings with straight streets, over which users move, following a predefined motion pattern. As an example, Figure 1 shows the setup of scenario **B3a**, and one trace that was obtained from it. As can be observed in Figure 1(a), the scenario mimics a Manhattan-like environment in which a base station (blue circle) provides connectivity to users. Our target user moves between buildings along the streets following a defined route. The figure also shows the simulation times where the user changes the direction.

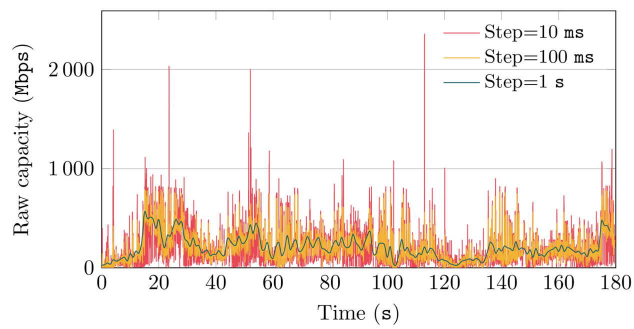
Figure 1(b) depicts the temporal evolution of the SINR experienced by the user during the simulation; it is worth recalling that the user keeps sending UDP traffic at 1Gbps during the whole simulation. In order to better show the variability of the mmWave channel, Figure 1(b) represents the SINR with different temporal granularity: 1 second, 100 ms and 1 ms. In all cases, the SINR is obtained as the average value during the specified time interval. As can be observed, the configuration with 1 second slot shows the overall variation with a general dynamic range of around 20 dB, but it is not able to capture the actual channel fluctuation. Indeed, finer representations show that the actual dynamic range of the SINR is above 50 dB (i.e., from close to −20 up to almost 40 dB), exhibiting rapid changes. We then represent in Figure 1(c) the communication capacity seen at the transport layer over the same channel, which is obtained by measuring the reception rate of UDP traffic. Once again, the capacity is measured with different granularity. As can be observed, when using time slots of 1 second the maximum capacity hardly reaches 500 Mbps, showing a relatively smooth variation. However, when decreasing the time slot we can observe that the transport protocols will perceive a much more varying channel, which would have a high impact on the protocol performance. It is worth recalling that Mahimahi operates with resolution of 1 ms, so that it will be able to reproduce the actual channel variability.



(a) Scenario and target user motion following the red dotted path. The blue circle indicates the location of the mmWave base station. The boxes in grey indicate the buildings



(b) Trace SINR evolution



(c) Trace capacity evolution

FIGURE 1. Traces and user motion in scenario ***B1***.

In order to obtain traces less affected by the specific building distribution, scenario **B4** considers that the location and size of buildings are randomly selected. In this scenario two mmWave base stations are deployed 100 m apart from each other. In parallel to the straight line connecting the two base stations, and at a distance of 70 m, users move from one base station to the other. Then, in each scenario realization, 8 buildings of random sizes (with maximum side size of 20 m) are placed at random positions between the base stations and users.

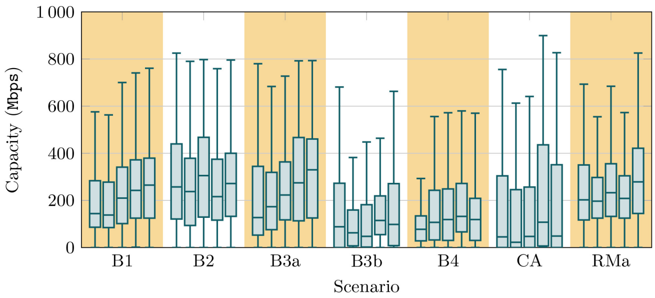


FIGURE 2. Boxplot of the capacity of the analyzed scenario. For each scenario, the plot includes the capacity variance of the 5 realizations.

TABLE 3. Statistics of the capacity of each scenario measured every 10 ms. In all cases the values have units of Mbps.

Scenario	min.	max.	std. dev.	avg.
B1	0.94	3703.79	193.94	244.60
B2	0.94	3592.62	217.77	286.79
B3a	1.18	4426.60	210.11	257.20
B3b	0.24	3862.53	196.31	141.85
B4	1.18	1885.81	175.13	364.71
CA	0.60	2158.54	264.61	184.92
UMa	0.59	4692.15	209.72	154.46

Besides, we have included an additional scenario (**CA**) where buildings are deployed according to particular positions. The plan view of the buildings replicates the main buildings of the NYU-AD campus, where some of them are very close to each other, while the principal one is more separated. The reason to add this scenario is to include a realistic layout that represents an existing location, rather than generic scenarios.

While in the previous cases we have considered the presence of buildings under different assumptions, in the last scenario, **UMa**, we simply apply the 3GPP propagation model for urban macro scenarios (without any buildings in the simulation scenario). The model defines channel losses of urban areas where the base station antenna is deployed on rooftops. The base station height ranges from 10 to 50m, whereas the user terminal height varies from 1.5 to 22.5m. In this case, users move following a random walk over a square of $30 \times 30\text{m}^2$, located at 25m away from the base station.

IV. PERFORMANCE EVALUATION

This section presents the performance evaluation of QUIC and TCP over the mmWave channels obtained from the scenarios described in the previous section. We will use the following performance indicators: throughput, packet delay and occupancy of the bottleneck link buffer. Apart from the channel trace, in the evaluation we will also modify the size of the bottleneck buffer and its delay.

In order to understand the behavior of both protocols, we will first characterize the used channels in terms of capacity. Later on, we will showcase the general differences in the response of TCP and QUIC in the analyzed scenarios. Then, we will compare the statistical behavior of both protocols when varying the bottleneck buffer, and link delay. Finally, we will study the design differences between both protocols, in terms of loss detection and congestion control mechanisms, which explain the observed performance discrepancies. Unless otherwise indicated, both QUIC and TCP use Cubic as their congestion control algorithm.

A. SCENARIOS CHARACTERIZATION

For each scenario described in Section III we generate 5 independent traces of 180 seconds, whose capacity variation is depicted in Figure 2. It shows boxplots of the capacity samples of each trace, where the box limits indicate the 25 and 75% percentiles and the median (50% percentile) is represented by the line within the box. In addition, the lower and upper whiskers indicate the 5% and 95% percentiles of the samples. In all cases, capacity samples are obtained with time slots of 10 ms. As can be observed, the traces show a high variability, exhibiting capacities of hundreds of Mbps with high probability.

In order to better quantify the channel variability, in Table 3 we indicate the capacity limits found in each scenario, as well as the average and the standard deviation. In this case, the values are averaged over the 5 traces generated for each scenario. In general, we observe that the capacity ranges from hardly 1 Mbps up to a few Gbps. In this sense, it is worth noting that, while the maximum capacity reaches Gbps, the average capacity is one order of magnitude lower (some Mbps). In addition, the statistical distribution of the capacity exhibits a standard deviation in the same order of the average value, evincing the high variability of the capacity for all channels. In the following we will configure the application sending rate so that it equals the average capacity of the corresponding scenario.

B. TEMPORAL BEHAVIOR

In Figure 3 we represent the temporal evolution of different performance metrics when using QUIC and TCP over one channel trace. In particular, we use one trace of the scenario **B1** with different link delays (5, 20 and 100 ms) and we set the bottleneck buffer size to 1 BDP. The BDP is calculated as the average channel capacity multiplied by the RTT. In all cases, we represent the channel capacity with a shaded background, whose units are indicated in the right-hand ordinate axis. This capacity is calculated with 100 **ms** granularity for the sake of visibility.

Figures 3(a) and 3(b) show the throughput obtained with QUIC and TCP, respectively. As can be observed, in both cases the throughput achieved is far below the average capacity (244.6 Mbps). Although a higher throughput could be achieved using larger bottleneck buffers, it would benefit both protocols. Comparing the protocols, we can observe that TCP exhibits the same behavior regardless of the configured delay. On the other hand, QUIC shows a better performance, which is specially relevant for low link delays. Interestingly, when configured with the longer delay (100 ms in red

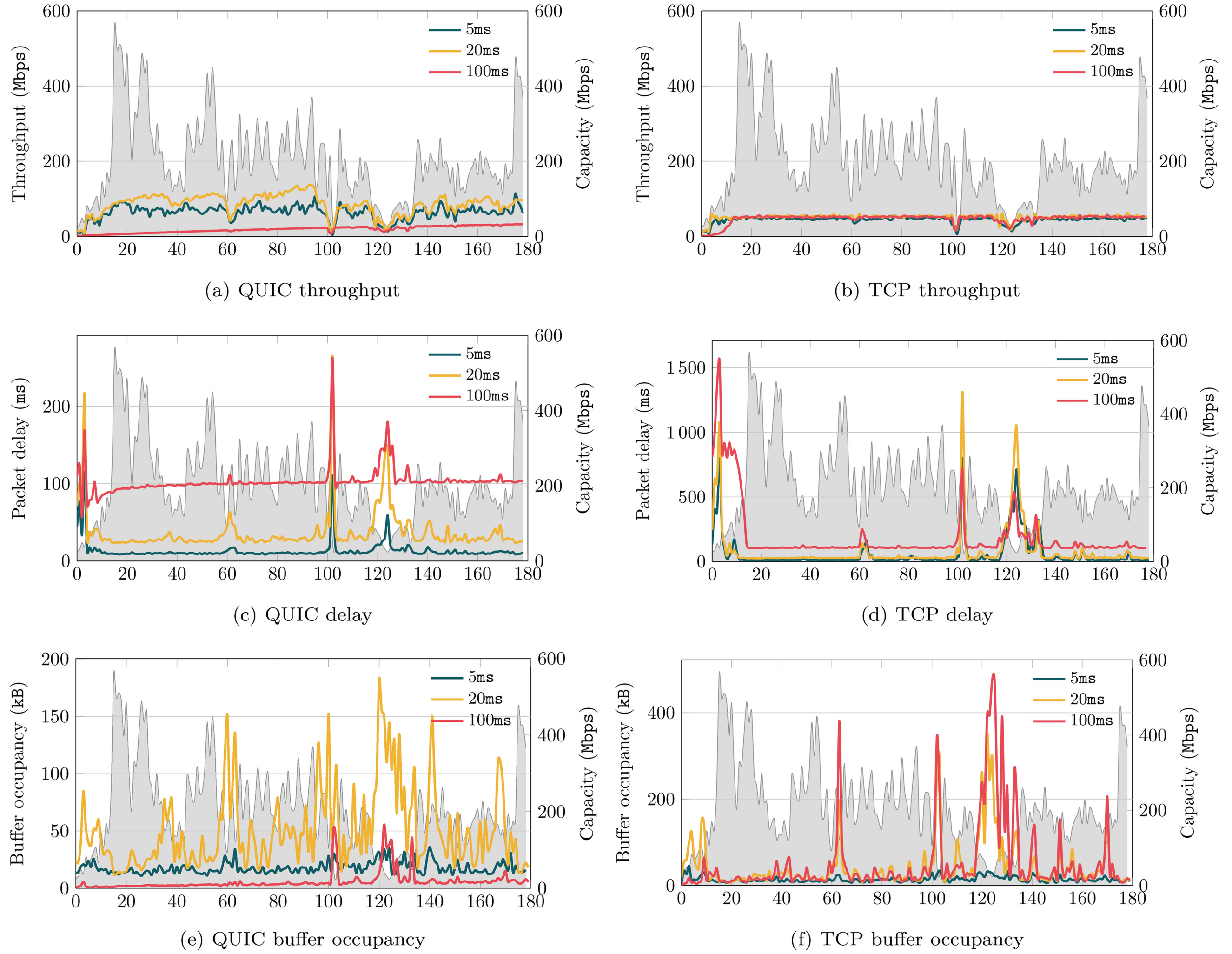


FIGURE 3. Example of performance comparison between QUIC and TCP in scenario B1 for different bottleneck delays. It includes throughput, packet delay and bottleneck buffer occupancy. In all cases the bottleneck buffer size is 1 BDP.

lines) TCP outperforms QUIC. In fact, QUIC presents a growing throughput over time, and we can observe that it takes several seconds to reach a stable level, even when it is below the one observed for TCP. The main reason behind the different behaviors is the design differences in the congestion detection implemented in QUIC, as we will explain below.

We then represent the packet delay in Figures 3(c) and 3(d). It is measured as the time elapsed since the packet leaves the sending application until it reaches the receiving one. It can be observed that, in general, the delay experienced when using both protocols is in the same range. However, when delay peaks appear (probably as a consequence of rapid channel capacity reduction), TCP induces packet delays one order of magnitude higher. Once again, this would be a consequence of the lack of adaptation to the channel variation, which would make TCP to continue sending traffic, leading to buffer-bloat and loss events. Finally, Figures 3(e) and 3(f) depict the temporal evolution of the bottleneck buffer occupancy. According to the behavior observed in terms of delay, TCP generates much higher peaks of the buffer occupancy. It is worth noting that, although these results belong to one single trace, similar differences between QUIC and TCP are also observed in other traces.

C. IMPACT OF BOTTLENECK LINK

In this section we analyze the statistical behavior brought about by QUIC and TCP upon varying the bottleneck buffer size and delay.

In Figure 4 we depict the boxplot of packet delay and throughput obtained by QUIC and TCP in each scenario, for two different sizes of the bottleneck buffer: 1 and 5 BDP. As mentioned above, the BDP is calculated by multiplying the average trace capacity by the configured bottleneck link delay. For the sake of visibility, the results combine the values obtained with the 5 independent traces of each scenario. As can be observed in Figure 4(a), the delay experienced when using QUIC is consistently lower than that observed for TCP. This difference is specially relevant in the **CA** scenario, which presents the highest standard deviation of the capacity, as was shown in Table 3. In addition, we

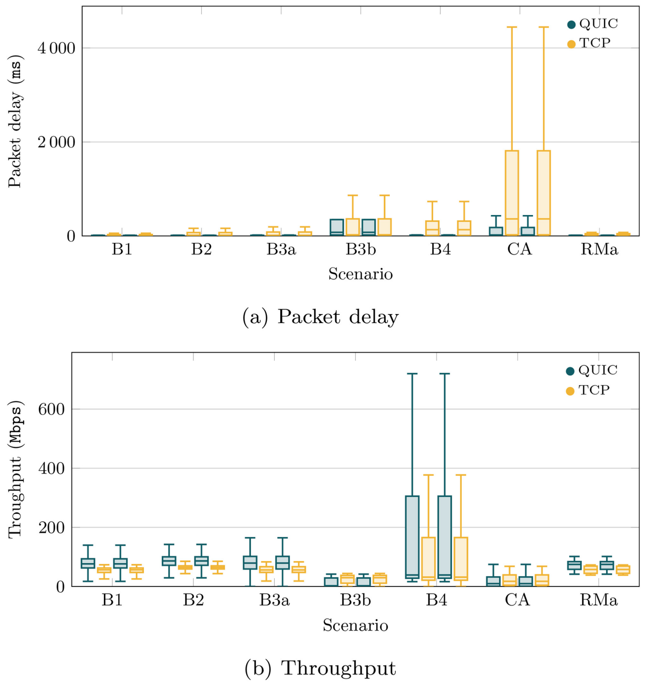


FIGURE 4. Impact of the bottleneck link buffer size over the protocol performance. For each scenario, we depict the boxplot for 1 (left) and 5 (right) BDPs. In all cases the bottleneck link delay is set to 5 ms.

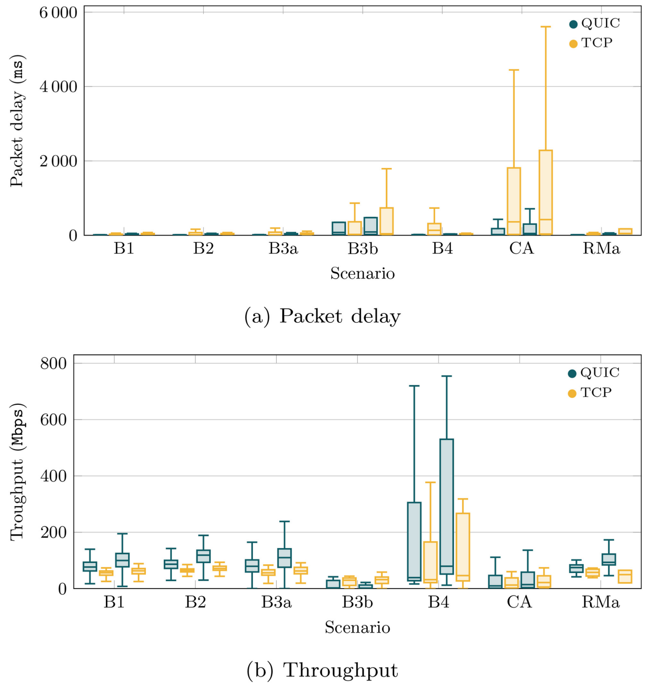


FIGURE 5. Impact of the bottleneck link delay over the protocol performance. For each scenario, we depict the boxplot for 5 (left) and 20 (right) ms. In all cases the bottleneck link buffer size is set to 1 BDP.

can observe that the bottleneck buffer size does not impact the delay. Similarly, Figure 4(b) evinces that the throughput achieved by QUIC is always above the values observed for TCP. In this case, the main difference appears in the scenario with the highest average capacity, **B4**, where TCP hardly achieves peak rates of 400 Mbps, while QUIC throughput reaches more than 700 Mbps. This behavior is consistent with the observations in the previous section, where QUIC was able to better exploit the channel capacity, inducing lower delays.

In Figure 5 we represent the same performance metrics (packet delay and throughput) for two different values of the link delay for all the scenarios. As expected, Figure 5(a) evinces that the delay experienced by both protocols increases with the link delay. However, we can observe that the delay configuration does not have the same impact over both protocols in terms of throughput. Indeed, Figure 5(b) shows that QUIC's throughput exhibits a slight growing trend when shifting from 5 ms to 20 ms delay; the same trend was shown in Figure 3(a). This behavior is specially relevant in the **B4** scenario, where the 75% percentile (upper box limit) of QUIC's throughput goes from 300 to more than 500 Mbps. On the other hand, the impact of slightly higher link delays over TCP is not clear. In many cases the results show that there is no impact whatsoever, while in others it seems to slightly jeopardize TCP's performance.

Altogether, the results reveal that QUIC exhibits a general better performance over highly varying mmWave channels, both in terms of throughput and delay. While a higher throughput could be achieved by saturating the channel, rather than limiting the traffic rate to match the average channel capacity, the fact that QUIC yields lower delay indicates that it adapts better to this type of channels. In the following section we will explore in more detail the internal differences between TCP and QUIC, to shed light on the observed performance discrepancies.

D. ANALYSIS OF BEHAVIOR DIFFERENCES

Some of the reasons behind the behavior differences that we discussed above are the improvements in terms of loss detection and congestion detection mechanisms included in QUIC [15, Sec. 4]. In Table 4 we enumerate such implementation differences, briefly explaining their meaning and discussing their potential impact over the protocol performance in mmWave scenarios.

As can be observed, there are not simple ways to analyze the impact of each difference in a separate manner, since in many cases they bring fundamental changes in the protocol implementation. Along with the mentioned differences, the QUIC specification [15, Sec. VII.8] indicates that when the congestion window is underutilized (bytes in flight are lower than the congestion window) the congestion window should not be increased when either in slow start or congestion avoidance.

In order to study whether the differences mentioned above have an impact on the congestion control, we have recorded the evolution of the congestion control for the **B1** scenario using the same setup as in Figure 6. In the case of TCP the congestion window is obtained from the application,

TABLE 4. Differences between TCP and QUIC.

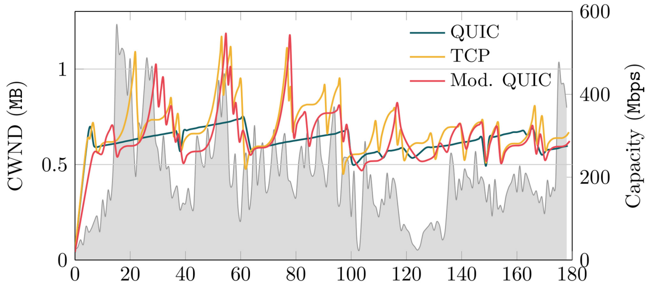
Difference	Meaning	Impact on our analysis
Separate packet number space	Different numbering for each encryption level	It has no impact, since we send a single stream and single encryption level
Monotonically increasing packet number	In removes ambiguity regarding the packet acknowledged by an ACK, and it permits more accurate RTT estimation. TCP implements the Karn's algorithm to avoid ambiguity, but it may lead to discarding some samples to calculate the RTT.	It may have a performance effect, due to different estimates of the RTT.
Clearer loss epoch	QUIC starts a loss epoch when a packet is lost, as it happens with every round trip of a burst. TCP will start the loss epoch for the first loss, even if more tries are also lost. As congestion control is reduced every loss epoch, QUIC does it more times.	It may help QUIC to avoid saturating the bottleneck buffer, thus reducing loss events if there is buffer bloating during periods of low capacity. The effect of this difference can be seen by tracking the bottleneck buffer occupancy, RTT estimate and CW size.
No reneging [50]	By default QUICs ACKs are similar to TCP SACKs. In addition QUIC does not allow neglecting (reneging) ACKs	We do not think it has an impact on the analysis we carried out, since there would be no reason for discarding received data, and it is in any case a rare phenomenon.
More ACK ranges	QUIC supports more than the 3 SACK ranges of TCP, thus accelerating loss recovery	It may have an impact considering the high volatility of the mmWave channel capacity: the high variation provokes rapid buffer bloating, leading to dis-contiguous loss bursts. If there are more than 3 bursts, TCP SACK cannot indicate them in one ACK, while QUIC can.
Probe Timeout replaces RTO and TLP	QUIC does not collapse congestion window when the RTO expires, but only with persistent congestion (loss of all packets during a period). TCP does the same if RACK-TLP algorithm is used.	As indicated in [15], this improvement makes a difference in application limited cases, which may happen in some cases.
Minimum congestion window is two packets	TCP sets it to 1 packet. If it is lost it needs to wait for the RTO expiration, which may be longer than the RTT. In addition, if the receiver delays the ACK of that packet, it leads to longer delays	It may have an effect if there is persistent congestion and the CC depletes the CW to start the fast recovery.
Handshake packets are not special	Losses of SYN/SYN-ACK in TCP are treated as persistent congestion, while QUIC treats them as any other packet	As we establish a single connection, we do not think this has an impact over the performance.

invoking the socket option **TCP_INFO** every time a packet is sent. On the other hand, for QUIC the congestion window is logged by modifying the source code of the corresponding implementation. Besides, QUIC results are obtained using both the regular QUIC operation, and later modifying it to disable the congestion window underutilization detection.

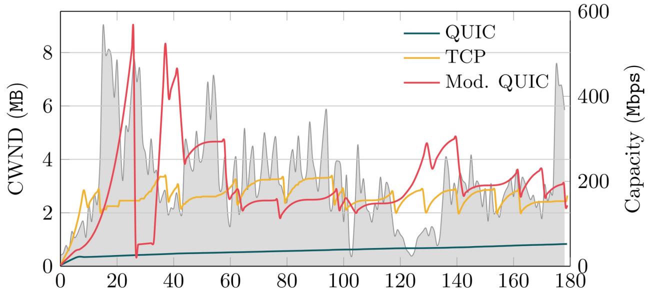
In Figure 6(a) we depict the evolution of the congestion window when both protocols use CUBIC and the bottleneck buffer is set to 1 BDP. Surprisingly, the congestion window with the regular QUIC operation does not exhibit the characteristic cubic shape. In fact, it seems to remain permanently in the congestion avoidance state, so that it grows linearly over time. By analyzing the code execution we have noticed that QUIC often detects window underutilization, leading to the observed behavior. Indeed, the behavior of the QUIC version without window underutilization detection (labeled as *Mod. QUIC*) is very similar to that yielded by TCP. In Figure 6(b) we show the congestion window evolution when the bottleneck buffer is set to 5 BDPs. As can be observed, the differences between QUIC and TCP are much more relevant, while the behavior of the modified QUIC version is rather alike to TCP.

Finally, Figure 6(c) shows the evolution of the congestion window when both protocols use RENO as congestion control algorithm. We can observe that the behavior of QUIC is again very similar to that exhibited when using CUBIC, due to the detection of window underutilization. On the other hand, TCP and the modified version of QUIC show a similar trend.

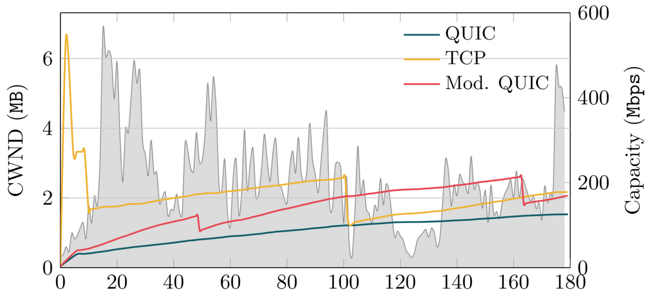
In addition, other analysis have been performed to study the impact of the receive buffers. However, the receiver application in our setup drains the buffer as soon as possible, so that the flow control is never triggered, neither in QUIC nor TCP. Altogether, we consider that the main reason for the different behaviors observed for TCP and QUIC is the capability of QUIC to detect situations when the congestion window is underutilized. As we have mentioned earlier, we configured the transmission rate of the application to the



(a) CUBIC with bottleneck buffer of 1 BDP



(b) Cubic of bottleneck buffer of 5 BDP



(c) RENO with bottleneck buffer of 5 BDP

FIGURE 6. Evolution of the congestion window in the B1 scenario. Results labeled as *Mod. QUIC* correspond to QUIC without detection of congestion window underutilization.

average capacity of the scenario over time. On the other hand, the rapid variation of the channel capacity (see Figure 1) may lead the congestion window to grow above the one required by the application. As a result of underutilized congestion window detection capacity, QUIC shows a more conservative approach, which avoids saturating the bottleneck link when the capacity decreases. In turn, it avoids buffer bloating effect, yielding lower delays.

V. CONCLUSION AND FUTURE WORK

5G cellular communications represent a highly promising telecommunication technology designed to meet the upcoming requirements for higher data rates and minimal delays. The adoption of the mmWave frequency band is an essential part of 5G networks, since it offers a larger capacity for delivering extensive data rates. However, the volatility of this type of channels in real setups gives rise to large capacity fluctuations, which hinder the performance of transport layer protocols.

In this paper, we compare and analyze the performance of the two most relevant transport protocols, TCP and QUIC, over different mmWave scenarios. For that, we first generate detailed capacity traces using the mmWave module of ns-3, which are afterwards used to feed the Mahimahi link emulator. This setup allows us to use real applications with capacity traces having a temporal resolution of 1 ms.

The performance comparison shows that QUIC exhibits a better adaptation to the mmWave channels. Indeed, we observe that it yields higher application throughput and lower packet delay. According to that, we also observe a lower occupancy of the bottleneck buffer when using QUIC.

In our future research we will extend this work in two different and complementary ways. First, we will investigate QUIC performance using different congestion control algorithms. On the other hand, we will analyze the benefits of the multi-streaming feature of QUIC over mmWave channels. In this sense, an appropriate stream scheduling may help to prioritize traffic flows, even when the link capacity is not fully exploited.

REFERENCES

[1] 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, 2019.
[2] B. Hindawi and A. S. Abbas, "Congestion control techniques in 5G mm wave networks: A review," in *Proc. 1st Babylon Int. Conf. Inf. Technol. Sci. (BICITS)*, 2021, pp. 305–310.
[3] M. Okano, Y. Hasegawa, K. Kanai, B. Wei, and J. Katto, "TCP throughput characteristics over 5G millimeterwave network in indoor train station," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2019, pp. 1–6.
[4] R. Poorzare and A. Calveras, "Open trends on TCP performance over urban 5G mmWave networks," in *Proc. 17th ACM Symp. Perform. Eval. Wireless Ad Hoc Sensor Ubiquitous Netw.*, 2020, pp. 85–92.
[5] M. Polese, R. Jana, and M. Zorzi, "TCP in 5G mmWave networks: Link level retransmissions and MP-TCP," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, 2017, pp. 343–348.
[6] V. D. Thai Hai, "Benefits of 5G millimeter-wave communication in IoT applications," Dept. Comput. Informat., Bournemouth Univ., Bournemouth, U.K., 2022.
[7] W. Yang, X. Zhou, W. Du, J. Sun, Y. Ren, and G. Xie, "A measurement study of TCP performance over 60GHz mmWave hybrid networks," in *Proc. IEEE 23rd Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2022, pp. 300–305.
[8] R. Poorzare and A. C. Augé, "How sufficient is TCP when deployed in 5G mmwave networks over the urban deployment?" *IEEE Access*, vol. 9, pp. 36342–36355, 2021.
[9] F. Siddiqui and Q. Chau, "Analysis of transport layer congestion control algorithms over 5G millimeter wave networks," in *Proc. 11th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2021, pp. 1–6.
[10] G. Lee, S. Choi, and S. Bahk, "Simulation study of TCP proxy in multi-connectivity enabled 5G mmWave network," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, 2019, pp. 865–869.
[11] Y. Ren, W. Yang, X. Zhou, H. Chen, and B. Liu, "A survey on TCP over mmWave," *Comput. Commun.*, vol. 171, pp. 80–88, Apr. 2021.
[12] M. Zhang et al., "Will TCP work in mmWave 5G cellular networks?" *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 65–71, Jan. 2019.
[13] A. Langley et al., "The QUIC transport protocol: Design and Internet-scale deployment," in *Proc. Conf. ACM Special Interest Group Data Communication—SIGCOMM*, vol. 17, 2017, pp. 183–196.
[14] J. Iyengar and M. Thomson, "QUIC: A UDP-based multiplexed and secure transport," IETF, Fremont, CA, USA, RFC 9000, May 2021. [Online]. Available: https://rfc-editor.org/rfc/rfc9000.txt

[15] J. Iyengar and I. Swett, "QUIC loss detection and congestion control," IETF, Fremont, CA, USA, RFC 9002, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9002
[16] C. Slezak, M. Zhang, M. Mezzavilla, and S. Rangan, "Understanding end-to-end effects of channel dynamics in millimeter wave 5G new radio," in *Proc. IEEE 19th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, 2018, pp. 1–5.
[17] R. Poorzare and A. C. Augé, "FB-TCP: A 5G mmWave friendly TCP for urban deployments," *IEEE Access*, vol. 9, pp. 82812–82832, 2021.
[18] M. Kim, S.-W. Ko, and S.-L. Kim, "Enhancing TCP end-to-end performance in millimeter-wave communications," in *Proc. IEEE 28th Annu. Int. Symp. Pers. Indoor Mobile Radio Commun. (PIMRC)*, 2017, pp. 1–5.
[19] 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)*, 2019, pp. 1–7.
[20] M. T. Hassan and M. M. Mowla, "Investigation of TCP performance in 5G mmWave networks," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, 2020, pp. 1688–1691.
[21] F. Abdulrazzak, E. Abdulaziz, and K. Al-Hussaini, "Performance analysis for TCP protocols over mm wave in 5G cellular networks," in *Proc. 1st Int. Conf. Intell. Comput. Eng. (ICOICE)*, 2019, pp. 1–6.
[22] T. Azzino, M. Drago, M. Polese, A. Zanella, and M. Zorzi, "X-TCP: A cross layer approach for TCP uplink flows in mmWave networks," in *Proc. 16th Annu. Mediterr. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, 2017, pp. 1–6.
[23] G.-H. Kim and Y.-Z. Cho, "mmS-TCP: Scalable TCP for improving throughput and fairness in 5G mmWave networks," *Sensors*, vol. 22, no. 10, p. 3609, 2022.
[24] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3584–3608, 4th Quart., 2019.
[25] K. Nepomuceno et al., "QUIC and TCP: A performance evaluation," in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, 2018, pp. 45–51.
[26] J. Jung and D. An, "Access latency reduction in the QUIC protocol based on communication history," *Electronics*, vol. 8, no. 10, p. 1204, 2019.
[27] F. Bulgarella, M. Cociglio, G. Fioccola, G. Marchetto, and R. Sisto, "Performance measurements of QUIC communications," in *Proc. Appl. Netw. Res. Workshop*, 2019, pp. 8–14.
[28] Y. Yu, M. Xu, and Y. Yang, "When QUIC meets TCP: An experimental study," in *Proc. IEEE 36th Int. Perform. Comput. Commun. Conf. (IPCCC)*, 2017, pp. 1–8.
[29] A. De Biasio, F. Chiariotti, M. Polese, A. Zanella, and M. Zorzi, "A QUIC implementation for NS-3," in *Proc. Workshop NS-3*, 2019, pp. 1–8.
[30] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove, "Taking a long look at QUIC: An approach for rigorous evaluation of rapidly evolving transport protocols," in *Proc. Internet Meas. Conf.*, 2017, pp. 290–303.
[31] R. Herrero, "Analysis of the constrained application protocol over quick UDP Internet connection transport," *Internet Things*, vol. 12, Dec. 2020, Art. no. 100328.
[32] E. Liri, P. K. Singh, A. B. Rabiah, K. Kar, K. Makhijani, and K. Ramakrishnan, "Robustness of IoT application protocols to network impairments," in *Proc. IEEE Int. Symp. Local Metropolitan Area Netw. (LANMAN)*, 2018, pp. 97–103.
[33] F. Fernández, M. Zverev, P. Garrido, J. R. Juárez, J. Bilbao, and R. Agüero, "And QUIC meets IoT: Performance assessment of MQTT over QUIC," in *Proc. 16th Int. Conf. Wireless Mobile Comput. Netw. Commun. (WiMob)*, 2020, pp. 1–6.
[34] F. Fernández, M. Zverev, P. Garrido, J. R. Juárez, J. Bilbao, and R. Agüero, "Even lower latency in IIoT: Evaluation of QUIC in Industrial IoT scenarios," *Sensors*, vol. 21, no. 17, p. 5737, 2021.
[35] S. Jeddou, F. Fernández, L. Diez, A. Baina, N. Abdallah, and R. Agüero, "Delay and energy consumption of MQTT over QUIC: An empirical characterization using commercial-off-the-shelf devices," *Sensors*, vol. 22, no. 10, p. 3694, 2022.
[36] P. Gärdborn, "Is QUIC a better choice than TCP in the 5G core network service based architecture?" M.S. thesis, School Elect. Eng. Comput. Sci., KTH Royal Inst. Technol., Stockholm, Sweden, 2020.
[37] D. Hasselquist, C. Lindström, N. Korzhitskii, N. Carlsson, and A. Gurtov, "Quic throughput and fairness over dual connectivity," *Comput. Netw.*, vol. 219, Dec. 2022, Art. no. 109431.
[38] G. Sinha, M. R. Kanagarathinam, S. R. Jayaseelan, and G. K. Choudhary, "CQUIC: Cross-layer QUIC for next generation mobile networks," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, 2020, pp. 1–8.
[39] H. Haile, K.-J. Grinnemo, P. Hurtig, and A. Brunstrom, "RBBR: A receiver-driven BBR in QUIC for low-latency in cellular networks," *IEEE Access*, vol. 10, pp. 18707–18719, 2022.
[40] H. Haile, K.-J. Grinnemo, S. Ferlin, P. Hurtig, and A. Brunstrom, "WIP: Leveraging QUIC for a receiver-driven BBR for cellular networks," in *Proc. IEEE 22nd Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2021, pp. 252–255.
[41] Z. Krämer, S. Molnár, M. Pieskä, and A. Mihály, "A lightweight performance enhancing proxy for evolved protocols and networks," in *Proc. IEEE 25th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, 2020, pp. 1–6.
[42] E. Khorov, A. Krasilov, M. Susloparov, and L. Kong, "Boosting TCP & QUIC performance in mmWave, terahertz, and lightwave wireless networks: A survey," *IEEE Commun. Surveys Tuts.*, early access, Aug. 3, 2023, doi: 10.1109/COMST.2023.3301820.
[43] H. Haile, K.-J. Grinnemo, S. Ferlin, P. Hurtig, and A. Brunstrom, "Performance of QUIC congestion control algorithms in 5G networks," in *Proc. ACM SIGCOMM Workshop 5G Beyond Netw. Meas. Model. Use Cases*, 2022, pp. 15–21.
[44] R. Netravali, A. Sivaraman, K. Winstein, S. Das, A. Goyal, and H. Balakrishnan, "Mahimahi: A lightweight toolkit for reproducible Web measurement," *SIGCOMM Comput. Commun. Rev.*, vol. 44, no. 4, pp. 129–130, Aug. 2014. [Online]. Available: https://doi.org/10.1145/2740070.2631455
[45] 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.*, 2015, pp. 283–290. [Online]. Available: http://doi.acm.org/10.1145/2811587.2811619
[46] R. Ford, M. Zhang, S. Dutta, M. Mezzavilla, S. Rangan, and M. Zorzi, "A framework for end-to-end evaluation of 5G mmWave cellular networks in NS-3," in *Proc. Workshop NS-3*, 2016, pp. 85–92. [Online]. Available: http://doi.acm.org/10.1145/2915371.2915380
[47] M. Mezzavilla et al., "End-to-end simulation of 5G mmWave networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2237–2263, 3rd Quart., 2018.
[48] M. Zhang, M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "NS-3 implementation of the 3GPP MIMO channel model for frequency spectrum above 6 GHz," in *Proc. Workshop NS-3*, 2017, pp. 71–78. [Online]. Available: http://doi.acm.org/10.1145/3067665.3067678
[49] S. Sun, G. R. MacCartney, and T. S. Rappaport, "A novel millimeter-wave channel simulator and applications for 5G wireless communications," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.
[50] N. Ekiz and P. D. Amer, "Transport layer reneging," *Comput. Commun.*, vol. 52, pp. 82–88, Oct. 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0140366414002035

SIDNA JEDDOU received the master's degree in electronics and telecommunication from Abdelmalek Essaâdi University, Tetouan, Morocco, in 2018. He is currently pursuing the Ph.D. degree in computer science and telecommunications with the National Institute of Posts and Telecommunications, Rabat, Morocco. His research interests focus on the Internet of Things, applications, and transport protocols.

LUIS DIEZ received the M.Sc. and Ph.D. degrees from the University of Cantabria in 2013 and 2018, respectively, where he is currently an Assistant Professor with the Communications Engineering Department. He has been involved in different international and industrial research projects. As for teaching, he has supervised 23 B.Sc. and M.Sc. Thesis, and he teaches in courses related to cellular networks, network dimensioning, and service management. His research focuses on future network architectures, resource management in wireless heterogeneous networks, and IoT solutions and services. He has published more than 40 scientific and technical papers in those areas. He has served as a TPC member and a reviewer in a number of international conferences and journals.

NAJID ABDELLAH received the M.Sc. degree in networking and communication systems and the Ph.D. degree in electronic engineering from ENSEEIHT, Toulouse, France. He has several years of research experience with ENSEEIHT, ENSTA. In 2000, he joined the National Institute of Posts and Telecommunications, Rabat, Morocco, as a Full Professor of Telecommunication Engineering, where he is the Chief of Communication Systems Department and the Cybersecurity Master Coordinator. He has devoted more than 22 years to teaching microwave engineering, wireless networking, network architectures, and network modeling courses, and directing research projects as the Head of AGNOX Team in Artificial Intelligence applied to wireless network performance analysis, wireless sensor networks, and cybersecurity.

AMINE BAINA received the computer engineering degree from the National Engineering School of Bourges, France, in 2005, and the Ph.D. degree in computer science in "Access Control for Critical Infrastructures" from the Laboratory of Systems Analysis and Architecture, Toulouse, in 2009. He is an Associate Professor with the National Institute of Posts and Telecommunications, Rabat, Morocco.

RAMÓN AGÜERO CALVO (Senior Member, IEEE) received the M.Sc. degree (First-Class Hons.) in telecommunications engineering from the University of Cantabria in 2001, and the Ph.D. degree (Hons) in 2008, where he is currently a Professor with the Communications Engineering Department. He has supervised six Ph.D. students and more than 80 B.Sc. and M.Sc. thesis. He is the main instructor in courses dealing with networks, and traffic modeling, both at B.Sc. and M.Sc. levels. Since 2016, he has been the Head of the IT Area (Deputy CIO) with the University of Cantabria. His research focuses on future network architectures, especially regarding the (wireless) access part of the network and its management. He has published more than 230 scientific papers in such areas and he is a regular TPC member and a reviewer on various related conferences and journals. He serves on the editorial board of IEEE COMMUNICATION LETTERS (Area Editor since 2023), IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, IEEE SYSTEMS JOURNAL, and *Wireless Networks* (Springer).