IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 31 January 2023, accepted 22 February 2023, date of publication 24 February 2023, date of current version 1 March 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3248953*

SURVEY

# Toward the Implementation of MPTCP Over mmWave 5G and Beyond: Analysis, Challenges, and Solutions

**REZA POORZARE, (Member, IEEE), AND OLIVER P. WALDHORST, (Member, IEEE)**
Data-Centric Software Systems (DSS) Research Group, Institute of Applied Research, Karlsruhe University of Applied Sciences, 76133 Karlsruhe, Germany
Corresponding author: Reza Poorzare (reza.poorzare@h-ka.de)

This work was supported by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK) under Project bwNET2020+.

**ABSTRACT** 5G and beyond 5G networks are going to be an inseparable part of human lives in the future. They will dominate any aspects of everyday activities from smart homes, remote surgery, and smart cities to autonomous driving, which demand high throughput through low latency end-to-end communication. Consequently, employing higher frequencies in millimeter-wave for the coming cellular networks will be inevitable. On the one hand, millimeter-wave can provide massive data rates; on the other hand, it suffers from some shortcomings such as blockage and misalignment that can occur because of the susceptible characteristic of the high frequencies. These flaws can mislead TCP in adjusting its sending rate efficiently and reduce the connection quality that a user discerns. Deploying Multipath TCP (MPTCP) is one of the schemes that can relieve the aforementioned issues and assist in utilizing the new generation mobile networks' high potential by leveraging its features in exploiting diverse paths in networks such as NR or LTE. A well-designed MPTCP can select the best path among the available ones, so it can enhance the perceived user experience. This paper fills the gap of having a comprehensive overview of MPTCP and its deployment over 5G and beyond 5G networks. It analyses millimeter-wave-based cellular networks, MPTCP procedures and parameters, state-of-the-art MPTCP congestion control mechanisms and schedulers, and the probable solutions that may enhance the protocol's functionality in cellular communication.

**INDEX TERMS** 5G, beyond 5G, millimeter-wave, TCP, MPTCP.

## I. INTRODUCTION

The evolution path of cellular networks is going in a way that 5G (Fifth Generation) will be the dominant cellular technology in the coming years replacing LTE (Long Term Evolution) globally by attracting USD 13.1 and 1.7 trillion investment by the USA and China, respectively [1]. As a sign of this, it can be mentioned that in the third quarter of 2022, 110 million new users started to use 5G, making the number reach 870 million by the end of the quarter. Moreover, at the end of 2022, this number will grow up to one billion. By 2028, the number of 5G users will reach five billion indicating the mentioned dominancy, and the most considerable monthly average mobile traffic will be for North America reaching 91 percent of its subscribers employing 5G [2]. Comparing the estimated 91 percent of 5G users in 2028, with 92 percent of LTE users in 2019 is another indicator that 5G will prepotent the previous cellular communication in the future.

All these numbers express the significance of the 5G network in shaping future communication and justify the high expectation from it [3]. In addition, for fulfilling the increasing demand for high data rates, employing the mmWave (millimeter-wave) spectrum is inevitable [4] in favor of bringing off the 3GPP (Third Generation Partnership Project) NR (New Radio) standardization features [5]. However, the coverage and blockage issues will be more intense as the frequency rises in the coming generations [6], [7]. These two well-known issues root in the intermittent character of the

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

mmWave spectrum and having a NLoS (Non-Line of Sight) state can cause the blockage issue. As a result, when a UE (User Equipment) and a gNB (next-generation Node Base), i.e., base station for 5G, can not see each other because of an existing hurdle on the way i.e., NLoS state, the communication bandwidth can be declined drastically, which deteriorates the throughput of the communication channel. This is in contrast to the LoS (Line of Sight) state where a UE and a base station can communicate without having any obstacles in between, using the channel bandwidth to its full potential [8], [9], [10], [11], [12], [13], [14]. Considering the coverage issue, as the frequency increases, the distance that it can travel declines, so simply leading to low coverage of mmWave [11], [15].

In current networks, the perceived quality for users highly relies on the performance of the transport layer's widely employed protocol, TCP (Transmission Control Protocol) [16]. TCP tries to guarantee the delivery of each packet based on its acknowledgment technique and based on the congestion control mechanism, the protocol strives to regulate the sending rate to avoid congestion and perform close to the network capacity. There has been a lot of research in the last decades to improve TCP for different network environments and scenarios [17], thus, various protocols, such as NewReno [18], Cubic [19], HighSpeed [20], Westwood [21], Westwood+ [22] Compound [23] and BBR (Bottleneck Bandwidth and Round-trip propagation time) [24], [25] appeared.

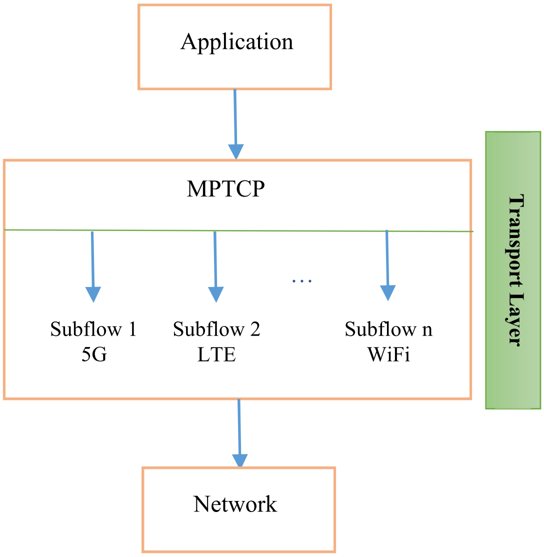

**FIGURE 1.** Transport layer during MPTCP deployment.

MPTCP (Multipath TCP) an extension of TCP, which was finally standardized in RFC 8684 [26], is gaining potent attention because of its new capabilities. Nowadays, smartphones are equipped with more than one NIC (Network Interface Card) and can utilize more than one network, such as WiFi or cellular connections. These features came to reality with the deployment of MPTCP initially in services such as Apple's Siri [27], [28], [29]. By exploiting MPTCP, the transport layer of the protocol stack would be like in Figure 1.

In this case, each interface will have its own IP (Internet Protocol) addresses and can transmit data independently. Conventional TCPs function based on creating a connection for each interface and then transmitting packets through it. In this case, if a connection tears down because of any reasons, for changing to another interface or re-establishing the connection, a 3-way handshake must be done from the scratch, and IP assignments should start over. On the other hand, MPTCP opens a connection for several subflows, then adds or removes subflows to the connection when it is feasible, then, for changing between the subflows, it does not need a new 3-way handshake. From this point forward, TCP means Single-Path TCP in this paper.

MPTCP mainly pursues three main goals:

- Goal 1 (Improve Throughput): The performance of MPTCP should be at least equal to the performance of TCP on the best path.
- Goal 2 (Do no harm): For fairness and Pareto-optimality, MPTCP should not utilize more resources on the shared path if it was a single path TCP. When a protocol is not Pareto-optimal, it may harm other users' performance without receiving any benefits [30].
- Goal 3 (Balance congestion): to adhere to resource pooling, MPTCP should do its best to offload the congested paths.

Resource pooling [31], which is one of the fundamental pillars of throughput enhancement, can be attained by goal three. In this case, individual MPTCPs try to transmit more data on their least congested path, resulting in emptying the congested paths. Besides, because MPTCP initiates more than one SF (SubFlow) in the same connection, it can overcome the HoL (Head-of-Line) issue, which occurs in TCP due to packets being blocked because of a dropped one on the connection.

### *A. CHALLENGES FOR MPTCP ON THE ROAD TOWARD mmWave 5G NETWORKS*

Considering mmWave frequency issues, i.e., blockage and low coverage, MPTCP can be a proper choice to overcome the mmWave cellular network flaws. If a UE can make use of more than one subflow at the same time in a connection, for example, 5G and another technology, if the 5G channel encounters difficulties in transmitting the data because of blockage or other shortcomings, employing the other connectivity would be beneficial, especially by preventing vertical and horizontal handovers that can occur and degrade the performance of the network [32]. Conventional TCP has been analyzed in wireless and cellular networks extensively, such as [9], [10], [11], [32], [33], and [34]. However, little research has been done on the compatibility of MPTCP over mmWave, mainly, over 5G mmWave. The lack of an adequate number of MPTCP analyses in 5G due to various reasons such as the absence of proper testbeds, complexity, and cost of prototyping [35], highlights the importance of our contribution to color the pave on the path of future research in pursuance of easing the way in designing novel MPTCPs for the future Internet. Despite the difficulties, there have been simulation analyses of MPTCP in 5G mmWave networks in [33] and [35]. First of all, the investigations indicated that MPTCP is not able to utilize the available bandwidth of mmWave to its full potential. Secondly, MPTCP is misled by the lossy environment of mmWave and chooses the LTE or other existing communication infrastructure mistakenly even when the mmWave channel is available. Thirdly, the existing fluctuations in mmWave such as blockage, confuse the MPTCP scheduler in doing the load balancing properly and it can not transmit packets based on the available free capacity of the links.

There have been some other works on the simultaneous use of 802.11ad's 60 GHz band and 802.11ac's 5 GHz frequency in [36], which can be guidelines for the MPTCP deployment in mmWave. The main focus of the efforts was on how MPTCP can deal with interference, mobility, and blockage issues occurring in mmWave intensified in smartphones due to their mobility characteristics [36], [37]. As a result, for the first time, MPTCP was imported to a mobile phone in [36] to figure out its functionality in mmWave and 5GHz bands when deployed simultaneously and the codes are available publicly for whom are interested in testing MPTCP on mobile phones. The results for the test utilizing iperf3 also revealed that MPTCP could not achieve the optimal value that was intended, which is another indicator that MPTCP comes across some issues when deployed in the mmWave band. Moreover, some other issues can reinforce the mentioned problems. Long-distance connections through weakening the cwnd (congestion window) adjustment process, asymmetry characteristics between mmWave and other channels leading to the HoL blocking issue, and random packet losses in wireless communication can be some of the intensifiers on the path of MPTCP deployment in mmWave networks [32] that should also be contemplated. By having a look on the mentioned issues, we have decided to deliver a survey that focuses on MPTCP functionality on cellular networks, especially 5G mmWave as a direction for the adaptation of MPTCP in the new cellular communication.

### B. RELATED SURVEYS

There have been some other surveys on MPTCP that aimed at various aspects of the protocol. A focus on load distribution and steering traffic into existing paths was done in [38] to classify different models by considering traffic splitting and path selection mechanisms that they employ. Investigation of MPTCP solutions that target the recording problem was done in [39] with a focus on bandwidth aggregation enhancement in heterogeneous wireless networks. In [40], the first focus was on the routing mechanisms on the Internet, then reviewing various multipath protocols based on a top-to-bottom protocol stack view by a look at different layers. Moreover, they could investigate multipathing operations from a mathematical view along with analyzing issues such as reliability and security.

As [40] had a top-to-bottom view, [41] concentrated on the multipathing solutions on the network layer analyzing the control plane and data plane issues. The survey investigated mechanisms of how a path should be selected, and then how the data should be splitted and transmitted to the selected paths. Another analysis of multipathing in different layers of the protocol stack was introduced in [42]. This work tried to classify and analyze multipathing through individual layers by determining the advantages and disadvantages of each view. A study of multipath TCP through its congestion control mechanism was done in [43] to cover different mechanisms and solutions for various circumstances by considering different performance features such as friendliness, load balancing, and Pareto optimality over different network scenarios. On the other hand, [44] investigated scheduling, which is another important component in MPTCP by categorizing it into single-criterion and multicriteria schedulers. In the former one, only one feature such as RTT is being used, while in the latter one more than one parameter is considered to choose the appropriate path.

**TABLE 1.** Related surveys.

| Ref. | Summary |
|---|---|
| [38] | Survey on distribution models, classifying of its various approaches, and investigation of traffic steering mechanisms |
| [39] | Survey on MPTCP mechanisms that tackle re-ordering issues in heterogeneous wireless networks with a focus on bandwidth aggregation mechanisms |
| [40] | Investigation of various multipath protocols based on a top-to-bottom protocol stack view by considering different layers |
| [41] | Concentrating on MPTCP solutions from the network layer perspective and issues on splitting traffic to different subflows |
| [42] | analyzing multipathing through different layers' perspectives and issues that can happen in different layers when MPTCP is deployed |
| [43] | Analyzing some of the existing MPTCP congestion control algorithms |
| [44] | Inspection of some of the existing MPTCP scheduler algorithms |

Deploying multipathing as an enabler for moving toward URLLC (Ultra-Reliable Low-Latency Communication) including 3GPP standards up to release 15 was done in [45]. Finally, a recent survey [46], focused on using multipathing as an enabler to achieve the goals of ATSSS (Access Traffic Steering, Switching, and Splitting) in using more than one channel at the same time based on release 16 standards [47]. Moreover, by having ATSSS enabled by MPTCP, using non-3GPP networks such as Wi-Fi along with 3GPP networks would be feasible [46], [48] and can ease the way of having an aggregation between 3GPP and non-3GPP ones to steer the traffic between different radio accesses to hone the user experience. Table 1 summarizes the related surveys.

Considering the ITU-R (International Telecommunication Union- Radiocommunication) category, there are five different scenarios including indoor hotspot-eMBB (Enhanced Mobile Broadband), dense urban-eMBB, rural-eMBB, urban macro-mMTC (massive Machine Type Communication), and urban macro-URLLC [49]. Our survey strives to analyze the use of MPTCP in cellular networks, more particularly in 5G mmWave, which is the foremost enabler for the urban and indoor scenarios along with the flaws that the spectrum exploitation can bring. Employing mmWave spectrum in 5G (and beyond 5G) and MPTCP practical usage is in their infancy and needs to be analyzed thoroughly until their final perfect application [32]. As a result, this work aims at filling the gap of having a comprehensive analysis of MPTCP and its deployment over 5G especially, 5G mmWave. As a result, the contributions of the paper include:

- A comprehensive analysis of MPTCP by having a detailed look at the most well-known variants and schedulers.
- An analysis of the state-of-the-art MPTCPs, their advantages, disadvantages, and compatibility with wireless and cellular communication with a particular focus on the eMBB scenario.
- An analysis of MPTCP's various congestion control and scheduling mechanisms to see whichever can be adapted to 5G and beyond 5G mmWave.
- An inspection of well-known 5G simulators' pros and cons to assist researchers in opting for the proper one.

In the first step, the fundamental of TCP will be briefly discussed, then comprehensive discussions on MPTCP's well-known congestion control and scheduling mechanisms with conclusions on their adjustability and adaptability to mmWave will be conducted. Finally, the compatibility of MPTCP over mmWave will thoroughly be investigated. As a result, the structure of the paper is as follows: Section two includes a fundamental explanation for TCP and its challenges over mmWave 5G, then section three brings an investigation of MPTCP and its history along with the evolution path. Section four provides a comprehensive analysis of the MPTCP's well-known congestion control algorithms and their impact on the performance of 5G mmWave networks and following it, section five will do the same on the various scheduler schemes. Section six brings an in-depth investigation of MPTCP and its compatibility with 5G mmWave. Section seven aims to provide investigations on the different 5G simulators by looking at their features. Finally, section eight concludes the paper.

## II. TCP EVOLUTION PATH ON ITS ADAPTATION TO mmWave 5G

This section focuses on the analysis of TCP. In the first part, a brief discussion of the TCP functionality has been brought, then the challenges of TCP over mmWave 5G have been discussed.

### A. FUNDAMENTALS OF TCP

The transport layer of the protocol stack has mainly two protocols. UDP (User Datagram Protocol), which is a connectionless protocol, and TCP, which is a connection-oriented one. Connectionless cannot guarantee the delivery and order of packets and is not able to adapt the sending rate to the available bandwidth. In contrast, connection-oriented protocols such as TCP strive to assure that individual packets are transferred successfully and in-order to the destination. Moreover, TCP provides congestion control to limit the amount of in-flight data bytes in the network to an upper bound of cwnd for the sake of preventing congestion.

TCP congestion control mechanism generally incorporates four phases: slow start, congestion avoidance, fast retransmit, and fast recovery. TCP triggers the slow start phase when a connection starts or is initialized. In this phase, for each received ACK (Acknowledgment), the size of cwnd is increased by one MSS (Maximum Segment Size); as a result, it doubles every RTT. Slow start aims to probe the network before entering the congestion avoidance phase. This phase is terminated when the slow start threshold is reached, or a packet drop is detected in the network [50]. The initial slow start cwnd normally equals ten segments in the Linux kernel [51].

After leaving the slow start phase, congestion avoidance is initiated. This phase is the heart of the congestion control mechanism and is different from one variant to another. NewReno AIMD (Additive increase multiplicative decrease) used to be the default approach in designing the other protocols' congestion avoidance. In this case, cwnd is increased by SMSS$*$SMSS/cwnd in every ACK, i.e., one full-sized segment per RTT, and is halved by noticing a packet loss through three duplicate ACKs or set to one when a RTO (Retransmission Time-Out) happens. SMMS (Sender Maximum Segment Size) refers to the sender's maximum segment size in the calculation. This method was appropriate for the initial wired networks where the only reason for a packet loss was congestion. However, this phase differed from one TCP to another by emerging new networks.

When three duplicate ACKs occur in the network, TCP takes it as a sign of loss. The reason is that this incident indicates that a single packet has been lost in the middle of a packet stream, so the receiver sends the same ACK more than three times. Given three duplicate ACKs (apart from the first ack), TCP does not wait for the RTO triggering, enters the fast retransmit phase, and resends the lost packet, thus can accelerate the process by preventing the cwnd initialization. After resending the packet successfully, the protocol returns to the congestion avoidance phase instead of the slow start; this transition phase is called fast recovery.

The congestion control mechanism included the four mentioned phases, which divide TCP mainly into three categories, including loss-based [19], [20], [52], [53], [54], [55], [56], such as NewReno, Cubic, High-Speed, delay-based [57], [58] like Vegas, or hybrid, which combine loss-delay-based algorithms, [24], [59], [60], [61], such as BBR and Compound. Each variant has been designed to fulfill the needs and requirements of a specific network or scenario.

### B. TCP CHALLENGES OVER mmWave 5G

To satisfy new demanding features and services, the coming mobile generations will use higher frequencies in the mmWave spectrum [62], [63]. These frequencies ease the way to have higher sending rates but suffer from some drawbacks because of the susceptible characteristics. As the frequency rises, the penetration power declines, and a reduced SINR (Signal-to-Interference-Plus-Noise Ratio), which degrades the network's performance, can lead to blockage. A blockage is a time that a UE and a gNB cannot make a proper connection because of existing hurdles such as buildings, cars, or even human bodies between them that are blocking the communication path [9], [32], [63] Besides blockage, a misalignment problem can happen whenever there is a mismatch between the transmitter and receiver beams [63]. These problems can mislead TCP in detecting the current situation of the network precisely as it cannot distinguish whether a drop is because of congestion or other flaws. Most of the time, it causes the congestion control mechanism to reduce the sending rate unnecessarily, leading to performance degradation.

The comprehensive analysis in [9] proved the inefficiency of TCP in urban deployments as there are numerous hurdles, and similar findings had been achieved in [11] for the high-speed scenario. Moreover, practical measurements such as [64] analyzed 5G mmWave networks in real infrastructures to attain almost identical results to the simulation scenarios. All the findings revealed that TCP could not adapt itself to the coming cellular networks.

One approach to handle the mentioned issues is exploiting new queuing techniques instead of the conventional drop-tail method. These techniques mostly focus on preventing buffer overflows beforehand to omit packet drops. For example, the findings in [12] proved that deploying some AQM (Active Queue Management) techniques such as CoDel (Controlled Delay) [65] and fq-CoDel (Fair Queuing Controlled Delay) [66] can enhance the performance of the protocol by preventing some packet drops. However, they need some modifications to be adapted to the new cellular networks [11]. Other schemes such as using Hybrid Automatic Repeat reQuest (HARQ) in the MAC (Medium Access Layer) layer of the protocol stack or RLC-AM (Radio Link Control-Acknowledged Mode) in the RLC layer can mask some packet drops from the TCP by resending the corrupted packets. However, this technique, called link-level retransmission [33], has some limitations and could not compensate for TCP flaws in mmWave.

Another proposed solution is deploying non-terrestrial networks to aid the mmWave, especially in 6G networks. This approach can provide connectivity for the time that a cellular channel is not available. Yet, it will force other issues, such as long delays, which contrasts with the URLLC use case of 5G [67].

One of the most paramount schemes that has attracted intense attention in recent years is using subsets of AI (Artificial Intelligence) such as machine learning [68] to improve the functionality of TCP over 5G mmWave networks. Briefly mentioning the state-of-the-art algorithms, FB-TCP (Fuzzy Based-TCP) [69] is a newly presented TCP that could achieve higher throughput and acceptable latencies than conventional TCPs in urban deployments using fuzzy rules. DL (Deep Learning) is another part of the AI that has been used in new TCPs such as DL-TCP (Deep-Learning-Based TCP) [63], which was suggested for disastrous situations when a UE and a gNB cannot establish a proper connection because of the collapsed buildings or trees.

The main focus of this work is on MPTCP, which is one of the potential candidates to be used in 5G mmWave and coming cellular communication to equip them with multipathing in order to employ more than a network at the same time. MPTCP's feasibility in using several network interfaces makes it possible for a device to connect to many networks, then steer and switch between them to increase performance and resiliency. As a result, in the following sections, various aspects of MPTCP by considering the aforementioned facts will be investigated to provide a clear insight into the protocol for being exploited in the new cellular generations.

## III. MPTCP's PAST, PRESENT, AND THE FUTURE

The working group for MPTCP was initially created within the IETF (Internet Engineering Task Force) in 2009 [70]. Then in RFC 6182 [71], the architecture for MPTCP was presented. Afterward, MPTCP coupled congestion control mechanism was introduced in RFC 6356 [30], and RFC 6824 [72] was for the experimental standardization of MPTCP. Eventually, the protocol was standardized in RFC 8684 [26].

Conventional TCPs suffer from various shortcomings that MPTCP can solve. For example, multihoming, which is the concept that aims at using several existing communication interfaces, can be enhanced by deploying MPTCP. Moreover, when more capacity is offered by existing additional paths, in contrast to TCP, MPTCP can make use of them and increase the available bandwidth among link failure tolerance. This can be worse when figured out that TCP cannot switch the communication path to a better one when available, for example, when the device enters an area of 5G coverage. All these issues accompanying new wireless technologies emergence are the motivations behind the increased will for the development of MPTCP. Ultimately, MPTCP can improve various aspects of TCP, such as friendliness, load-balancing, stability, and Pareto optimality.

For establishing the first subflow in a MPTCP connection, two ends exchange a MP-CAPABLE key during the conventional 3-way handshake, and if a node wishes to add a subflow, it exploits the MP-JOIN option. When a connection is established, MPTCP can use subflows for transmitting data [43].

### A. MPTCP AND mmWave COMPATIBILITY IN THE eMBB USE CASE

Considering the ITU-R defines five deployment scenarios, the main focus of this work is on the analysis of MPTCP over the urban and indoor eMBB, in which a high data rate is one of the most prominent factors. For providing the 20 Gbps peak data rate and 100 Mbps user experience in the downlink channel as the goals of 5G, the new cellular generations move toward employing the mmWave spectrum [62], so in this paper, 5G refers to 5G mmWave.

Mentioned issues such as blockage caused by frequent bandwidth degradation due to numerous existing hurdles in these scenarios, can be serious impediments to the way of MPTCP's proper functionality as they can mislead both congestion control and scheduling mechanisms of the protocol [35], [33], [73].

Delay increment is one of the first problems that the protocol encounters in a blocked 5G channel due to a sudden bandwidth reduction [32], [11]. As a result, the scheduling mechanism is a component that can intensely be misled and could not distinguish a blocked channel from a non-blocked one to make proper decisions. This issue itself can also drive out-of-order and buffer blockage in the receiver [74]. Plus the shadowing issue [75], in which the majority of users are active and movement is an inseparable part of their characteristics, fading can be intense in wireless and cellular communication. Both shadowing the former, and fading the latter, damage the performance of MPTCP over 5G mmWave networks.

On the other hand, the congestion control mechanisms of different MPTCP variants are not sufficient to adjust the sending rate in a way that can find proper upper bounds for the blocked and non-blocked situations [9], [11], [76]. Moreover, the protocol can suffer from bandwidth underutilization and unnecessary packet losses in the network due to LoS to NLoS transitions, which reduces the bandwidth of the 5G mmWave channels leading to a higher packet loss rate and latency [15]. Furthermore, the short coverage of mmWave due to its cells and high spectrum can be another issue for end-to-end protocols such as TCP [6].

To sum up, MPTCP can experience serious issues such as HoL, bufferbloating, receiver buffer blocking, and underutilization of the network resources of 5G and beyond 5G networks that intend to deploy higher frequencies to satisfy high data rates in urban deployments.

### B. MPTCP PIONEERING PATH, CHARACTERISTICS, AND APPLICATIONS

After detecting TCP's issues and moving toward MPTCP, the first idea was to deploy a TCP such as NewReno on each subflow independently, which was called uncoupled MPTCP. Although straightforward, such approaches would fail to accomplish the MPTCP goals stated in Section I. These kinds of protocols are severely unfair to TCPs, as they control the sending rate of their subflows independently without considering others and damage fairness in a way that could lead to starvation by dominating the resource utilizations [77]. Furthermore, they are not able to steer packets from congested paths to non-congested ones. As a result, Pareto optimally, fairness, and resource pooling can be damaged dramatically. Besides the aforementioned criteria, an MPTCP should be responsive to find proper upper bounds for individual subflows in various situations. Responsiveness is how fast a MPTCP can converge to subflows equilibrium by sending equal portions to various paths. Uncoupled MPTCPs are also unable to attain this goal, as individual subflows function independently.

All things considered, about uncoupled algorithms, a loss-based coupled MPTCP called LIA (Linked Increases Algorithm) was proposed in [30] as a starting point for coupled protocols emergence.

## IV. MPTCP CONGESTION CONTROLLING MECHANISMS WITHIN mmWave 5G

This section aims at providing a comprehensive analysis of well-known MPTCP congestion control variants, then tries to answer the question if they may be adaptable to mmWave 5G or not, principally, the eMBB scenario. Given the eMBB scenario, throughput and packet loss rate are important KPIs (Key Performance Indicators) for user experience improvement, however, some MPTCPs could hone other aspects such as latency, which does not seem a critical factor in the eMBB scenario at first glance. Nevertheless, in this case, we will indicate that the protocol is suitable for time-sensitive applications. However, more interestingly, improving latency and finding the fastest path can indeed aid the protocol to improve throughput as it reduces the assembly time on the receiver side by reducing issues such as HoL, buffer blocking, and out-of-order packet delivery.

Furthermore, adhering to goal two, coupled MPTCPs strive to be fairer, which is not crucial in the eMBB, on the other hand, following goal three, they try to be more aggressive, which can be one of the enablers of improving throughput. Finally, MPTCP can boost the tolerance for link failure circumstances, which by itself can ameliorate the throughput, because when a link becomes down, the other one can continue without any issue. This godsend is even at hand for MPTCPs that improve latency, thus, they can also somehow contribute through enhancements.

### A. LOSS-BASED CONGESTION CONTROL MPTCPs

In the first step, common loss-based congestion controls will be inspected, then the delay-based ones will follow. LIA, which is the earliest introduced coupled loss-based MPTCP is the first to come in the next sub-section.

#### 1) LINKED INCREASES ALGORITHM

As resource pooling is a critical aspect of MPTCP, LIA [30] strives to perform resource pooling by coupling the subflows to attain goals one and two, but not three. Moreover, it shows that congestion can be avoided by achieving resource pooling if packets are transmitted through low-loss rate subflows. LIA tries to keep the sum of aggregated bandwidth of subflows at least equal to its counterpart TCP on the best path, which is brought off by exploiting a parameter called $\alpha$. This parameter is deployed to modify the additive phase of the AIMD approach of the congestion avoidance shown in equation (1),

For each ack on subflow $r$:

$$wr = wr + min\{\frac{\alpha}{w_{total}}, \frac{1}{w_r}\} \tag{1}$$

In this equation, $w_r$ is the cwnd size on subflow $r$, and $w_{total}$ is the cwnd that MPTCP keeps for the connection. As LIA is pursuing the aim of attaining at least the minimum throughput of TCP on the best path in all circumstances, it is not feasible to choose a constant value for $\alpha$, so it is calculated based on formula (2):

$$\alpha = w_{total} * \frac{Max\left(\frac{w_r}{{rtt_r}^2}\right)}{(\sum_1^i (\frac{w_r}{rtt_r}))^2} \tag{2}$$

In this equation, $rtt_r$ is the round trip time for subflow $r$, $Max$ is for choosing the maximum available value, and $i$ is the number of subflow$s$. When a loss is detected due to three duplicate ACKs, LIA operates like NewReno by halving the cwnd without any modification.

If a MPTCP is unfriendly, it can lead to some drawbacks when sharing a path with a TCP. In this case, it can impair the functionality of TCP by occupying more resourcing. As a result, friendliness is an important factor for a MPTCP and by relying on this fact and deploying the aforementioned mechanism, LIA could achieve goals one and two. However, it could not attain goal three, which is resource pooling. The reason is that resource pooling means sending no traffic to congested paths, but LIA gradually decreases the traffic in routes with high congestion. If both increase and decrease adjustments become fully coupled, resource pooling could be achieved. Yet, it leads to another issue called flappiness, described in the following. When two subflows' congestion statuses are close, the protocol favors sending all packets through one subflow; then, after a while, it changes the subflows and transmits all data from another one, then this process continues. Flappiness can degrade the overall throughput, which is undesirable for MPTCP considering goal one. As a result, to prevent flappiness, LIA prefers not to perform resource pooling entirely, achieving only goals one and two.

Furthermore, to accomplish goal three's aims, a MPTCP should be more aggressive, meaning quickly finding the optimum equilibrium for the paths. However, this can not be the case for LIA as it is not aggressive enough because it gradually steers traffic between the paths.

To sum up the protocol's mechanism, LIA could resolve the fairness problem of uncoupled MPTCPs, achieve throughput at least equal to TCP on the best path, and somehow solve the resource pooling problem by increasing the traffic in non-congested paths and decreasing it in congested ones gradually, however, it could not attain the resource pooling ultimately. This protocol is a pure loss-based MPTCP, in which individual losses are the indicators for congestion in the network. It does not have any mechanisms to detect and differentiate various conditions such as congestion, random packet drops caused by channel quality degradation, blockage, or other 5G mmWave network issues. As a result, LIA cannot be a good choice to be exploited in wireless networks, let alone 5G mmWave ones. Because a single packet drop caused by blockage would force the protocol to back off, decrease its sending rate on the mmWave [9], and unload traffic from the mmWave channel, violating goal one.

Finally, if 5G mmWave is deployed with LTE and a temporary blockage happens in mmWave, due to abrupt buffer overflow, LIA assumes that this network is congested (it may not be) and initiates transmitting the packet over LTE as the non-congested path. After termination of the blockage state and switching from NLoS to LoS, it would take some time for LIA to gain the possible throughput of the mmWave channel with the assumption of not occurring another random packet drop on this path as LIA does not include a mechanism that routinely checks different subflows to see that whether a better route is available or not.

#### 2) OPPORTUNISTIC-LINKED INCREASES ALGORITHM

One of the main features of TCP is being Pareto-optimal. In this case, a user cannot increase its throughput without damaging others' benefits or forcing congestion costs on the network [78]. Authors in [79] could prove that LIA suffers from being non-Pareto-optimal, so when some users start to use MPTCP, they hurt the other's performance without incrementing their throughput. Moreover, they manifested that LIA is aggressive in using resources compared to TCP, damaging inter-fairness. Besides the aforementioned principal issues, it was also shown that LIA cannot satisfy the responsiveness and resource pooling at the same time and should come up with a trade-off between them. In order to compensate for the mentioned issues, a new MPTCP called OLIA (Opportunistic-Linked Increases Algorithm) [79] was proposed. OLIA strives to be Pareto-optimal, fairer to TCPs, responsive, and non-flappy. As a result, by considering these facts, OLIA is more friendly and aggressive than LIA in sharing the resources and steering traffic respectively.

Like LIA, OLIA is a coupled, loss-based MPTCP with no changes in its reaction to losses. By deploying a new increment phase in the congestion avoidance phase, OLIA strives to provide equal traffic to two similar paths. However, when the quality of the paths differs, the one with a lower loss rate and higher capacity gets more data to satisfy the load-balancing goal.

How OLIA does achieve its aims is as follows. If the set $R_u$ contains all available paths for the users, $r \in R_u$ will be a subflow. One of the OLIA's principal tasks is counting the successfully transmitted packet in two periods, between the recent two losses and since the last loss. In this case, $\gamma_{1r}(t)$ and $\gamma_{2\mathrm{r}}(\mathrm{t})$ can be for the first and second periods, respectively. Moreover, the maximum value of these two parameters can be shown as $\gamma_{\mathrm{r}}(\mathrm{t})$; in all parameters, $t$ refers to the time. By employing $\mathrm{rtt}_{\mathrm{r}}(\mathrm{t})$ and $w_r(t)$ that indicate RTT and cwnd for the subflow $r$ on time t, OLIA defines two sets called M(t) and B(t).

M(t) is the set of the best routes that their cwnds are set to the maximum value. However, B(t) includes the best paths that their cwnds are not set to the largest available value. The ultimate goal of OLIA is shifting traffic from $M$ to $B$, thus it adjusts the sending rate by exploiting Equation (3). For each received ACK, $\mathrm{w}_{\mathrm{r}}$ is increased:

$$w_r = \frac{{}^{w_r}\!/\!(rtt_r{}^2)}{(\sum_{pin\ R}^{1} {}^{w_p}\!/\!{}_{rtt_p})2} + \frac{\alpha_r}{w_r} \tag{3}$$

And for each loss, it halves the cwnd like NewReno and LIA. In the equation, how $\alpha_r$ is estimated for each subflow $r$ determines the aggressiveness of the sending rate increment. In this case, when the protocol finds some best subflows functioning in the most extensive available cwnd, i.e., members of $M$, it sets the value of $\alpha$ to zero for those routes, as they do not need any modifications. However, if OLIA detects that some subflows are the best ones but are not exploiting the largest cwnd, i.e., members of $B$, in this case, by making $\alpha$ positive for routes that are in $B$ not in $M$, and making $\alpha$ negative for $M$, strives to forward traffic from $M$ to $B$. As a result, subflows in B increase their sending rate faster. The output of this mechanism is that the sending rate increment will be more quickly on the best paths with smaller cwnds and slower on the best paths with larger cwnds leading to redirecting the traffic from the best ones with larger cwnds to the ones with smaller cwnds.

To sum up, it is true that some studies, such as [80], indicated that OLIA is always better than LIA in wireless networks. However, OLIA can suffer from severe issues if deployed in 5G mmWave networks. OLIA is a loss-based TCP, and individual losses are signs of congestion in the network. Yet, various causes such as blockage drops can occur in the 5G mmWave network, which may dramatically reduce OLIA's throughput. Moreover, as NewReno is not a good choice for 5G mmWave networks [9], [11], so protocols that adapt their congestion control mechanism from NewReno may not be suitable for 5G mmWave networks.

### 3) BALANCED LINKED ADAPTATION

Balia (Balanced linked adaptation) [81], another MPTCP, was proposed to bring a trade-off between MPTCP friendliness, responsiveness, and cwnd oscillation, to generalize LIA and OLIA. To accomplish its goal, Balia modified the increment and decrement segment of the congestion avoidance phase and did not follow the NewReno cwnd halving mechanism when it detects a loss in the network. If $w_r$ is the cwnd for the route $r$, and $rtt_r$ is the round-trip time of the route, for each ACK on route $r$ Balia functions as Equation (4):

$$w_r = w_r + \frac{x_r}{rtt_r(\sum rtt_k)^2}(\frac{1+\alpha_r}{2})(\frac{4+\alpha_r}{5}) \tag{4}$$

In this equation, $x_r$ equals $w_r/rtt_r$. When a loss occurs in the network, Balia reacts as Equation (5):

$$w_r = w_r - \frac{Wr}{2} min\{\alpha_r, 1.5\} \tag{5}$$

Which $\alpha$ equals $\frac{max\{x_k\}}{x}$.

The experiments indicated that Balia's friendliness is close to LIA, which shows that this protocol is not aggressive when deployed with TCPs. Considering the responsiveness, LIA and OLIA indicated poor performance. However, Balia, due to its renewed congestion avoidance phase, showed much better recovery time, which is a sign of being responsive. In terms of cwnd adjustment, Balia showed less fluctuation than Reno, which was another enhancement for the protocol. From the friendliness and aggressiveness point of view, as Balia could generalize LIA and OLIA, it could achieve better friendliness and aggressiveness, as proved in the simulation results in [32]. However, more practical experiments are needed to validate this outcome, as it seems the aggressiveness of OLIA can be as high as Balia based on the congestion control mechanism it adapts.

To sum up, Balia is a loss-based MPTCP that could offer better friendliness, responsiveness, and window oscillation compared to former MPTCPs. This protocol could show sufficient functionality in wired networks accompanying other MPTCPs and TCPs. However, to assess the compatibility of this protocol with 5G mmWave networks, we should look at Balia's congestion control mechanism, which is a loss-based one. Moreover, it increases the sending rate linearly and in a slow manner, which is another negative point for being deployed in 5G mmWave networks. Finally, when a loss is detected, multiplicative decrement is not an appropriate approach in networks that incorporate random packet losses rooted in issues such as a blockage.

### 4) DYNAMIC LIA

As LIA, OLIA, and Balia's main focus is on the increment part of the congestion avoidance, and LIA and OLIA employ NewReno's halving technique when a loss is detected, D-LIA (Dynamic LIA) [82] focuses on the decrement. The idea behind D-LIA's approach says that the time interval between packet drops is a good indicator of a network's status and halving the sending rate frequently in many loss occurrences degrades the protocol's performance, and it takes much time to recover from the backed-off cwnd. As a result, instead of halving cwnd in individual losses, D-LIA decreases the sending rate by a factor called $\beta$. The leading player specifying the value for $\beta$ is the passed time between the packet losses. As a consequence, for each packet drop equation (6) is used to adjust the cwnd in subflow $r$:

$$w_r = max\{\beta w_r, 1\} \tag{6}$$

If $\beta w_r$ is less than one, $w_r$ sets to one; otherwise, $w_r$ is adjusted to $\beta w_r$. For calculating $\beta$, D-LIA exploits equation (7):

$$Br = (\varphi \times \gamma) + (1 - \varphi) \times \beta r' \tag{7}$$

where $\beta_r'$ indicates the estimated $\beta_r$ in the most recent packet drop, $\varphi$ is a value between zero and one for balancing and is called the balancing factor, and the result of dividing cwnd between the last drop and the current one is put in $\gamma$, which is calculated by using equation (8):

$$\gamma = min\{\frac{w_r'}{w_r}, 1\} \tag{8}$$

where $w_r'$ is the value of the subflow cwnd in the last drop. A large value for $\gamma$ toward one indicates frequent losses in the network and can be taken as a congestion sign, so cwnd should be reduced dramatically. The reason is that large $\gamma$ shows that the time interval between the losses is shortening. On the other hand, a small $\gamma$ is a sign of few losses in the network; as a result, cwnd needs to be reduced negligently. For the increment phase of the congestion avoidance, D-LIA employ's the Linux default implementation (LIA). Simulation results indicated that D-LIA could outperform LIA in terms of throughput.

The analyses done in [82] are not sufficient enough to prove the advancement of the protocol and more investigation should be done in order to determine whether D-LIA is able to bring any advantages regarding important KPIs such as throughput, friendliness, or aggressiveness or not. As the protocol sticks to the main principles of LIA, it may not be more friendly and aggressive than LIA, OLIA, and Balia.

To sum up, D-LIA is a loss-based MPTCP that was proposed to enhance the performance of LIA. As a result, it inherits the drawback of LIA in wireless, especially 5G mmWave networks. Moreover, the simulation and comparison part for the protocol is between D-LIA and LIA, so more analyses are needed to prove the improved functionality of the protocol.

### 5) DELAYED-ADAPTIVE LIA

Moving forward, a thorough analytical investigation of MPTCP has been done in [83], then a novel coupled protocol called DALIA (Delayed-Adaptive LIA) was proposed to handle LIA's issues. One of the main difficulties that the majority of MPTCP variants face is when simultaneously deployed in Wi-Fi and cellular communication. In this case, packet losses and packet reordering can be two of the main issues. Furthermore, another crucial difference is that in cellular networks, individual flows can employ distinct buffers, however, WiFi flows must share the same buffer in the same AP (Access Point).

Moreover, using large buffers in cellular networks is expected, which can lead to bufferbloating and intensifies the packet reordering issue for MPTCP when deployed along with WiFi. Consequently, the authors in [83] presented a general analytic model that includes all network parameters to model it, parameters such as packet loss probability, short-lived flow behavior, reordering delay, buffer loss probability, RTTs, throughput, and loss computation. This model can indicate the behavior of the network and MPTCP, as a result, is an appropriate guideline for designing new protocols. Accordingly, they used the model to propose their novel protocol.

One of the most critical problems for LIA is that it cannot estimate delay variation in different subflows; as a result, it is not compatible with heterogeneous networks, and delay variation can impact the LIA's performance adversely. The proposed solution of DALIA is to add a route delay learning mechanism to LIA in a way that the new protocol could balance delays in different subflows and enhance the LIA's functionality. This feature can also lead to another improvement, which is reducing the reordering delay.

The first leverage of DALIA is exploiting the time stamp that the TCP packet includes. This timestamp is used in a TCP destination to calculate the one-way delay, then is returned to the sender in the ACK packet, inspired by LEDBAT [84]. This parameter can assist DALIA in attaining its goal, i.e., equipping LIA with the route delay learning mechanism. Then, DALIA changes the cwnd adjustment of LIA to control the sent packets to different routes. If $\mathrm{w_r}$ is cwnd for subflow $r$, $\mathrm{D_i}$ is the one-way delay of subflow $r$, and $\mathrm{rtt_r}$ is the RTT value for subflow $r$, then:

For each ACK, DALIA increases the sending rate by using equation (9):

$$wr = wr + min\{\frac{1}{w_r}, \frac{\sum jD_j}{MD_j}\frac{Max_j\left(\frac{w_j}{{rtt_j}^2}\right)}{{\sum j(\frac{w_j}{rtt_r})}^2}\} \tag{9}$$

If the minimum equals $1/w_i$, the mechanism will be the same as NewReno. Moreover, if the one-way delay is equally distributed in all paths, i.e., M (the obtained equal delay in all paths), the cwnd adjustment will be identical to LIA. The added factor to the mechanisms of LIA by DALIA, i.e., $\frac{\sum jD_j}{MD_j}$, is for equilibrating delays and congestion in different subflows leading to robust mechanisms in enhancing the friendliness and aggressiveness of the protocol. As a result, This mechanism ensures that all paths in a heterogenous network get enough traffic, and the delay can be improved, fulfilling MPTCP design aims. The importance of this mechanism can be identified when two different networks with distinct characteristics co-exist, such as cellular and WiFi. In this case, the cellular one is an impairing element for the WiFi, however, DALIA reveals this issue by tracing subflows' delays. The simulation results proved the correctness of the analytical model. Moreover, they indicated that DALIA is more responsive and friendlier than LIA and OLIA.

To sum up, DALIA can keep the benefits of LIA by adding a new factor to the congestion avoidance phase in a way that the protocol can balance the delays between various subflows and prevent performance degradation in heterogeneous networks, especially when cellular and WiFi networks are deployed simultaneously. Based on the results, this protocol can function appropriately in generations before 5G as it can equalize delays between different subflows. However, as the protocol keeps the loss-based nature of LIA and halves the sending rate in every packet loss, it may not be able to react appropriately to 5G mmWave network issues such as blockage. To make the protocol compatible with the new cellular generations, the congestion avoidance phase of the protocol should be modified. This modification first should start with the reaction of the protocol to losses. Then, the protocol should be more aggressive in the increment phase to adapt to the huge bandwidth of the 5G mmWave networks.

#### 6) BDP ESTIMATION BASED SLOW START

Authors in [85] claimed that the reason for MPTCP performance degradation in 5G mmWave networks is because of the long-lived period of the protocol in the slow start phase. As a result, they proposed that modifying the slow start phase is a necessary step that should be taken. The main cause is that before happening a packet loss, the protocol is in the slow start phase and doubles its sending rate, which leads to a high number of packets in flight. If one of the communication channels encounters an issue, the arrived packet from the other one will be blocked in the receiver's buffer waiting for the other path. For handling the issue of the throughput reduction caused by the 5G mmWave shortcoming, BESS (BDP Estimation based Slow Start) [85] was proposed.

When the connection is in the slow start phase, BESS approximates the BDP (Bandwidth Delay Product) for all subflows. To calculate BDP, the capacity of a link in bits per second is multiplied by the delay of the link. This value indicates the maximum possible bits (of bytes if calculated in bytes) that can be transmitted in a particular time.

Then, the approximated value is compared with the in-flight packets to whether they end the slow start or not, in another word, this value is a comparison base with the transmitted packets to have a new mechanism for terminating the slow start phase. As a result, the main leverage of BESS relies on the output of the BDP and inflight packet comparison. BESS calculates BDP by employing the bandwidth and delays of the existing subflows. However, it is not like the conventional TCP, and the multiplication of these values cannot give an accurate value for the BDP, so a unified value for delays is captured and deployed. This is the principal reason behind the dependency of the MPTCP's performance on the subflow with the largest delay. The next step is comparing the calculated value with the inflight packet, and when a subflow is getting close to the number, it should leave the slow start phase and initiate the congestion avoidance. As the congestion avoidance phase for BESS is adapted from Balia, this protocol inherits most characteristics of its predecessor such as the level of friendliness and aggressiveness.

For simulating the behavior of BESS, NS3-mmWave [86] was used by the co-existence of 5G mmWave and LTE, and the renewed slow start has been embedded into Balia as the representative of MPTCP, then the regular slow start in TCP, Balia, and CUBIC was used for the comparison. The result showed that when the delay of mmWave is smaller than 50 ms, BESS has a clear enhancement compared to other protocols. However, when the delay goes over this value, all protocols experience throughput degradation.

Looking at the receiver's buffer revealed that BESS occupies less space compared to the other two protocols so that it can handle the out-of-order packet issues sufficiently. For analyzing this KPI, the mmWave delay had been set to 1 ms, while the one for LTE was varying. The main reason is that BESS prevents the cwnd overshooting in the slow start, so when the protocol initiates the congestion avoidance, the buffers do not encounter a flood of backlogged packets. In terms of the UE's and eNB's (E-UTRAN Node B) distance, all protocols suffer from a high distance, however, BESS continues to have the highest throughput when the distance differs from 100 m to 150 m. Finally, when a transition from LoS to NLoS occurs, BESS also could attain the highest throughput.

To sum up, BESS was proposed to modify the slow start mechanism in MPTCP to adapt the protocol to 5G mmWave characteristics. The results showed that when this technique is embedded in Balia, it could enhance the protocol's performance in 5G mmWave co-existing with LTE, and proved the compatibility of BESS with the new mobile generation. For the next step, analyzing the protocol embedded in other MPTCPs and practical environments can be a helpful guideline for future work.

### *B. DELAY-BASED CONGESTION CONTROL MPTCPs*

As the loss-based MPTCPs such as LIA, OLIA, and Balia fail to accomplish MPTCP's three goals in heterogeneous networks, delay-based MPTCPs started to be probed. In contrast to loss-based TCPs that take a loss as an indicator of network congestion, delay-based TCPs strive to control the queuing packets and prevent drops by employing some proactive mechanisms using features such as link delay [87]. Moreover, blind mechanisms that loss-based MPTCPs deploy in increasing the sending rate until a packet loss occurrence may result in a bufferbloat [88] issue as large RLC buffers are used in cellular networks. Putting the aforementioned issues besides the reordering delay problem existing in networks, especially heterogeneous ones for MPTCP, proves the necessity of establishing delay-based MPTCPs that have a clearer insight from the network and adjust their sending rate more appropriately than loss-based ones.

### 1) WEIGHTED VEGAS

One of the first delay-based protocols proposed to extend MPTCP to delay-based congestion control areas was wVegas (weighted Vegas) [89]. The main motivation behind designing wVegas was that packet losses in a network are indications of heavy congestion, especially in wired networks, and cannot reflect light-congested situations, so wVegas suggested triggering load balancing by performing traffic shifting before loss occurrences to replace the reactive mechanism by a proactive one. To do so, a connection should specify the volume of traffic to be shifted from one of its subflows to another, leading to the congestion equality principle. The congestion equality principle refers to a situation where all subflows of a connection can get an equal volume of traffic, leading to fairness and enhancing the protocol's performance. As a result, wVegas, which is based on TCP Vegas [57], emerged.

The way that wVegas strives to achieve the congestion equality principle is by controlling the aggressiveness of individual subflows by assigning them a variable called weight, and as the weight for a path gets higher, it will compete more aggressively for utilizing the available bandwidth. The value for a weight inversely correlates to the extent to which the corresponding subflow is congested, so as the traffic volume for a path declines, the related weight will increase. The central claim of wVegas is that the queuing delays can reflect the network's ongoing condition better than packet losses, which were borrowed from the original Vegas. By relying on this mechanism, wVegas can be more friendly but less aggressive, as its most important aims are to reduce the delay and share the resources equally.

For evaluating the protocol, the only modification was embedding the weight adjustment part to the original Vegas. The simulation results in NS3 (Network Simulator 3) [90] showed that wVegas has more stable functionality than LIA and can converge to the equilibrium point faster. Moreover, when having more than two flows, intra-fairness could be enhanced for wVegas.

To sum up, wVegas tries to improve fairness and efficiency by equal distribution of traffic between various subflows with the help of a parameter called weight. However, its deficiency in competing with loss-based protocols leads to underutilization of resources in a network, especially a 5G mmWave one, so the attained throughput will become low. The reason is the Vegas-based strategy of the protocol, which cannot use highspeed networks to their full potential.

### 2) DELAY-EQUALIZED FAST

Another issue for Loss-based MPTCPs is that they suffer from slow responsiveness in 5G mmWave networks [87], in another word they cannot find the new equilibrium point when the network's state changes. On the other hand, wVegas cannot also react fast enough to changes because of its slow responsiveness. All the mentioned issues cause low performance in 5G mmWave networks [87].

Thus, DEFT (Delay-Equalized FAST) [87], a newly proposed MPTCP, tries to handle the mentioned problems that MPTCPs encounter in 5G mmWave networks. DEFT is a delay-based MPTCP that strives to estimate the lower bound of backlogged packets for subflows to ensure fast responsiveness, as a result, it can be more aggressive than wVegas.

The main reason for the DEFT emphasis on responsiveness is because this factor is critical in 5G mmWave networks as the state of a network changes frequently leading to new equilibrium points. However, if a protocol is stuck in the old points, it intensely affects the performance. If *wrold* is the old cwnd equilibrium point and *wrnew* is the new one, generally, equation (10) can be used to estimate the new point where $\beta$ is the responsiveness factor:

$$wrnew = wrold + \beta t \tag{10}$$

The two principal leverages that DEFT deploys to be more responsive and fairer are $\varepsilon r$, which indicates the number of backlogged packets for subflow $r$, and $\alpha r$, which denotes the targeted value for the backlogged packets on subflow $r$. If $wr$ is the cwnd value for subflow $r$, rttr is the RTT value for the subflow, and dr indicated minimum RTT, i.e., Round-trip propagation delay, DEFT employs equations (11) and (12) to adjust the sending rate.

If $\varepsilon r < \alpha r$ then:

$$wr = wr + min\{2wr, \gamma\alpha r\} \tag{11}$$

And if $\varepsilon r > \alpha r$ then:

$$wr = wr - \gamma(\varepsilon r - \alpha r) \tag{12}$$

where the first equation is the increment phase, and the second is for the descending phase. The scaling factor, i.e., $\gamma$, is between zero and one for specifying how sensitive the protocol should be. By looking at the equations, it can be inferred that the ultimate aim of DEFT is to adjust the cwnd in a way that the number of backlogged packets is approximately around the optimal value. By moving above the optimal value, i.e., $\alpha r$, DEFT reduces the sending rate. In contrast, if $\varepsilon r$ is less than $\alpha r$, it shows getting away from the targeted value, so cwnd will be increased. In addition to increasing the throughput and decreasing the RTT by accurately modifying the sending rate based on the mentioned equations, another essential goal can be achieved by the DEFT congestion control mechanism. As a result, when a route's condition changes, DEFT can detect it and immediately adjust the cwnd to the optimal value. This is more important when a degraded path is recovered and needs an instant increment in its sending rate.

For the evaluation, DEFT was deployed in two different scenarios with two and four obstacles to mime the blockage. Both 5G and LTE were deployed in the topology, and the carrier frequency for 5G was set to 28 GHz to satisfy the mmWave spectrum in order to mime mmWave channels.

The deployed simulation module was ns3-mmWave [86], like [85], and the exploited scheduler was LowestRTT [91], i.e., the default one in the Linux kernel.

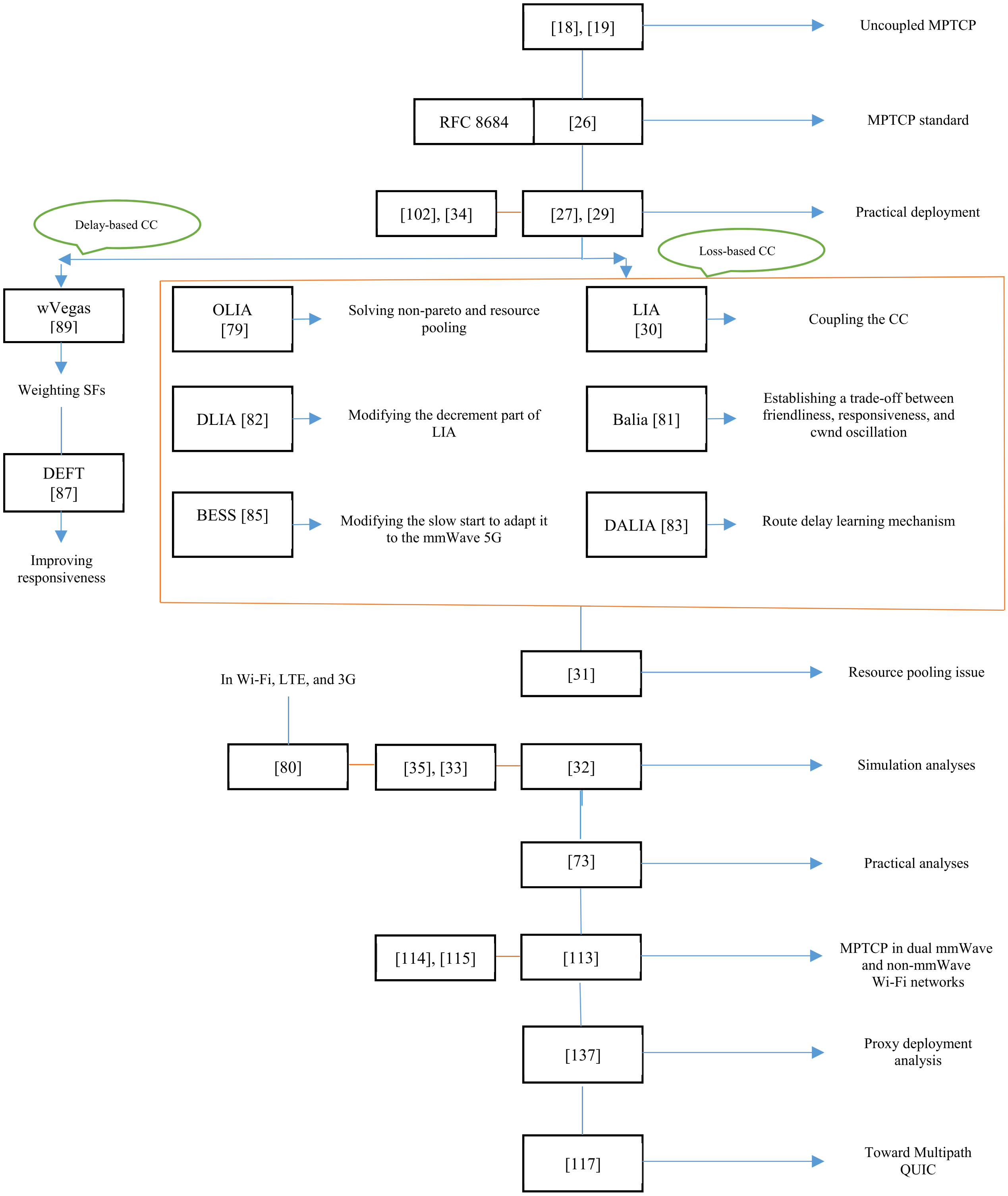

**FIGURE 2.** **MPTCP trend over 5G mmWave networks.**

The simulation results revealed that DEFT could attain higher goodput than Balia, DALIA, and wVegas through acceptable E2E (end-to-end) delays. E2E delays were equal to other protocols' or slightly higher, which is acceptable considering the high gained throughput.

Besides goodput and E2E delays, DEFT could achieve the most increased intra-protocol fairness compared to the other ones.

To sum up, DEFT, a delay-based MPTCP, was proposed to satisfy the uHSLLC (ultra-High Speed Low Latency

Communications) [92] feature of 5G networks along with 3GPP's newly presented functionality called ATSSS [47], which is for deploying MPTCP in 5GCN (5G Core Network) in order to steer the traffic over multi-RAT (Radio Access Technology). The simulation results indicated that the protocol could achieve its design goals and can be evaluated in practice to discover its potential capabilities when deployed over 5G mmWave networks in various scenarios.

Considering all characterized MPTCPs, the ones that are able to improve throughput, or decrease packet loss rate can be proper choices for the mmWave 5G. Moreover, the protocol should be aggressive enough that can steer traffic between the subflows and find the equilibrium point quickly, especially, in blockage situations. Among the investigated coupled protocols, DEFT can fulfill the demands by finding the fastest paths and delivering more packets.

At the first glance, it should be inferred that DEFT hones the latency, but finding the path with the lowest delivery time is one of the key criteria to improve the throughput as it can eliminate HoL blocking, buffer blocking, and out-of-order packet delivery issues. All three issues are crucial, especially the last one, which reduces the time that the receiver needs to assemble packets in order before the final delivery. Table 2 denotes a summary of different MPTCPs with their adaptability over 5G mmWave networks. Moreover, Figure 2 indicates the graphical evolution path of MPTCP toward mmWave 5G, also including various congestion control algorithms.

### C. IMPLEMENTATION CHARACTERISTICS FOR MPTCP

As designing a protocol is one way to go by researchers, implementing it is another side of the coin, which should be done by programmers. When implementing and inserting a protocol into the Linux kernel, some changes might be done and new parameters should be defined. Given existing MPTCPs, LIA, OLIA, Balia, and wVegas could pass the mentioned stage and be implemented in the Linux kernel in MPTCP released, i.e., v0.96 [93].

Given LIA, it does not try to push lots of parameters to the kernel on the programming side. The most important parameter in the implementation is alpha, reflecting the alpha in the corresponding RFC, which initially sets to one to have similar behavior to NewReno in the beginning. Then by having enough information, it can calculate alpha and adjust the cwnd based on that. As a result, the most important parameter for LIA is the initial value of alpha. The other parameters such as the slow start threshold and RTO has adapted from NewReno, which is using mostly the Linux kernel's default values. Finally, the implementation is using *alpha_scale_den* and *alpha_scale_num* for the scaling part of the nominator and denominator to shift the cwnd value whenever necessary.

As OLIA strives to have a category called M, which contains subflows with maximum sending rate, it defines two sets in the code including *M* and *B_not_M* dividing all subflows into two different categories described in section IV-A. Then calculates the maximum sending rate by comparing the current cwnd to the maximum one recorded till then. Moreover, based on the comparison of the current cwnd of a subflow and the maximum cwnd, decides how to set two epsilons for *M* or *B_not*_M, i.e., it sets the value to zero for routes in *M*, and one, i.e., positive, for routes in *B_not*_M. As a result, Two important inserted parameters into the Linux

**TABLE 2.** The compatibility of different MPTCPs over 5G mmWave.

| Congestion Control | Type | Congestion Indication Means | Compatibility with 5G mmWave | Summary |
|---|---|---|---|---|
| **TCP-NewReno [18]** | Uncoupled | Loss-based | Not compatible | Exploits a TCP on individual SFs independently- degrades fairness. |
| **LIA [30]** | Coupled | Loss-based | Not compatible | Coupling the SFs for better fairness- could achieve goals 1 and 2. |
| **OLIA [79]** | Coupled | Loss-based | Not compatible | Handles the non-Pareto feature of LIA- improves resource pooling |
| **Balia [81]** | Coupled | Loss-based | Not compatible | brings a trade-off between friendliness, responsiveness, and cwnd oscillation to generalize LIA and OLIA. |
| **D-LIA [82]** | Coupled | Loss-based | Not compatible | Modifies the decrement phase of the congestion avoidance of LIA. |
| **DALIA [83]** | Coupled | Loss-based | Not compatible | Adds a route delay learning mechanism to LIA and could balance delays in different subflows. |
| **BESS [85]** | Coupled | Loss-based | Compatible | Modifies the slow start phase of MPTCP by BDP estimation for each subflows and comparing it to the in-flight packets. |
| **wVegas [89]** | Coupled | Delay-based | Not compatible | Assigns a weight for each sobflow based on the congestion status. |
| **DEFT [87]** | Coupled | Delay-based | Compatible | Estimates the lower bound of backlogged packets to ensure fast responsiveness- finds new equilibrium points. |

kernel by OLIA are *epsilon_num* and *epsilon_den* with the initial values of zero and one, respectively, that can play major roles in determining the aggressiveness and friendliness of the protocol. Like LIA, the other parameters including slow start threshold and RTO are adapted from the default ones of the Linux Kernel.

The first step for Balia is comparing the current cwnd with the maximum one until then, to have the maximum sending rate anytime. Then it calculates a variable called *alpha* based on equation (13):

$$alpha = max\_rate/rate \tag{13}$$

Then calculates *ai*:

$$ai = \frac{\left(\sum_{pin\ R}^{1} rate\right) 2 * 10 * w_r}{(x_r + max\_rate) * (4x_r + max\_rate)} \tag{14}$$

where $x_r$ equals $w_r/rtt_r$.

Variable *ai* along with *md*, which is calculated by a particular formula using sending *cwnd*, *alpha*, and scaling are used to calculate the sending rate for each subflow. Other important parameters such as RTO are borrowed from the Linux kernel default ones.

wVegas implementation is similar to Vegas with some modifications along with adding a variables called *weight*. For calculating *wight*, the sending rate of a subflow is divided by the sum of the sending rates of all subflows. With the help of weight, the value for *alpha*, which is one of the important variables in Vegas, is calculated. Then using *diff*. i.e., a parameter from vegas, and calculating *alpha* by means of *weights*, the sending rates can be adjusted for each subflow.

## V. MPTCP SCHEDULING SCHEMES IN mmWave 5G

In the previous section, a thorough analysis of different MPTCPs' congestion control mechanisms and their compatibility with 5G mmWave networks were investigated. However, besides congestion control, packet scheduler and path manager are other paramount pillars of MPTCP. This section aims at analyzing various schedulers and the probability of their compatibility with 5G mmWave networks. In a broad view, the scheduler can be categorized into single-criterion and multicriteria. In the first one, only one parameter such as RTT, cwnd, packet loss ratio, inflight packets, or data segment is used to select the subflow for data transmission. On the second one, a combination of more than one feature can be used to have a more accurate decision [44].

### A. DEFAULT SCHEDULER

By default, Linux employs the LowestRTT scheduler in its kernel. It means that when a MPTCP-enabled source such as a server starts to send packets, first it selects the subflow with the lowest RTT and retransmits data until the cwnd of the subflow is full; then, it switches to the second path with the lowest RTT. This scheduling mechanism experiences some flaws when deployed in wireless networks as channels quality can differ from time to time, in contrast to wired networks, which have a low packet loss probability [94], wireless networks suffer from a higher rate [95]. Furthermore, the problem can be more severe by having frequent blockages in 5G mmWave networks and transitioning from LoS to NLoS states. When a UE is blocked by an obstacle, the RTT of that channel increases dramatically until it switches back to the LoS state, but LowestRTT cannot be aware of this transition which is one of its blatant downsides [35], [96]. In order to handle the issue and improve the scheduler's functionality, various schedulers have been proposed that we are going to inspect in this section.

### B. STREAM CONTROL TRANSPORT PROTOCOL

One of the first steps to exploit multiple paths was proposing SCTP (Stream Control Transport Protocol) [97]. In this case, one of the two subflows is used as the main path, and the other is for the first path's failure time or sending control packets. For extending SCTP, CMT-SCTP (Concurrent Multipath Transfer SCTP) [98] was recommended as an extension to SCTP. Despite its efforts for providing multipathing, the protocol, based on its round-robin scheduler, cannot perform very well in heterogeneous networks and suffer from out-of-order packets in the receiver.

### C. DELAY AWARE PACKET SCHEDULING

In order to solve the buffer blocking problem, a novel scheduler called DAPS (Delay Aware Packet Scheduling) [99] was suggested. The reason for comparing it with SCTP was that if the round-robin scheduler of the protocol can be enhanced, it would also be a suitable and generalized scheduler for MPTCP. The main power of DAPS is to estimate the delays on different subflows and use them as a key parameter for scheduling packets. The primary motivation behind DAPS was the need for a large buffer size as the network parameters such as RTT, loss probability, or bit rate are increased. When these parameters have large values, providing large buffers will be costly, and in the case of feeding, it will lead to the bufferbloating issue.

The first step for DAPS is calculating the number of ideal packets that should be transmitted into each subflow utilizing one-way delay and capacity for each subflow. Then the ideal number of all packets sent into the network can be calculated based on the numbers for individual subflows. Afterward, based on having all packets and each subflows packet numbers, DAPS creates a vector that contains the optimal order of the subflows for engaging the receivers' buffers as low as possible. DAPS proposes a suitable scheduler for heterogeneous networks by relying on these three steps.

The essential part of DAPS is estimating the one-way delay for each subflow, which is done by a timestamp procedure. As a result, DAPS keeps track of three parameters, the time for sending the packet, the delay for sending the ACK, and the time for sending the ACK. Then by subtracting the sum of the last two from the sending time, the one-way delay can be calculated.

For evaluating the scheduler, a network of two subflows with different bandwidths and delays was deployed. The simulation results showed that the scheduler could outperform round-robin and enhance throughput and delay. Furthermore, DAPS could keep its superiority in the round-robin mechanism and transmit more packets in terms of packet occupancy in the receiver. This superiority could be gained even when deliberate miscalculations were forced on DAPs in the range of 10% to 100%. Interestingly, in all conditions, DAPS could outperform round-robin.

To sum up, DAPS is a scheduler that, based on its delay-aware mechanism, could improve different KPIs, such as goodput, buffer occupancy, and the number of sent packets compared to round-robin. In order to find out the compatibility of the scheduler to 5G mmWave networks, we should have a look at the simulation scenario, where one of the subflows employs 1.6 Mbps and the other one has 400 Kbps bandwidth, which both are very low. Moreover, the delays are 20 ms and 200 ms, which are way far from desired latency for 5G mmWave networks. As a result, there are no evident signs that this scheduler can perform sufficiently in 5G mmWave networks, and more proof should be provided to see the superiority of DAPS in 5G mmWave networks.

### D. OUT-OF-ORDER TRANSMISSION FOR THE IN-ORDER ARRIVAL SCHEDULER

One of the problems that most schedulers suffer from is that they do not consider all subflows at the same time. The intense effect of this issue is when the RTTs of subflows differ. In this case, the number of out-of-order packets and the time interval between them will be increased. The adverse impact of the increased time gaps between packets' arrival is on real-time applications such as video gaming. As a result, to enhance the quality of experience in time-sensitive applications, OTIAS (Out-of-order Transmission for the In-order Arrival Scheduler) was proposed in [100].

When a MPTCP capable sender intends to transmit data, it should take three steps: 1) selection of the subflow based on cwnd availability, 2) if there are more than one subflow capable of sending data, choosing the appropriate one, and 3) specifying the amount of data that should be transmitted on each subflow. OTIAS has targeted the first two steps to improve the functionality of the MPTCP scheduler. Consequently, without considering the cwnd availability, all subflows can be candidates to be selected to send data through, considering the first step.

For improving the second step, OTIAS used a parameter called $T$. This parameter is the period of time for each subflow from scheduling a packet to delivering it to the receiver. When OTIAS wants to schedule a packet, it considers each subflows $T$, and then sends the packet to a subflow with the lowest $T$. The main goal of this mechanism is to deliver out-of-order packets in an in-order approach leading to decreased latency. The main tool for estimating $T$ for individual subflows is the number of RTTs that a packet should wait in the corresponding subflow to be sent. By adding 0.5 to this value and then multiplying the result by the subflow's RTT, $T$ can be calculated for the correlated path.

The simulation results indicated that the out-of-order packets for OTIAS always occupy less space than the LowestRTT. Moreover, the functionality of MPTCP enabled with OTIAS is enhanced, which led to stable, improved throughput.

To sum up, OTIAS could improve the functionality of the MPTCP scheduler by modifying the first two steps in scheduling packets. It tries to select the path with the lowest data delivery time even if it is unavailable based on the current cwnd size. Selecting the path with the lowest delivery time is one of the critical touchstones in honing the throughput, as it helps to prevent HoL blocking, buffer blocking, and out-of-order packet delivery issues. The reason is that the receiver can deliver final in-order packets quickly without the need for the pre-delivery procedure. However, it seems that this scheduler cannot adapt to 5G mmWave networks. The reason is that when a blockage occurs in the network, the delay for the mmWave channel is increased; as a result, OTIAS will choose the other subflows, for example, LTE. Consequently, the large available bandwidth in the mmWave channel will be wasted entirely. Moreover, random packet drops will frequently happen in wireless channels, which OTIAS should embed a proper reaction.

### E. SHORTEST TRANSMISSION TIME FIRST AND BLOCK ESTIMATION

As the scheduler is responsible for choosing the subflow that data should be transmitted, it has a critical role in lowering the latency in multipath communication, especially when different technologies such as 5G, LTE, or Wi-Fi are used. As a result, deploying asymmetric networks is one of the major hurdles in the way of implementing MPTCP, as they intensify the out-of-order problem. Moreover, when latency-sensitive applications are used, the MPTCP's scheduler also plays a fundamental role in fulfilling the aim. Based on the mentioned reasons, deploying reactive schedulers such as the default one in the Linux kernel is not a good choice when the latency is a critical KPI, thus, proactive mechanisms should be deployed [101]. If no appropriate scheduler is deployed, packets through faster links will be delivered first, and then they will wait in the buffer for the slower ones leading to HoL blocking problem. This can even be more intensified as the cwnd is a restricting factor. However, if this was not true, lower latencies could be achieved [80], [102].

Motivated by the mentioned flaws, authors in [101], and [103] have proposed schedulers called STTF (Shortest Transmission Time First) and BLEST (BLock ESTimation) on Linux's default MPTCP. The latter scheduler was for handling the HoL issue on the receivers' buffers, and the former one strived to choose the fastest subflow first. BLEST tries to estimate the blocking in the buffer considering the selected subflow for the current transmission. It means that if the scheduler selects subflow $r$, what could be the blocking amount for the receiver's buffer? This is a proactive mechanism as selecting the path is done before scheduling a packet. In this case, in the first step, BLEST sometimes opts for the slower path, i.e., the one with the higher RTT, because sending all packets from a faster subflow will cause HoL, then the remaining data are transmitted through the faster subflow. Selecting the lower path first, then the faster one alleviates the buffer-blocking problem. Sometimes, BLEST also skips the available path as it can detect that sending data could cause the HoL issue and wait for favorable routes.

On the other hand, STTF tries to obtain characteristics of different subflows and estimates the transfer time for individual segments. This mechanism aims to find the faster path, then schedule data to be transmitted through it even if it is full, as the name indicates. The employed parameters to calculate the transfer time are RTT for each subflow, congestion state for individual subflows, and already queued segments in each subflow.

The simulation results for an asymmetric network including WLAN (Wireless Local Area Network) and 3G indicated that both new schedulers are capable of choosing the best path compared to LowestRTT. More interestingly, as the burst size and RTT increase, the BLEST and STTF maintain their superiority compared to TCP and LowestRTT. Moreover, comparing the page load time also indicated that the proposed schedulers could outperform Linux's default scheduler in most cases.

To sum up, as MPTCP cannot fulfill the need for latency-sensitive applications in asymmetric networks [104], also proved by practical experiments [105], new schedulers should be proposed. As a result, BLEST and STTF were presented in order to choose the best subflow for transmitting the data. As these schedulers are trying to select the best path based on some factors proactively, their application in 5G mmWave networks can be beneficial as they can avoid sending more data to blocked channels.

### F. EARLIEST COMPLETION FIRST

The investigation of the downside of the Linux default scheduler in heterogeneous networks continued in [106], and the testbed and practical analysis results conformed to the ones in previous studies such as [80], [99], [103], and [107] by indicating the deficiency of the LowestRTT in asymmetric networks. Moreover, this flaw is more intense in the video streaming connections; as the heterogeneity of the network increases, the actual bit rates for clients get further than the ideal one. The reason is that the RTT-based schedulers fail to select the fastest paths. As a result, more parameters should be employed to have a clear manifest from the subflows' conditions, leading to the proposal of the ECF (Earliest Completion First) scheduler [106]. In addition to RTT, ECF utilizes cwnd size and connection's sending buffer's size to sufficiently assign packets. In the first step, ECF selects the best subflow with the lowest RTT based on the available RTTs. Then, when it is going to schedule a packet, calculates the sum of the waiting time for sending the packet from the best path and the transmission delay. If we imagine that cwnd is not available for this path, then the waiting time would equal the subflow's RTT, so after passing the RTT time, the cwnd would be open, and the transmission time could be calculated by multiplication of the subflow's RTT in the division of packet size in cwnd of the subflow r. This can be seen in equation (15):

$$transmissionTime = RTT_f + (\frac{k}{cwnd_r}) \times RTT_f \qquad (15)$$

Based on the notation in [106], $RTT_f$ is the $RTT$ for the fastest path, $k$ is the number of packets to be scheduled, and $cwnd_r$ is the cwnd size for the subflow. If this value is smaller than the RTT for the other subflows, then the fastest subflow will be selected to transmit the data even if its cwnd is full. Otherwise, the other path will be the candidate for sending the data. ECF can also function when subflows are more than two as the principle keeps its validation.

For evaluation, ECF was implemented in the Linux kernel and compared to three other schedulers, including the default, i.e., LowestRTT, DAPS [99], and BLEST [101] in video streaming, downloading a file, and web browsing scenarios.

The results for the video streaming when the bandwidth for subflows was stable showed that ECF could function close to the ideal bit rate and has enhanced performance compared to the other three schedulers. After ECF, the order was BLEST, default, and DAPS. The worst performance for DAPS was the interesting output of the scenario, as it could not even function as well as the default scheduler. The reason for the improved functionality of ECF is in its mechanism of dividing the traffic more sufficiently and increasing the portion of the fastest path. ECF aggressively deploys the quickest path by increasing its cwnd sharply.

In contrast, this aggressiveness is higher for the slower path in the LowestRTT scheduler, which leads to the worst cwnd adjustment for the default scheduler after ECF, BLEST, and DAPS. As a result, cwnd initialization and entering the slow start phase decreases in ECF drastically compared to when the others were exploited. In terms of out-of-order packets, ECF also could show the best performance by reducing the out-of-order delay for almost all of the packets to 0.8 seconds or less. This superiority could not be the same in symmetric networks as all schedulers except DAPS could reduce the out-of-order delay.

When the bandwidth is constantly changing for the WiFi and LTE network for a dynamic configuration, ECF could continue to have the highest average throughput compared to others. Interestingly, the worst performance was for DAPS, not the LowestRTT scheduler, and BLEST could be as good as the default one or slightly better than that.

In terms of downloading time, for small and large objects, ECF and default experienced intimate performances by completing the download at approximately the same time. However, for medium-sized objects, ECF could slightly reduce the downloading time. This superiority could be preserved in page loading as ECF could perform better than the others, especially when the heterogeneity was higher.

To sum up, ECF is a scheduling algorithm that tries to send packets from the fastest subflow as much as possible. The algorithm relies on the connection level's RTT, cwnd, and buffer size. Based on the ECF strategy in choosing the best paths, it seems that this scheduler has the capability of adapting to 5G mmWave networks. When the network is in a LoS state, the algorithm can choose the mmWave channel as the best path to transmit the data. In contrast, when blockage forces the network into a NLoS state, ECF can switch the priorities between the mmWave and LTE channels. Moreover, the advantage of ECF in reducing cwnd resetting is beneficial for the 5G mmWave network because this network supports vast data rates, and frequent cwnd initializations could damage the network's performance.

### G. OFFLOADING BY RESTRICTION

To handle the scheduling algorithm, especially LowestRTT in loss-based MPTCPs, a scheduler called OBR (Offloading By Restriction) was proposed in [96]. The foremost aim of this scheduler is to handle the packet transmission in heterogeneous networks when *NR* and LTE co-exist because if there are no suitable strategies, packets are sent to the *NR* channels without caring that the RTT is fluctuating until the channel is filled. The reason is that $RTT_{NR}$ usually is smaller than $RTT_{LTE}$, except when NLoS is the ongoing state. In order to control the sent packets to *NR*, OBR exploits a parameter called Cutoff, denoted by $\alpha$ in the protocol. The Cutoff factor is used to reduce the number of transmitted packets to NR in order to offload a portion of data to LTE; as a result, the cwnd for NR is adjusted by means of equation (16) when the subflow is fully deployed:

$$wNR = \alpha wNR \tag{16}$$

The outcome of the equation is the cwnd in *NR*, so the offloaded data to LTE will equal equation (17):

$$WLTE = (1 - \alpha)wNR \tag{17}$$

The offloading mechanism is canceled when the value of $\alpha$ equals one, which is a one-way approach from *NR* to LTE.

The simulation result performed in MATLAB indicated that as the PLR (Packet Loss Ratio) of the *NR* path increases, the need for more offloading from *NR* to LTE is required. More obviously, when PLR in LTE is low and the channel has a more stable condition, the protocol can function better, as the LTE channel is a substitute for the offloaded data from *NR*.

Finally, results showed that OBR could achieve higher throughput compared to LIA and MOA(Multipath Offloading Algorithm) [108], an energy-efficient offloading mechanism. However, when the PLR for *NR* decreases, MOA offers better performance than OBR.

To sum up, OBR is a strategy to offload the traffic from *NR* to LTE when the *NR* path is full. In order to accomplish its goal, the protocol offloads a fraction of the *NR* to LTE determined by the Cutoff factor. This approach can compensate for the throughput degradation by the blockage issue in 5G mmWave networks and alleviate it. However, to find its path to practice, more simulation and practical analysis should be done.

### H. ADAPTIVE OFFLOADING FRAMEWORK

The adaptive offloading framework proposed another scheduling method for handling the random loss issue in 5G mmWave [109]. The motivation behind the scheduler is to select paths with lower delay and packet loss rates to transmit the data. The authors claim that by modifying the MPTCP offloading control module, it is possible to make a balance in a way that the throughput of the network can be enhanced in the case of video streaming. They claimed that if the scheduler is embedded in LIA and chooses the LTE or WiFi as the preferred over the mmWave for transmitting the video segment, the throughput will be improved because of the channel's low packet loss rate. However, this conclusion must be supported by simulation or practical tests.

To sum up, the adaptive offloading framework declares that transmitting the video segments from the low delay and packet loss rate channels can improve the throughput. This claim may be valid in some situations, however, not employing the high available bandwidth in the mmWave can waste the provided high data rate.

### I. MULTIPATH TCP SCHEDULER

Moving forward, besides 5G mmWave networks, mmWave is also deployed in other wireless environments such as IEEE 802.11ad, which utilizes the 2 GHz wireless channel in the mmWave spectrum. The 802.11ad frequency in the 60 GHz spectrum can be used with 802.11ac, which employs lower channel frequencies simultaneously. This co-existence can benefit from switching the transmission channel to 802.11ac when 802.11ad is blocked. As a result, providing more resiliency and reliability among enhancing the throughput [110]. Deploying these kinds of networks is going to be common as APs and user devices can support different frequencies such as 2.4, 5, 6, and 60 GHz, providing up to 7.2 Gbps [111], [112].

Several types of research have indicated that exploiting conventional MPTCP in the aforementioned networks can lead to performance degradation [113], [114], [115], leading proposing MuSher (Multipath TCP Scheduler) [110] scheduler on top of LIA, which could solve the issue. This scheduler aims the main problem of the dual-band 802.11ad/ac, which is the existence of dynamic situations, however, it could also enhance other aspects.

The first leverage of designing MuSher is that its designers believe the chief cause of throughput degradations is that a scheduler sends data to different subflows without considering the ratio between their throughputs, which leads to out-of-order packets in the receiver. As a result, the first design step for MuSher is to find the unique optimal throughput ratio between different paths by relying on its novel mechanism. For estimating this ratio, the number of the sent packets in subflows is observed in the runtime, and the ratio is calculated every 200 ms to skip large intervals or pre-mature estimations.

In addition to calculating the throughput ratio, MuSher monitors the network to find changes. For this, MuSher makes special attention to the total throughput and sending queues of either 802.11ad or 802.11ac when they are reduced.

Besides handling the network dynamics to solve the blockage problem, MuSher strives to return the cwnd value to its size just before the last loss event to make the recovery time as small as possible. Moreover, when a blockage occurs, the client sends a signal to the server to notify it about the ongoing condition, which is the initialization of the cwnd recovery process. All these modifications assist the scheduler in honing MPTCP's performance in dual-band networks.

The simulations indicated that MuSher could achieve its goals and enhanced the network's performance in different scenarios compared to LowestRTT. When the channels have varying characteristics, MuSher could attain higher throughput compared to the Linux default scheduler, i.e., Lowest RTT. It is intriguing that as the variations increase, the difference between the two schedulers gets higher, and MuSher reaches a much better performance.

When the user is moving, MuSher also could achieve better performance compared to LowestRTT. However, this enhancement is negligible when the user directly goes toward or away from the AP. On the other hand, the improvement is noticeable when the movement is lateral.

The above-obtained results were also confirmed for the scenario with blockage, and based on the fast recovery enhancement of MuSher, the user experience could improve compared to the time when LowestRTT was deployed. The improvements are because of MuSher's well-performed scanning mechanism that can accurately analyze the network's conditions.

To sum up, MuSher is a scheduler designed for heterogeneous networks when both 802.11ad and 802.11ac are deployed simultaneously. This scheduler functions based on scanning the network and estimating the throughput ratio between two existing subflows. The simulation results showed the proficiency of the scheduler compared to LowestRTT. As the exploited spectrum in 802.11ad could be mmWave, MuSher's well-behaving can be extended to the new cellular network as it can be concluded that this scheduler, based on its improved functionality, especially in dealing with blockage events, can be a suitable choice for 5G mmWave networks.

### J. DEEP REINFORCEMENT SCHEDULER

One of the main tools for extending new schedulers is machine learning techniques, selected in [116]. The deployed technique in the scheduler is DQN (Deep Q Network), and RL (Reinforcement Learning) algorithm trying to converge to the optimal point based on the trial-and-error mechanism in order to steer traffic appropriately into different paths scheduling purpose. The model was trained based on the provided values in a file-downloading process in a network.

In order to evaluate the new scheduler, a scenario consisting of multiple subflows having 5 Mbps bandwidth was used. In order to make the diversity, the latency for one of the subflows was constantly changing. The results indicated only a slight difference between the scheduler and LowestRTT, and the enhancement range was between 4.45% and 7.58%.

To sum up, the new scheduler indeed aimed at improving the performance of MPTCP in 5G networks by including an RL technique in the scheduling module of MPQUIC. However, the slightly improved performance cannot guarantee an enhancement in 5G networks. Moreover, fluctuations in the training process can be a hurdle in implementing the scheduler in 5G networks, as the complexity increases drastically.

The positive point of the approach was MPQUIC (Multipath QUIC) [117] implementation to prevent the necessity of implementing the scheduler in the Linux kernel's transport layer. In contrast to conventional transport layers, QUIC (Quick UDP Internet Connections) [118], a protocol based on UDP (User Datagram Protocol), is not applied in the Linux kernel so that it can be easily implemented and updated.

To conclude, scheduling is one of the most important components of MPTCP, which can impact the functionality of the protocol dramatically. Scheduling along with path manager and congestion control, form the basis of the multipathing and determine how the protocol functions. The scheduler can operate based on only one feature such as RTT, i.e., single-criterion, or more than one parameter such as a combination of RTT, cwnd, and loss rate, i.e., multi-criteria. Moreover, in designing a novel scheduler various aspects such as the lifetime of the connection, the level of redundancy to prioritize some packets including latency-sensitive ones, and the characteristic of the transmission environment should be considered.

More importantly, a scheduler should be an application-aware component, so it can function based on the needs and requirements of the application and can be tested in a runtime environment [119] implemented in the Linux Kernel [120].

Table 3 indicates the compatibility of the various schedulers with 5G mmWave. Furthermore, Figure 3 shows the position of the congestion control and scheduler modules in the transport layer before the network layer. In the first step, the scheduler selects the corresponding path for data transmission, then the congestion control mechanism adjusts the sending rate.

### K. SCHEDULERS FROM THE OPERATIONAL POINT OF VIEW

By thoroughly looking at the schedulers and their mechanisms, an approach can be dividing them into five categories, including elemental, negotiation-based, policy-based, machine learning-based, and cross-layer-based schedulers [121].

In elemental schedulers, the decision is made regarding some network parameters such as RTT, cwnd, or packet losses. These schedulers are very common and easy to design, test, and be compared. The most common elemental scheduler is the LowestRTT, which is the default scheduler in the Linux kernel. Another important scheduler introduced in this survey is BLEST [103], which estimates the blocking time and decides whether to utilize the subfow or not to transmit the data in order to prevent HoL blocking. ECF [106], which considers RTT, cwnd, and sending buffer size in its scheduling algorithm also falls into this category. STTF [101], which strives to find the fastest path to transmit data by employing RTT, congestion state, and queued segments is also a part of this category.

Moreover, they were some other efforts in designing elemental schedulers such as LWS (Large Window Space) [122], which considers the most recent size for the congestion window in scheduling packets and selects the subflow with the largest available congestion window to enhance the throughput.

Negotiation-based schedulers consider other parameters than network ones like QoS or data type, such as [123], which aims at enhancing QoE for Web services. The scheduler tries to opine the RTT, then based on that establishes a trade-off between the throughput and QoE. The goal is to prevent the QoE variation by attaining the lowest needed throughput.

In policy-based schedulers, as the name indicates, the aim is to establish a policy, then adhere to that policy in scheduling packets. These policies can consider cost, connection lifetime, or some other factors. There are not many schedulers of this category for wireless networks, however, an example of this category can be RED (REdundant Diversity scheduling) [124]. It is true that redundancy scheduling policies can be useful for time-sensitive applications in heterogeneous networks as they send replicated packest to different subflows, however, shared bottlenecks can be a hurdle in the way of these schedules to perform well. As a result, RED was proposed to handle this problem by prioritizing packet replication by uncorrelated paths. RED puts a step higher than MPTCP redundant scheduler and makes some guesses to find the same path subflows in order to determine subflows not sharing the same bottleneck. The separation of subflows that don't belong to the same bottleneck can relieve the effect of shared bottlenecks and provide some avails for time-sensitive applications.

Machine learning-based schedulers deploy various algorithms including supervised, unsupervised, and reinforcement learning. As an example of reinforcement learning, [116], [125] tried to deploy DQN based on a trial-and-error approach to find an optimal point for traffic steering. However, there are serious doubts about machine learning schemes, especially, the reinforcement ones in having the capability of converging to the optimal point for a scheduler in wireless networks due to the intermittent characteristics of these networks.

The last category including cross-layer-based schedulers intends to use information from the higher layer, i.e., the application layer, or lower ones. Due to the lack of a standard framework for cross-layer approaches, lack of proper architecture, and difficulties in designing such approaches [126], employing this strategy can be strenuous. However, some researchers have tried to use this mechanism, such as QAware [127]. QAware claims that having local information of queues can help the scheduler make more suitable decisions, thus, they combine end-to-end connection delay with the local information on queues, i.e. each subflow's queue, in order to make a novel scheduling mechanism for maximizing the throughput, which could outperform LowestRTT and ECF in a simple topology.

Among the categories, elemental, Policy-based, and machine-learning schedulers are gaining popularity and there are many ongoing efforts to utilize these approaches in designing new schedulers. The reason for the elemental scheme deployment can be found in its easy-to-use mechanism. In this case, a scheduler can be redesigned based on previous ones just by modifying and adding some new parameters or it can be redesigned from scratch by the use of innovative parameters. In the policy-based controller regard, setting up a policy and then following it for different scenarios and enhancing it can be convenient in order to increase the functionality of a specific scheduler in that particular circumstance. About machine learning schedulers, the reason for the prevalence can be found in, first, the popularity that machine learning approaches are gaining in the networking community, second, enlargening the data sets that are available to the public, and third, the emergence of new robust algorithm that can assist researchers in creating new schedulers.

By having an overall look on the section we can get to the following conclusion: a scheduler has the responsibility of selecting a path for data to be transmitted through. This component functions before the congestion control one, so its proper functionality can even assist the congestion control to operate more sufficiently. As a consequence, it should select the path in a way that enhances the resource utilization in the network, which can be attained by steering packets in robust mechanisms to the existing subflows. It should be done in a way that prevents HoL blocking, buffer blocking, buffer bloating, and out-of-order packet delivery issues. For handling these issues a scheduler should be more responsive and can have a clearer insight into the paths, as a result, can utilize them to their potential.

Figure 4 shows the graphical evolution path of the MPTCP scheduler toward mmWave 5G.

## VI. MPTCP STEPS TOWARD mmWave 5G

This section tries to track the MPTCP path toward mmWave 5G. Moreover, it is going to analyze technologies that can ease the way of MPTCP toward mmWave 5G such as proxy. Finally, real-world experiments will be investigated to find out the functionality of the protocol in practice.

**TABLE 3.** Compatibility of the various schedulers to 5G mmWave.

| Scheduler | Compatibility with 5G mmWave | Summary |
|---|---|---|
| **DAPS [99]** | Not Compatible | estimate the delays on different subflows and use them as a key parameter for scheduling the packets to handle the buffer blocking issuer |
| **OTIAS [100]** | Not Compatible | Considering the period of time for each subflow from scheduling a packet to delivering it to the receiver in order to improve QoS for time-sensitive applications |
| **BLEST [103]** | compatible | Tries to estimate the blocking probability in the receiver's buffer considering the selected subflow for the current transmission to prevent buffer blocking and HoL |
| **STTF [101]** | compatible | Tries to obtain characteristics of different subflows and estimates the transfer time for individual segments to find the fastest path |
| **ECF [106]** | Compatible | 1) Selects the best subflow with the lowest RTT<br>2) Calculates the sum of the waiting time for sending the packet from the best path and the transmission delay<br>3) If the calculated value is smaller than the RTT for the other subflows, then the fastest subflow will be selected even if cwnd is not available. |
| **OBR [96]** | compatible | Deploying the Cutoff factor to reduce the number of transmitted packets to NR in order to offload data to LTE in order to alleviate the heterogeneity of the NR and LTE deployment simultaneously. |
| **Adaptive Offloading Framework [109]** | Not Compatible | Selects paths with lower delay and packet loss rates to transmit the data by modifying the MPTCP offloading control module to reduce the random loss issue in 5G networks. |
| **MuSher [110]** | compatible | finds the unique optimal throughput ratio between different paths by observing the sent packets in subflows in the runtime, and calculating the ratio every 200 ms. |
| **Deep Reinforcement Scheduler [116]** | Not Compatible | tries to converge to the optimal point based on the trial-and-error mechanism by using DQN. |

### A. MPTCP VERSUS TCP

As mentioned earlier, 5G mmWave can provide high data rates through its wide spectrum. However, some flaws such as blockage and random packet drops are hurdles in the way of implementing TCP and MPTCP over the network. One of the strategies that can assist TCP in 5G mmWave networks can be exploiting techniques such as HARQ and RLC-AM mode, which is called link-layer retransmission [33]. HARQ is a method that is used in the MAC layer, and its responsibility is to ask for some additional information from the sender when it detects some errors in a packet. As a result, the receiver can recover the main packet based on the acquired redundancies transmitted by the sender.

Moreover, there is a layer called RLC on top of MAC, which is one of the 5G protocol stack layers [128]. When the RCL-AM is enabled, additional retransmissions can be done to recover some of the dropped packets. The primary purpose of the link-layer retransmission, i.e., the mentioned methods, is to reduce the burden of loss and congestion detection from the transport layer and help to keep cwnd as high as possible. However, the number of retransmission efforts that HARQ and RLC-AM can support is finite, and if they cannot mask the losses from TCP, the protocol will do it by itself [33].

The simulation results in [33] proved that deploying link-layer retransmission can enhance the throughput of TCP (CUBIC in the simulations) over wireless networks when the distance between the UE and gNB is not high. However, the methods lose their superiorities and fail to function properly with increased distance, so the best case is to deploy both HARQ and RLC-AM at shorter distances. Moreover, the results proved that, when short-lived flows exist in the network, if the retransmission could be done by lower layers instead of the transport layer, it can improve downloading by reducing the time.

In the second phase of [33], the deployment of MPTCP in mmWave and LTE was analyzed. They analyzed the protocol's behavior in a network consisting of mmWave and LTE channels at the same time. The first output of the results was about the complementary role of the LTE link in covering mmWave flaws and enhancing the overall throughput. The second conclusion was about one of the MPTCP's flavor's performances, Balia. The results revealed that this protocol misunderstood the fluctuated nature of mmWave as congestion, as a result, to accomplish the third goal of TCP, it shifted the traffic to the LTE channel. However, this mechanism damages the functionality of Balia in a way that its throughput goes under the TCP's on the best path. Yet, based on the observations, the uncoupled CUBIC i.e, functioning in each subflow independently, is able to tolerate this issue as it sees subflows as individual paths by damaging fairness, which sacrifices the second goal of MPTCP design. The behavior of Balia was repeated in OLIA, and this MPTCP could not also appropriately utilize mmWave.

The third analysis was about short and long-lived flows' existence. When there are short-lived flows such as downloading small-size files, Balia can function a bit better than CUBIC. However, for long-lived flows, CUBIC can perform way better. Looking at all the results infers that for long-lived sessions and long distances, employing coupled MPTCP cannot satisfy all the design goals for the protocol. The authors' suggestion is to expand CUBIC to a coupled protocol so that based on its mechanism and added coupled nature, it may fulfill all three design goals.

Finally, deploying the LTE as the way of sending the ACKs, i.e., uplink, was considered, however, it did not show any improvements and can be omitted from the possible proposals.

It is essential to see how MPTCP reacts to different wireless channels because deploying more than one in a connection can cause issues such as out-of-order packets in the receiver [80], [129]. In contrast to TCP, MPTCP keeps

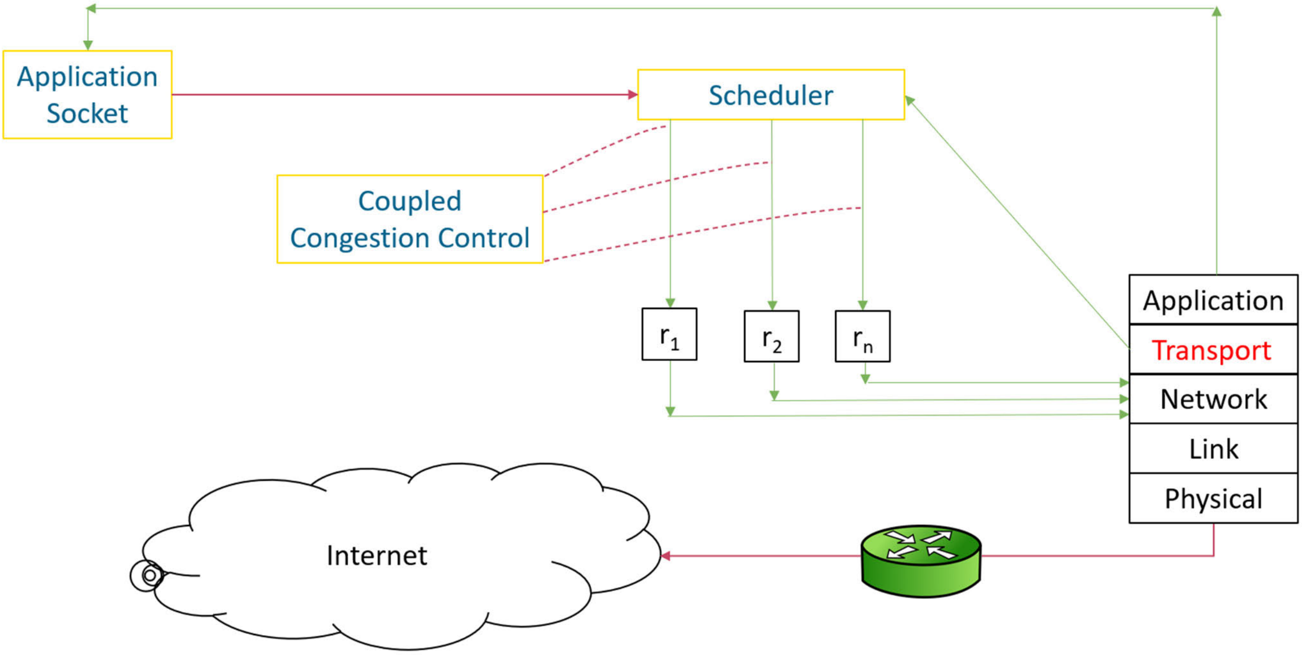

**FIGURE 3.** Congestion control and scheduler implementation positioning before the network layer.

different cwnd for each subflow and, based on their values, sends the packets over them. As paths can be heterogenous, packets will arrive in different orders than those sent. MPTCP can detect out-of-order packets by the comparison of the connection-level sequence number and SF-level sequence number. The former is for the whole flow kept by the TCP socket, and the latter is for subflows. When these two values are not equal in the receiver, it indicates the out-of-order entrance to the receiver. Analyzing the out-of-order problem in [130] by using four TCPs designed for reordering problems, which are D-SACK [131], Eifel [132], TCP-DOOR [133], and F-RTO [134] was conducted to give an insight into the problem. To have a thorough analysis, four different scenarios were chosen, and in all scenarios, there are two wireless channels with different data rates. Scenario one includes two highspeed channels, scenario two includes two low-speed channels, scenario three one highspeed and one low-speed channel, and the last scenario one medium-speed and one low-speed channel. The deployed technology in low-speed wireless channels was IEEE 802.11b, and in the others, IEEE 802.11g.

The simulation results showed that TCP-DOOR could outperform other TCPs in the first scenario in terms of throughput by an increased out-of-order ratio compared to the others. It is interesting to mention that F-RTO had a performance lower than MPTCP without any particular reordering.

In the second scenario, except for Eifel which could not show any enhancement, all other three protocols could gain some improvements compared to MPTCP.

In the third scenario, D-SACK could improve the throughput more than the others and achieve the highest. Like the previous scenario, Eifel could not achieve much enhancement compared to the default MPTCP.

In the last scenario, D-SACK, F-RTO, Eifel, and TCP-DOOR could improve the throughput, and the improvement for the first two was greater than for the last ones. However, MPTCP, without any reordering, relied on the slow channel and could not gain a throughput of more than 2 Mbps. Considering the outputs for all scenarios, having a proper packet scheduling strategy for preventing out-of-order packets issue, is necessary to prevent MPTCP throughput degradation, especially in heterogeneous networks. A well-designed scheduler should be able to handle packet transmission to relieve the out-of-order packet delivery problem.

### 1) MPTCP WITHIN mmWave

For putting the step to a higher stage, analyzing a network having two wireless channels, one of them employing mmWave, is necessary to see how MPTCP reacts in this situation. As a result, a comprehensive investigation of MPTCP over 802.11ad and 802.11ac was done in [115]. As IEEE802.11ad can support the mmWave spectrum, this analysis would be a guideline in deploying MPTCP over 5G mmWave networks because this wireless technology is also sensitive to blockage and mobility. Furthermore, having FST (Fast Session Transfer) option in 802.11ad makes it possible to shift the traffic between the technologies. By analyzing coupled and uncoupled MPTCPs over dual-band 802.11ad/802.ac, a clear image of the protocol's performance would be in hand. The coupled MPTCPs included LIA, OLIA, and Balia, on the other side, CUBIC was exploited as the uncoupled one.

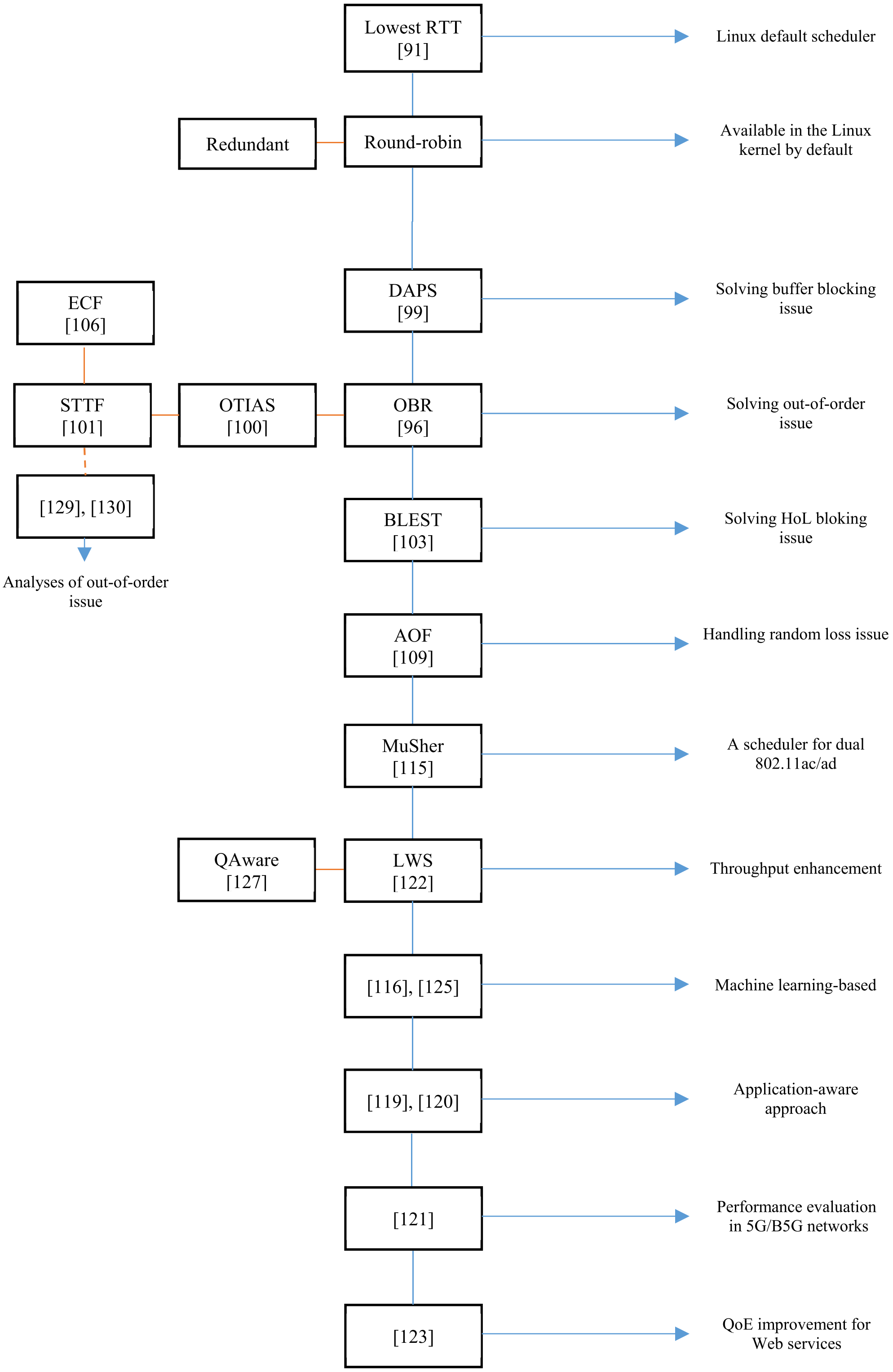

**FIGURE 4.** MPTCP scheduler trend over 5G mmWave networks.

The experimental results revealed that all four MPTCPs could improve the performance over the dual-band network compared to TCP in the order of CUBIC, LIA, OLIA, and Balia with the percentage of 29%, 9%, 9%, and 2%, respectively, when the UE and AP are close; and 32%, 27%, 5%, and 14%, for long distances. Moreover, the results showed that uncoupled CUBIC could gain almost the throughput equal to the sum of the channels, which can be attained by sacrificing fairness. The results also showed that coupled MPTCPs need to have some modifications to deploy different paths simultaneously and benefit from having more than one interface on the devices, which can assist them in fulfilling design goal one.

The deployed spectrum for 802.11ad was 60 GHz throughout the test to satisfy the mmWave features. As a result, it can be concluded that coupled MPTCPs are not suitable to be employed in 5G mmWave networks while keeping fairness. On the other hand, uncoupled ones can achieve higher performances by giving up the second design goal.

### *B. MPTCP DEPLOYMENT IN 5G AND 4G NETWORKS, ANALYSES, AND FEASIBILITIES*

Through the use of TCP in cellular communication, the chance of connection interruptions because of various issues such as blockage in 5G networks can be increased and MPTCP has been mentioned as one of the candidates that can relieve this problem. Unfortunately, there are not many investigations on MPTCP through 5G due to some difficulties such as device shortage, lack of proper testbeds, and complexity of the implementation [35]. Nevertheless, there have been some simulations, especially in NS-3 to analyze the functionality of MPTCP when one of the networks is 5G. The results in [35] showed that the LoS-NLoS transition problem is intense in 5G mmWave networks and can be a misleading element for MPTCP. However, they claim that 5G devices can connect to various RATs (Radio Access Technologies) and that the deployment of MPTCP can bring some benefits, then they supported their claims by providing simulation results. The results showed that having LTE and 5GmmWave at the same time utilizing an uncoupled MPTCP can bring 30-40 percent throughput enhancement. However, this outcome was not valid for the Balia (a coupled MPTCP) and it could achieve low throughput compared to TCP on the mmWave channel. The reason is that Balia sees the mmWave as a lossy medium and steers the traffic to the LTE, which has a much lower bandwidth compared to mmWave. As a result, the definition of a congestion control mechanism for MPTCP that can meet three design goals when employed in mmWave is a challenging and open issue. Some congestion controls waste the available bandwidth, however, achieving fairness, and some others act vice versa. The main reason is that having an optimal balance between the traffic through all subflows is difficult.

Furthermore, the dependency of the mmWave on the distances was emphasized throughout the analyses, and as the distance increased the performance of MPTCP was getting better compared to TCP deployed on mmWave. This effect was negatively severe at 73 GHz as its lossy characteristic is more extreme than at 28 GHz.

As the next cellular generation may connect to more antennas such as LTE and mmWave simultaneously, MPTCP deployment can ease the use of this concept without the necessity of the lower layer interactions [33]. However, the question is whether MPTCP is ready to be exploited in such networks; The question that authors in [33] strived to answer by having simulations tests to investigate the functionality of various MPTCPs such as LIA, OLIA, Balia, and uncoupled CUBIC by co-deploying LTE and mmWave networks. The results revealed that the viability of MPTCP relies on some factors such as the distance between the UE and the base stations and being in LoS or NloS states. When the UE is in the LoS state and the distance is low, MPTCP can outperform TCP in 28 GHz and 73 GHz mmWave deployment. However, when LTE replaces 73 GHz, the performance tends to decline close to TCP, when uncoupled cubic is used, however, coupled versions are not capable of competing with TCP as they try to be fairer rather being more aggressive. Furthermore, the deployment of 73 GHz in long distances penalized the network performance because of the lossy channel characteristics of the spectrum. When the distance was getting closer to 150 meters, the best performance was for the uncoupled CUBIC over 28 GHz and LTE, which indicates that having a secondary reliable network such as LTE can assist mmWave unstable channel in reaching higher throughput. As a consequence, the distance from the base station and the reliability of the second network have critical roles in the functionality of MPTCP when deployed in mmWave channels.

Another important outcome is that coupled congestion control mechanisms such as Balia fail to achieve high throughputs. This can lay in the design scheme of these protocols as they try to stick to goals two and three of MPTCP to be fairer and perform congestion balancing between flows. In this sense, as the mmWave can be seen as a lossy flow, to adhere to goal three, the traffic from high potential mmWave is steered to the LTE one, which impairs the accomplished throughput.

There is a comprehensive simulation study in [32] in order to find the best combination of congestion control and scheduler in 4G and 5G. As had been shown previously in [73], the deployment of MPTCP in 4G and 5G can not bring benefits by itself and a proper combination of the congestion control and scheduler should be selected. As a result, authors in [32] ran tremendous simulations to find the ideal combination in order to achieve optimal throughput and latency in the network.

For the testbed scenario, they studied a non-standalone scenario where 4G and 5G base stations were at the same tower and the 5G cell was providing mmWave connectivity. There was a building in the simulations to block the mmWave signals in order to mime NLoS states. Then, they tested congestion controls one by one accompanying all schedulers in turn.

The first outcome of the tests could prove the previous conclusions, that blockage can degrade SINR in 5GmmWave but not in LTE.

Moving toward congestion control analyses with schedulers, LIA was deployed through various tests and the combination of the protocol with BLEST had the highest throughput. Nevertheless, ECF had the worst performance among the other four, i.e., LowestRTT, redundant, and

Roundrobin. After BLEST, redundant, LowestRTT, roundrobin, and ECF had the best performance in order. The reason for the redundant scheduler reaching the second rank is that it sends packets through both LTE and 5G, as a consequence, the blockage cannot affect its functionality much. Despite the intelligent mechanism of ECF, as it cannot adapt itself to the intermittent character of the 5G mmWave channel, it finished last among the all schedulers.

The previous trend could be the same when the congestion control algorithm has changed to OLIA. However, the received throughputs for all algorithms were minorly lower than LIA. One interesting outcome was that when the blockage lasts for a long time, ECF stops sending packets through the 5G channel as it assumes that the channel cannot provide any benefits.

The trend for the Balia was the same as the previous ones, but the story was a little different in a way that all schedulers could attain higher throughput compared to LIA and OLIA, and the combination of Balia with BLEST had the highest throughput among all.

When deploying wVegas, there were no significant differences between all the schedulers and they could achieve almost the same low throughput.

In conclusion. as BLEST can provide proper reactions to network changes, it can quickly adapt itself to the new conditions in the network and attained the best functionality among all schedulers and its best performance was for the time deployed with Balia. Sending packets through all available subflows is the key element of ignoring blockage states in the network for redundant and relying on this privilege, this scheduler could lay in the second place. LowestRTT and round-robin didn't exhibit much difference through the simulations. Finally, even though ECF employs an intelligent mechanism, the existing fluctuations in 5G mmWave channels could mislead the algorithm into reaching low throughput and it ended up as the worst scheduler.

In conclusion, despite the fact that the MPTCP deployment over 5G mmWave networks is in its infancy, some research has been done recently to indicate the issues that the protocol encounters in the new cellular generation. The results could reveal that there is a necessity of designing new congestion control and scheduling algorithms for MPTCP in a way that the protocol can handle issues such as blockage when exploited in 5G mmWave networks.

### C. MPTCP PROXY SOLUTIONS

Nowadays, not all servers exploit MPCTP, so what is the solution in this situation? The answer is simple, using proxies between servers and clients utilizing virtualization. Virtualization techniques such as SDN (Software–Defined Networking) [135] and NFV (Network Function Virtualization) [136] are critical enablers in replacing with real hardware components to mime their behavior.

An experimental analysis of MPTCP by means of a socks proxy for non-capable MPTCP servers was done in [137]. The primary purpose of the investigations was to see how well two proxies, including KVM (Kernel-based Virtual Machine), which is a hypervisor-based proxy, and docker, which is a container-based, function when deployed by MPTCP. In the experiments, the scheduler was LowestRTT, and the congestion control mechanism was CUBIC. The main responsibility of the socks proxies placed before the non-MPTCP server is to provide the MPTCP features. The first outcome of the results showed an increased response time because of the relaying process of the proxies and enabling multipathing for clients.

Furthermore, docker could perform better between two proxies because it is lighter than KVM and does not need to run its own operating system. In terms of CPU usage, docker also could outperform KVM by occupying less power of the processing unit. Finally, in the presence of short, medium, and long-lived flows, in all situations, when RTT is not high, docker could perform slightly weaker than the default MPTCP.

In contrast, when the delay between two end-points is high, the performance of docker degrades intensely. However, for MPTCP, this loss in performance is only for short and medium-lived flows, which confirms the expectation from MPTCP in order to adapt well to long and large flows.

To conclude, using proxies to enable MPTCP features in a network depends on the delay between two ends. Between the different proxies, the container-based one could show enhanced functionality to the hypervisor-based one due to its lightness. Finally, when deploying virtualization in a network, additional parameters should be analyzed before the implementation, especially delays between other ends.

### D. REAL-WORLD EXPERIMENTS

Besides experimental and simulation-based analysis, some real-world practice needed to be done to evaluate the performance of MPTCP in practice. A real-world investigation of MPTCP version 0.89v5 [138] over Android 4.4.4 in Nexus 5 was done in [102]. Furthermore, the full Mesh path manager and LowestRTT scheduler were exploited by the MPTCP implementation as it is in the Linux default configuration. A socks proxy was deployed so it could perform the TCP <-> MPTCP transformation where MPTCP capability is not provided.

The testbed incorporated two main scenarios, one for uploading and the other for downloading. There was a Nexus 5 mobile phone deploying WiFi and 3G/4G, connecting to a TCP server through a socks proxy. Different applications containing Facebook, Messenger, Dropbox, and Google Drive were used in the uploading scenario. The first two included routine activities with photo uploading, and the last two were for uploading a 20 MB file covering long connections.

In the downloading scenario, the first twelve websites from Alexa were used. Afterward, a music application called Spotify, then Dailymotion, and YouTube for video streaming were deployed. By considering the duration of the TCP connection throughout the test, 74% of the connections were short flows, which shows that quick TCP connections are the dominant ones in daily activities.

After testing TCP and acquiring the percentage of the different flows, it was the turn for MPTCP to be analyzed in LTE and WiFi networks. We should notice that by default, Android selects the WiFi path as the default one to establish the MPTCP connection. It means that, in some cases, for short-lived flows, cellular subflow establishment was not required, and the data could be transmitted through the WiFi channel. Furthermore, another paramount factor for choosing the channel in Linux is RTT, which is lower in WiFi than cellular most of the time, so if the cwnd is available for WiFi, it will be the preferred subflow. The results showed that when both channels are exploited, data is sent over the WiFi channel most of the time because it has the lowest RTT and the LowestRTT scheduler prefers to opt for the WiFi channel as the transmitting media. This can even be verified when LTE is the default sobflow for connections that last more than two RTTs. In this case, a WiFi subflow can be established, and the LowestRTT scheduler will send data over it.

Besides scheduling, how packets are dispensed over various subflows depends on the congestion control strategy of the protocol in adjusting the cwnd. For conservative cwnd adjustment, packets will be sent over WiFi until a loss happens, then switching the path to the cellular network. On the other hand, aggressive approaches will cause WiFi's cwnd to fill up, then transfer data from the other channel, i.e., cellular. This is why in bulk downloading or uploading, the cwnd for WiFi is filled up fast, and most of the time, cellular subflow is deployed, which was in accommodation with the test results that showed when uploading a file to Dropbox. In this case, the cellular channel was used 91% of the time. Finally, reinjection [139], which is retransmitting a loss packet from another subflow than its original one, was also measured during the test, and the results indicated 0.5% for the uplink and 2% for the download transmissions.

A comprehensive analysis of MPTCP over WiFi, LTE, and 3G was done in [80], and the authors strived to cover different KPIs, including downloading time, loss rate, RTT, and out-of-order delay. The tests were done over different file sizes to mime various short- and long-lived connections. For WiFi, the lowest RTT, and for LTE, the lowest loss rate were the advantages considered in the analysis.

The downloading time for MPTCP short connections could not bring any enhancements as the time for establishing the second path could not be enough. In this case, TCP over WiFi could show the best performance. As the files size increases, having a subflow over LTE assists MPTCP to gain its superiority by reducing the downloading time by utilizing LTE as the low-loss rate channel. LTE was proved to have the lowest loss rate in the tests between the deployed networks compared to WiFi and 3G, which was its main leverage in helping MPTCP. As expected, TCP or MPTCP over 3G had the worst performance due to the high delay of the path. Putting a step ahead, deploying 4-path MPTCP could also enhance the download time, especially in larger sizes. Considering the different congestion control mechanisms employment, LIA, OLIA, uncouple Reno, and TCP could function close to each other with slight differences. However, this will be different as the file size is enlarged.

To see how the congestion avoidance phase functions, very large files also were included in the test, and the infinite value of the slow start threshold for Linux was changed to the TCP's default value, i.e., 64 KB. In this case, WiFi could not be the best path for guaranteeing high performance, and TCP lost the race to MPTCP all the time. Moreover, between MPTCPs, 4-path ones could always attain higher performance, and between them, OLIA is better than LIA but not with much difference. Furthermore, Reno could lower download time by degrading fairness among Coupled and uncoupled ones. One of the interesting results was the portion of each subflow in routing the packets, and as the file size increases, the loss rate of LTE can compensate for its higher RTT and transmits more data than WiFi.

One of the paramount KPIs affecting user experience is latency, especially in time-sensitive applications, as most users exploit routinely. Two primary outcomes were concluded from the results, 1) the latency for WiFi is lower than LTE and 3G. 2) RTT fluctuations in the network can mislead MPTCP in estimating the RTT, which leads to a delay in cwnd adjustment. To put it in another way, RTT variation is one of the issues that can degrade the performance of MPTCP and should be dealt accurately. Moreover, it indicates that some of the network characteristics can have an intense impact on the functionality of MPTCP, which can be distinct based on the deployment scenarios.

The last measured parameter was the out-of-order delay. In order to take out the time, the receiver's buffer was set to a large size so the receiver's window would not be a hurdle. Here, the best combination was for a WiFi/LTE network, as more than 75% of packets were delivered in sequence. In contrast, WiFi/3G suffered from 75% of out-of-order delivered packets. The reason behind this deficiency is the intense difference between the WiFi and 3G channels' characteristics, especially in terms of RTT, which showed itself as a critical player in determining the performance.

To sum up, deploying MPTCP can bring advantages and improve the user experience. However, some network parameters such as RTT and loss rate affect its functionality and should be noticed before the deployment. Furthermore, we should consider that the results will lose their validity by emerging new WiFi networks and loss rate reduction, and a re-evaluation process should be done.

An experimental test over different bandwidths, propagation delay, queuing delay, and loss probability to evaluate MPTCP's performance were done in [140]. In the first step, it was proved that the receiver's buffer size has a significant impact on the attained aggregated throughput, and as it increases, the aggregated throughput will be higher.

In contrast, a smaller buffer size will lead to unaggregated throughput.

Secondly, by increasing loss probability on the bottlenecks, both LIA and OLIA, based on the loss-based congestion control mechanism they have, initiated traffic shifting from the lossy path to the non-lossy one. Because the two protocols try to empty the lossy path, the aggregated throughput is almost equal to zero, showing that throughputs will be similar to the throughput of the best path.

For evaluating load balancing, a scenario of three bottlenecks was exploited, and the result indicated that OLIA is able to perform the best congestion balancing, however, LIA cannot make any balance between the subflows similar to uncoupled NewReno. Nevertheless, OLIA also fails to attain load balancing when the network's BDP increases. As a result, BDP can be one of the principal players in determining MPTCP performance, along with RTT and loss rate. Considering all experiences shows that MPTCP is capable of using multiple routes simultaneously, and it can hone the performance of the network by deploying more than one subflow in a connection. However, different scenarios lay distinct hurdles in the way of MPTCP. For this, in designing new MPTCPs, various parameters such as RTT, BDP, the number of subflows, the deployed technologies in subflows, duration of a connection, i.e., short, medium, or long-lived, being coupled or uncoupled, and the priority for fairness should be taken into account.

## VII. APPROPRIATE SIMULATION ENVIRONMENTS

There are various simulation tools for 5G networks developed by researchers and scientists. Each simulation tool has its own advantages and disadvantages that can be deployed based on the needs and necessities. In this section, we strive to briefly introduce some well-known simulations and present our guidelines for selecting different ones for various scenarios based on a researcher needs.

One of the well-known modules for simulating 5G networks is the MATLAB 5G library [141], which was the based tool for designing and implementing some other simulators. This module can cover a wide range of 5G features such as 5G NR PDSCH (Physical Downlink Shared Channel), TR 38.901 Propagation Channels, OFDM (Orthogonal Frequency-Division Multiplexing) Waveforms with NR Subcarrier Spacings, and many more, which can help for an end-to-end simulation. Not being free and non-modular form for TCP can be counted as the noticeable features of the simulator. In modular simulators, it is easier to access and modify various parts, for example, the application layer's different parts can be easily changed in NS3 through the application module

A MATLAB-based simulator called Vienna LTE is another tool that can be exploited for simulation purposes [142], [143]. Being open-source, the capability of simulating networks included hundreds of nodes, supporting new features such as MIMO (Multiple-Input Multiple-Output), and many more are the superiorities of the simulator. At first, the simulator was for the LTE networks, however, an extension was released for 5G networks [144]. The Vienna 5G System Level Simulator is able to simulate frequencies below 6 GHz or mmWave frequencies up to 100 GHz, as a result, it is capable of creating 5G mmWave networks. The flexibility of the simulator in defining various scenarios and placing UEs and APs arbitrarily among other powerful tools, assist creating heterogeneous networks. More importantly, different types of users can be defined, which is paramount in distinguishing various nodes such as humans, cars, and others.

Furthermore, the blockage issue can be mimed in the simulator, which can be employed to create LoS and NLoS states. Finally, having an active forum [145] with numerous attendances is one of the main leveraging for the simulator to be developed, enhanced, and become more reliable day after day. All mentioned reasons can motivate a researcher to select the Vienna 5G System Level Simulator as the default simulator.

Another simulator for 5G mmWave called K-SimNet was designed by Seoul National University [146]. This simulator has been developed on top of NS3 and inherits the modular-based characteristic of its original. The first aim of K-SimNet is to create an infrastructure for miming 5G stand-alone end-to-end communication, which means that everything in the topology could be 5G ones, including the core, i.e., 5GCN, which was replaced instead of LTE core network, i.e., EPC (Evolved Packet Core). Moreover, it includes other 5G features such as traffic management on multi-RAT, SDN/NFV capabilities for network virtualization, and multi-connectivity. This module is also suitable for simulating 5G mmWabe networks, especially when the deployment of 5GCN is paramount, however, new patches and updated supports need to be provided to keep the module from being obsolete.

When a module can simulate only the link between UEs and Antennas, it is called a link-level simulator. However, by covering large areas and adding the capability of simulating numerous UEs and Antennas, a system-level simulator such as WISE (WIreless Simulator Evolution) [147] can be born. In this way, a vast range of KPIs such as throughput, latency, cell average throughput, packet retransmission number, packet loss rate, handover rate, fairness, and many more can be observed and tracked. As time passes, system-level simulators can fix their places in academics and industry because of their valuable roles in miming real networks.

In the beginning, WISE aimed at simulating LTE networks up to Rel-14 and had some validations such as 3GPP calibration campaigns. By emerging 5G NR and its specification, WISE included the necessary changes to support the new mobile generation capabilities by following the ITU-R five test environments in [148]. Another important characteristic of WISE is its support of 100 GHz, which is very crucial in simulating 5G mmWave networks. Furthermore, WISE modified the beam sweeping to save time and improved the simulation results up to 60%.

To sum up, WISE is an open-source C++ system-level simulator used for academic and industrial purposes. Plus, being updated constantly makes the simulator a suitable choice.

The University of A Coruña proposed a link-layer simulator called GTEC [149] to simulate 4G networks and then extended it to cover 5G cellular networks. The aim of the simulator mainly focused on transceivers functionality miming and channel model simulation with the support of both OFDM and FBMC (Filter Bank MultiCarrier). The modular characteristic of GTEC paves the way for modifying different aspects of the simulator simply and is a powerful tool for researchers who want to broaden the module's functionality. However, more analysis and investigations are needed to prove the simulator's functionality in different situations and deployment scenarios.

A system-level simulation platform was proposed in [150] to simulate different networks simultaneously. By exploiting this module, the deployment of LTE-A and WiFi networks at the same time could be possible, which makes the simulating of heterogeneous networks simpler. In order to attain the aforementioned aims, FDD (frequency division duplex), multiple access techniques, and CSMA/CA were incorporated, so miming the LTE-A and WiFi became feasible. One of the essential downsides of the simulator is supporting a part of the protocol stack for both LTE-A and WiFi, making it hard to have a clear understanding of the whole stack.

Another flaw is that the module is not capable of exploiting mmWave frequency, which is essential in stimulating the coming cellular networks. Moreover, the implementation of the channel does not include the newest 3GPPP channel model. In a nutshell, this module contains a gap to be filled in order to be able to simulate 5G networks.

There is an LTE-EPC simulator module based on NS3 called LENA [151]. This module can imitate both small and macrocells, enables analyzing SON (Self Organized Network), makes the simulation of downlink and uplink, Multi-RAT technology, heterogeneous networks, and many more.

Being open-source, having a supportive team, and striving to keep the module updated [152] are the most important superiorities of LENA. Furthermore, LENA enfolds various layers such as RRC (Radio Resource Control), PDCP (Packet Data Convergence Protocol), RLC, physical layer, MAC layer, different channels, and antenna models.

One of the best modules for simulating 5G mmWave is called ns3-mmWave [153], [154], covering most of the 5G mmWave features. The deployed channel model explanation, i.e., 3GPP channel mode, which also includes mmWave spectrum, can be found in [155], and the information about dual connectivity has been presented in [156] and [157]. A thorough investigation, tutorial, and analysis of the module, including the validation of the functionality and detailed aspects of the simulator, were presented in [86], which is a precious guideline for the researcher who would like to analyze the functionality of 5G mmWave networks. The module can simulate below 6 GHz or up to 100 GHz, i.e., the 3GPP channel model [158], providing a valuable infrastructure to test different situations and deployment scenarios.

**TABLE 4.** A comparison of various simulators.

| Simulator | Advantages | Disadvantages |
|---|---|---|
| **MATLAB [141]** | Covering a lot of 5G features | Not being open-source<br>Not being modular for TCP |
| **Vienna [143], [144]** | Covering a lot of 5G features<br>Open-source<br>Supporting vast networks<br>Supporting from below 6 GHz up to 100 GHz<br>Blockage miming<br>Having an active forum<br>Overall: recommended | NA |
| **K-SimNet [146]** | Covering a lot of 5G features<br>Being modular | Updates and new patches are not released frequently<br>Lack of a supportive team |
| **WISE [147]** | Covering a lot of 5G features<br>Following ITU-R's five test environments principles<br>Supporting up to 100 GHz<br>Open-source<br>Being updated constantly<br>Covering a lot of 5G features<br>Overall: recommended | NA |
| **GTEC [149]** | Modular<br>Channel modeling with the support of OFDM and FBMC | Lack of thorough analysis in order to prove the simulator's functionality |
| **System Level Simulator for 5G networks [150]** | Supporting LTE-A and WiFi | Not supporting most of the 5G features<br>Supporting a part of the protocol stack<br>Not capable of deploying mmWave<br>Not including the newest 3GPP channel model |
| **LENA [151]** | Covering a vast range of networks<br>Open-source<br>Having a supportive team<br>Frequent updates for the module | The lack of maneuverability of the module in 5G networks |
| **Ns3-mmWave [86]** | Covering most of the 5G features<br>Supporting the 3GPP channel model<br>Including frequencies below 6 GHz up to 100 GHz<br>Open-source<br>Modular<br>Blockage miming<br>Supporting dual-connectivity<br>Comprehensive analysis for the module validation<br>Supporting a vast range of topologies and diverse networks<br>Having an active forum with a supportive team<br>Being able to connect to DCE | NA |

Almost most of the 5G network features including MIMO, OFDM-based physical layer, flexible/variable TTI (Transmission Time Interval), TDMA (Time Division Multiple Access) MAC approach, an enhanced RLC layer, cross-layer simulation possibilities, and many more have been embedded in the module. Moreover, the module has the capability of simulating a large number of UEs and Antena along with heterogeneous networks that can be connected to Direct Code Execution [159] to deploy the Linux stack, including MPTCP, as its TCP/IP protocol stack. Besides all mentioned advantages, the ns3-mmWave as a separated fork based on the NS-3 module [160] is open-source, modular, updated regularly, and has a supportive team [161]. As a result, by comparing all the simulators, ns3-mmWave is the recommended module for analyzing, investigating, and testing 5G mmWave networks, as it can fulfill all the needs and necessities.

## VIII. CONCLUSION

With the advent of new devices equipped with different network interfaces ready to be deployed simultaneously, MPTCP is one of the promising approaches to accelerate the movement to the simultaneous use of more than a network at the same time. Besides, 5G, especially along with mmWave for providing a high data rate considering the eMBB use case, and mmWave Wi-Fi communication are getting more attraction these days, thus, adapting MPTCP to these networks can also be beneficial. In this paper, we have provided a survey on the existing MPTCP congestion control and scheduling mechanisms outlining their advantages and disadvantages, which were employed to interfere with whether a protocol can be compatible with cellular networks, especially 5G mmWave, or not. The conclusion revealed that for having a better user experience, there is a necessity of exploiting MPTCP in the coming networks. However, most of the variants are not compatible with 5G mmWave networks, so new protocols should be designed to use the full potential of these networks accompanying other networks. The analyses revealed that MPTCP exploitation can bring some benefits, however, various aspects should be considered before the deployment, which can be found as follow:

- One of the most important criteria that affect the functionality of MPTCP in wireless and cellular networks is the distance between the UE and the base stations, and as it increases, the throughput can be degraded. On the other hand, when the distance is low, MPTCP can gain the benefit of using subflows. However, this general conclusion can be different based on the employed frequency. As a result, when employing or designing a MPTCP, the deployment scenario should be considered, for example, the approach for the urban scenario should be different from the rural one.
- From the congestion control point of view, being coupled or uncoupled is another critical factor. Uncoupled MPTCPs ignore being fair and treat each subfow independently, which leads to achieving high throughput sacrificing fairness. However, coupled ones strive to adhere to the fairness principle, which can damage the aggregate throughput when deployed in high-potential networks such as mmWave ones. As a consequence, based on the needs and necessities, a trade-off between throughput and fairness should be considered. As cellular communication moves toward exploiting high bandwidth, fairness losses its significance in some scenarios, correspondingly, it is paramount to determine if fairness is a precedence factor for the targeted scenario or not. In this case, being coupled or uncoupled can be specified accordingly.
- Sticking to goal three of MPTCP can impair the performance of the protocol in mmWave networks. In this case, MPTCP sees the lossy channel of mmWave as a congested subflow and tries to steer the traffic to the other one, for instance, LTE, which leads to the underutilization of the mmWave. Wherefore, in designing MPTCP for mmWave networks, the intermittent characteristics of channels, which are in contrast to goal three, should be considered.
- The spectrum of the mmWave channel is another paramount factor as it increases, the fluctuating character of the channel can be more intense misleading MPTCP. As an example, the 73 GHz spectrum has more lossy characteristics than the 28 one, which makes it hard for MPTCP to reach high performance. This impact can even be more severe when the distance between the UE and the base station is high.
- The lifetime of the flow is another major element in determining MPTCP functionality. For the short-lived flows, MPTCP can not bring considerable benefits. Even so, as the connection time increases, MPTCP deployment can be advantageous.

These conclusions show that new approaches should be employed in the design of MPTCP for the new cellular communications, especially, for 5G mmWave. They can include extending aggressive TCPs such as HighSpeed or BBR in a coupled manner, defining novel congestion control rules for a coupled MPTCPs, using machine learning-based approaches, or extending QUIC to have MP-QUIC for the congestion control component in order to increase the bandwidth utilization to its full potential. On the other hand, designing new mechanisms for the scheduler by utilizing new schemes such as extending it into a multi-criteria component or using machine learning-based mechanisms is crucial to relieve some problems such as head-of-line blocking, bufferbloating, buffer blocking, and out-of-order problems.

For future work, the main focus should be on the scheduler and congestion control mechanisms of MPTCP to adapt it to various scenarios and conditions. As the scheduler is in charge of choosing the responsible subflow for the transmission, the efforts should aim for schemes that can discriminate dissimilar circumstances in the network such as LoS or NloS ones, and make the decision based on them. To do so, multi-criterion or intelligent mechanisms based on machine learning can be considered. Moreover, beyond end-to-end

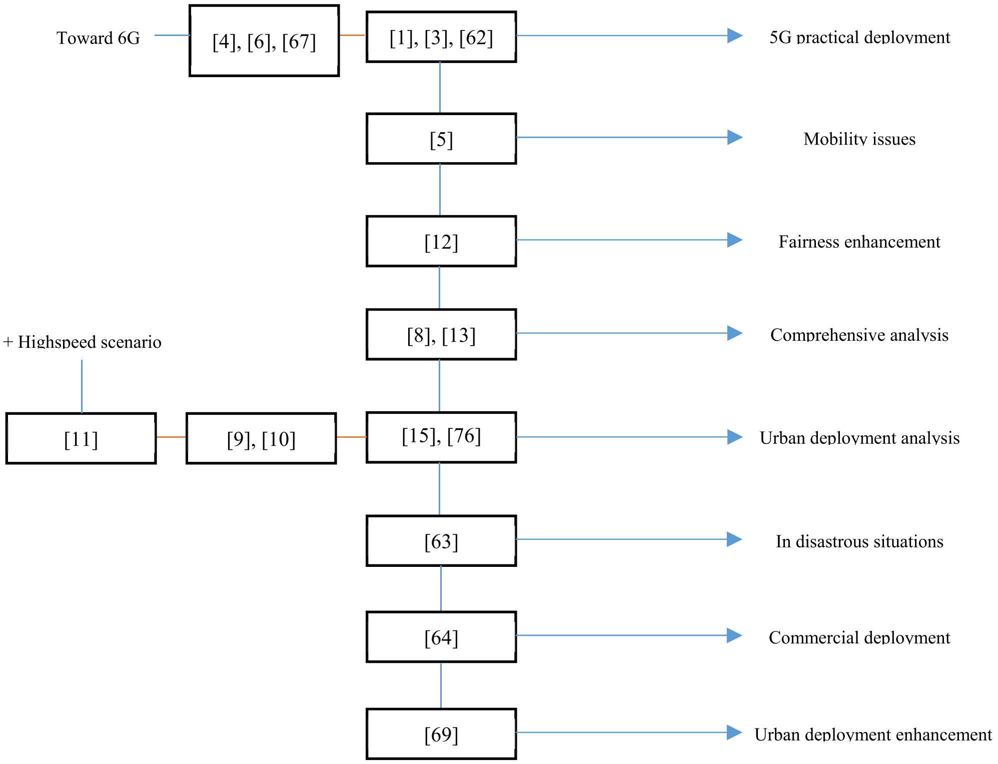

**FIGURE 5.** TCP trend over 5G mmWave networks.

parameters can also be taken into account to provide an open loop sight for the scheduler.From the congestion control point of view, first of all, apt coupled algorithms should be designed for MPTCP to be exploited in 5G mmWave networks along with the other ones. The approach can be sketching a mechanism from scratch or expanding conventional aggressive TCP such as HighSpeed or BBR to adapt them to the coming cellular communication. Another approach for the expansion can be focusing on the ongoing attempt on MP-QUIC to make it more befitting for 5G mmWave networks.

Besides the mentioned explication, deploying machine learning-based approaches in making the congestion control algorithms smarter to tackle the issues such as blockage can be lucrative. Furthermore, like the scheduler, beyond end-to-end parameters can be applied to make the functionality of the MPTCP congestion control component more robust.

## APPENDIX

See Figure 5 for the graphical evolution of TCP toward mmWave networks.

## ACKNOWLEDGMENT

The authors alone are responsible for the content of this article.

## REFERENCES

[1] *The 5G Economy in a Post-COVID-19 Era: The Role of 5G in a Post-Pandemic World Economy*. Accessed: Nov. 7, 2022. [Online]. Available: https://www.qualcomm.com/content/dam/qcomm-martech/dm-assets/documents/qualcomm_5g_economy_in_a_postpandemic_era_report_2020.pdf

[2] Several Authors. (Nov. 2022). *Ericsson Mobility Report*. Publisher Fredrik Jejdling, Ericsson. [Online]. Available: https://www.ericsson.com/4ae28d/assets/local/reports-papers/mobility-report/documents/2022/ericsson-mobility-report-november-2022.pdf

[3] G. Barb and M. Otesteanu, "4G/5G: A comparative study and overview on what to expect from 5G," in *Proc. 43rd Int. Conf. Telecommun. Signal Process. (TSP)*, Jul. 2020, pp. 37–40, doi: 10.1109/TSP49548.2020.9163402.

[4] 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.

[5] M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Mobility management for TCP in mmWave networks," in *Proc. 1st ACM Workshop Millim.-Wave Netw. Sens. Syst. (mmNets)*, vol. 2017, pp. 11–16, doi: 10.1145/3130242.3130243.

[6] H. Tataria, M. Shafi, A. F. Molisch, M. Dohler, H. Sjöland, and F. Tufvesson, "6G wireless systems: Vision, requirements, challenges, insights, and opportunities," 2020, *arXiv:2008.03213*.

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

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

[9] 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, 2021, doi: 10.1109/ACCESS.2021.3063623.

[10] 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.*, Nov. 2020, pp. 85–92, doi: 10.1145/3416011.3424749.

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

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

[13] J. Lorincz, Z. Klarin, and J. Ožegović, "A comprehensive overview of TCP congestion control in 5G networks: Research challenges and future perspectives," *Sensors*, vol. 21, no. 13, p. 4510, Jun. 2021, doi: 10.3390/s21134510.

[14] 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.

[15] 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)*, Apr. 2019, pp. 1–6, doi: 10.1109/WCNC.2019.8886119.

[16] J. Postel, *Transmission Control Protocol*, document STD 7, RFC793, RFC1122, RFC 3168, RFC 6093, RFC 6528, RFC 793, Sep. 1981. [Online]. Available: https://tools.ietf.org/html/rfc793

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

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

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

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

[21] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, "TCP westwood: Bandwidth estimation for enhanced transport over wireless links," in *Proc. 7th Annu. Int. Conf. Mobile Comput. Netw.*, Jul. 2001, pp. 287–297, doi: 10.1145/381677.381704.

[22] 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. 34, no. 2, pp. 25–38, Apr. 2004, doi: 10.1145/997150.997155.

[23] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," in *Proc. IEEE INFOCOM 25th Int. Conf. Comput. Commun.*, 2006, pp. 1–12, doi: 10.1109/INFOCOM.2006.188.

[24] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Netw. Congestion*, vol. 14, pp. 20–53, Dec. 2016, doi: 10.1145/3012426.3022184.

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

[26] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, *TCP Extensions for Multipath Operation With Multiple Addresses*, document RFC 8684, Mar. 2020. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8684

[27] *Apple MPTCP Support*. Accessed: Nov. 2021. [Online]. Available: https://support.apple.com/en-us/HT201373

[28] O. Bonaventure and S. Seo, "Multipath TCP deployments," *IETF J.*, vol. 12, pp. 24–27, Nov. 2016.

[29] Apple, "Advances in networking," in *Proc. Worldwide Developers Conf. (WWDC)*, Jun. 2017. Accessed: Dec. 2021. [Online]. Available: https://developer.apple.com/videos/wwdc2017

[30] C. Raiciu, M. Handly, and D. Wischik, *Coupled Congestion Control for Multipath Transport Protocols*, document IETF RFC 6356, Oct. 2011. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6356

[31] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, "Design, implementation and evaluation of congestion control for multipath TCP," in *Proc. 8th USENIX Conf. Netw. Syst. Design Implementation*, Mar. 2011, pp. 99–112.

[32] M. I. T. Lubna and C. YZ, "Performance evaluation of MPTCP on simultaneous use of 5G and 4G networks," *Sensors*, vol. 22, pp. 856–858, Oct. 2022, doi: 10.3390/s22197509.

[33] 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)*, May 2017, pp. 343–348, doi: 10.1109/INFCOMW.2017.8116400.

[34] F. Aschenbrenner, T. Shreedhar, O. Gasser, N. Mohan, and J. Ott, "From single lane to highways: Analyzing the adoption of multipath TCP in the internet," in *Proc. IFIP Netw. Conf. (IFIP Netw.)*, Espoo Helsinki, Finland, 2021, pp. 1–9, doi: 10.23919/IFIPNetworking52078.2021.9472785.

[35] 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.

[36] S. Aggarwal, M. Ghoshal, P. Banerjee, and D. Koutsonikolas, "An experimental study of the performance of IEEE 802.11ad in smartphones," *Comput. Commun.*, vol. 169, pp. 220–231, 2021, doi: 10.1016/j.comcom.2021.01.006.

[37] S. Aggarwal, M. Ghoshal, P. Banerjee, D. Koutsonikolas, and J. Widmer, "802.11ad in smartphones: Energy efficiency, spatial reuse, and impact on applications," in *Proc. IEEE INFOCOM*, May 2021, pp. 1–10, doi: 10.1109/INFOCOM42981.2021.9488763.

[38] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "On load distribution over multipath networks," *IEEE Commun. Surveys Tuts.*, vol. 14, no. 3, pp. 662–680, 3rd Quart., 2012, doi: 10.1109/SURV.2011.082511.00013.

[39] A. L. Ramaboli, O. E. Falowo, and A. H. Chan, "Bandwidth aggregation in heterogeneous wireless networks: A survey of current approaches and issues," *J. Netw. Comput. Appl.*, vol. 35, no. 6, pp. 1674–1690, 2012.

[40] S. K. Singh, T. Das, and A. Jukan, "A survey on internet multipath routing and provisioning," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2157–2175, 4th Quart., 2015.

[41] J. Qadir, A. Ali, K.-L. A. Yau, A. Sathiaseelan, and J. Crowcroft, "Exploiting the power of multiplicity: A holistic survey of network-layer multipath," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2176–2213, 4th Quart., 2015.

[42] M. Li, A. Lukyanenko, Z. Ou, A. Ylä-Jääski, S. Tarkoma, M. Coudron, and S. Secci, "Multipath transmission for the internet: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2887–2925, 4th Quart., 2016.

[43] C. Xu, J. Zhao, and G.-M. Muntean, "Congestion control design for multipath transport protocols: A survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 4, pp. 2948–2969, 4th Quart., 2016.

[44] B. Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, "Packet scheduling in multipath TCP: Fundamentals, lessons, and opportunities," *IEEE Syst. J.*, vol. 15, no. 1, pp. 1445–1457, Mar. 2021, doi: 10.1109/JSYST.2020.2965471.

[45] M.-T. Suer, C. Thein, H. Tchouankem, and L. Wolf, "Multi-connectivity as an enabler for reliable low latency communications—An overview," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 156–169, 1st Quart., 2020.

[46] H. Wu, S. Ferlin, G. Caso, O. Alay, and A. Brunstrom, "A survey on multipath transport protocols towards 5G access traffic steering, switching and splitting," *IEEE Access*, vol. 9, pp. 164417–164439, 2021, doi: 10.1109/ACCESS.2021.3134261.

[47] *System Architecture for the 5G System (5GS)*, document TS 23.501 v16.4.0, 3GPP, Mar. 2020.

[48] J. Ha and Y.-I. Choi, "Support of a multi-access session in 5G mobile network," in *Proc. 25th Asia–Pacific Conf. Commun. (APCC)*, Nov. 2019, pp. 378–383, doi: 10.1109/APCC47188.2019.9026455.

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

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

[51] J. Chu, N. Dukkipati, Y. Cheng, and M. Mathis, *Increasing TCP's Initial Window*, document RFC 6928, Apr. 2013.

[52] T. Kelly, "Scalable TCP: Improving performance in highspeed wide area networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 2, pp. 9–83, Apr. 2003, doi: 10.1145/956981.956989.

[53] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 2514–2524, doi: 10.1109/INFCOM.2004.1354672.

[54] D. Leith. (Oct. 2008). *H-TCP: TCP Congestion Control for hIGH bANDWIDTH dELAY pRODUCT pATHS*. IETF Internet Draft. Accessed: Oct. 25, 2017. [Online]. Available: https://tools.ietf.org/html/draft-leith-tcp-htcp-06

[55] G. Marfa, C. Palazzi, G. Pau, M. Gerla, M. Y. Sanadidi, and M. Roccetti, "TCP Libra: Exploring RTT-fairness for TCP," UCLA Comput. Sci. Dept., Los Angeles, CA, USA, Tech. Rep. UCLA-CSD TR-050037, 2005.

[56] C. Caini and R. Firrincieli, "TCP hybla: A TCP enhancement for heterogeneous networks," *Int. J. Satell. Commun. Netw.*, vol. 22, no. 5, pp. 547–566, Sep. 2004.

[57] L. S. Brakmo, S. W. O'Malley, and L. L. Peterson, "TCP Vegas: New techniques for congestion detection and avoidance," *SIGCOMM Comput. Commun. Rev.*, vol. 24, no. 4, pp. 24–35, Oct. 1994, doi: 10.1145/190809.190317.

[58] J. Sing and B. Soh, "TCP new Vegas: Improving the performance of TCP Vegas over high latency links," in *Proc. 4th IEEE Int. Symp. Netw. Comput. Appl.*, Cambridge, MA, USA, Jul. 2005, pp. 73–80, doi: 10.1109/NCA.2005.52.

[59] D. Kliazovich, F. Granelli, and D. Miorandi, "Logarithmic window increase for TCP Westwood+ for improvement in high speed, long distance networks," *Comput. Netw.*, vol. 52, no. 12, pp. 2395–2410, Aug. 2008, doi: 10.1016/j.comnet.2008.04.018.

[60] 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. Protocols Fast Long Distance Netw.*, Feb. 2006, pp. 87–91.

[61] 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, CA, USA, Apr. 2007, pp. 31–36.

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

[63] 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, 2019, doi: 10.1109/ACCESS.2019.2945582.

[64] A. Narayanan, E. Ramadan, J. Carpenter, Q. Liu, Y. Liu, F. Qian, and Z.-L. Zhang, "A first look at commercial 5G performance on smartphones," 2019, *arXiv:1909.07532*.

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

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

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

[68] T. Zhang and S. Mao, "Machine learning for end-to-end congestion control," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 52–57, Jun. 2020, doi: 10.1109/MCOM.001.1900509.

[69] R. Poorzare and A. C. Auge, "FB-TCP: A 5G mmWave friendly TCP for urban deployments," *IEEE Access*, vol. 9, pp. 82812–82832, 2021, doi: 10.1109/ACCESS.2021.3087239.

[70] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *TCP Extensions for Multipath Operation With Multiple Addresses*, document IETF MPTCP Proposal, 2009.

[71] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, *Architectural Guidelines for Multipath TCP Development*, document IETF RFC 6182, Mar. 2011. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6182

[72] A. Ford, C. Raiciu, M. Handley, and O. Bonaventure, *TCP Extensions for Multipath Operation with Multiple Addresses*, document IETF RFC 6824, Jan. 2013. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc6824

[73] L. Ding, Y. Tian, T. Liu, Z. Wei, and X. Zhang, "Understanding commercial 5G and its implications to (Multipath) TCP," *Comput. Netw.*, vol. 198, Oct. 2021, Art. no. 108401, doi: 10.1016/j.comnet.2021.108401.

[74] S. R. Pokhrel and A. Walid, "Learning to harness bandwidth with multipath congestion control and scheduling," *IEEE Trans. Mobile Comput.*, vol. 22, no. 2, pp. 996–1009, Feb. 2023, doi: 10.1109/TMC.2021.3085598.

[75] I. Syrytsin, S. Zhang, G. F. Pedersen, and A. S. Morris, "User-shadowing suppression for 5G mm-wave mobile terminal antennas," *IEEE Trans. Antennas Propag.*, vol. 67, no. 6, pp. 4162–4172, Jun. 2019, doi: 10.1109/TAP.2019.2905685.

[76] 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)*, May 2019, pp. 1–7, doi: 10.1109/ICC.2019.8761718.

[77] C. Paasch and O. Bonaventure, "Multipath TCP," *Commun. ACM*, vol. 57, no. 4, pp. 51–57, Apr. 2014, doi: 10.1145/2578901.

[78] F. P. Kelly, "Mathematical modelling of the internet," in *Mathematics Unlimited-2001 and Beyond*. Berlin, Germany: Springer-Verlag, 2001, pp. 685–702.

[79] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, "MPTCP is not Pareto-optimal: Performance issues and a possible solution," *IEEE/ACM Trans. Netw.*, vol. 21, no. 5, pp. 1651–1665, Oct. 2013, doi: 10.1109/TNET.2013.2274462.

[80] Y.-C. Chen, Y.-S. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, "A measurement-based study of multipath TCP performance over wireless networks," Univ. Massachusetts, Amherst, MA, USA, Tech. Rep. UM-CS-2013-018, 2013.

[81] Q. Peng, A. Walid, J. Hwang, and S. H. Low, "Multipath TCP: Analysis, design, and implementation," *IEEE/ACM Trans. Netw.*, vol. 24, no. 1, pp. 596–609, Feb. 2016, doi: 10.1109/TNET.2014.2379698.

[82] T. Lubna, I. Mahmud, and Y.-Z. Cho, "D-LIA: Dynamic congestion control algorithm for MPTCP," *ICT Exp.*, vol. 6, no. 4, pp. 263–268, Dec. 2020, doi: 10.1016/j.icte.2020.03.005.

[83] S. R. Pokhrel and M. Mandjes, "Improving multipath TCP performance over WiFi and cellular networks: An analytical approach," *IEEE Trans. Mobile Comput.*, vol. 18, no. 11, pp. 2562–2576, Nov. 2019, doi: 10.1109/TMC.2018.2876366.

[84] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, *Low Extra Delay Background Transport (LEDBAT), Internet Engineering Task Force (IETF)*, document RFC6817, Dec. 2012. [Online]. Available: https://tools.ietf.org/html/rfc6817

[85] Y. Liu, X. Qin, T. Zhu, X. Chen, and G. Wei, "BESS: BDP estimation based slow start algorithm for MPTCP in mmWave-LTE networks," in *Proc. IEEE 88th Veh. Technol. Conf. (VTC-Fall)*, Aug. 2018, pp. 1–5, doi: 10.1109/VTCFall.2018.8690815.

[86] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-end simulation of 5G mmWave networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2237–2263, 3rd Quart., 2018, doi: 10.1109/COMST.2018.2828880.

[87] C. Lee, J. Jung, and J.-M. Chung, "DEFT: Multipath TCP for high speed low latency communications in 5G networks," *IEEE Trans. Mobile Comput.*, vol. 20, no. 12, pp. 3311–3323, Dec. 2021, doi: 10.1109/TMC.2020.3000041.

[88] 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.

[89] Y. Cao, M. Xu, and X. Fu, "Delay-based congestion control for multipath TCP," in *Proc. 20th IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2012, pp. 1–10, doi: 10.1109/ICNP.2012.6459978.

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

[91] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, "Experimental evaluation of multipath TCP schedulers," in *Proc. ACM SIGCOMM Workshop Capacity Sharing Workshop*, Aug. 2014, pp. 27–32, doi: 10.1145/2630088.2631977.

[92] B. Zong, C. Fan, X. Wang, X. Duan, B. Wang, and J. Wang, "6G technologies: Key drivers, core requirements, system architectures, and enabling technologies," *IEEE Veh. Technol. Mag.*, vol. 14, no. 3, pp. 18–27, Sep. 2019, doi: 10.1109/MVT.2019.2921398.

[93] *mptcp_v0.96*. Accessed: Nov. 2022. [Online]. Available: https://github.com/multipath-tcp/mptcp/commits/mptcp_v0.96

[94] H. X. Nguyen and M. Roughan, "Rigorous statistical analysis of internet loss measurements," *IEEE/ACM Trans. Netw.*, vol. 21, no. 3, pp. 734–745, Jun. 2013, doi: 10.1109/TNET.2012.2207915.

[95] S. Biswas, ''Large-scale measurements of wireless network behavior,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 45, no. 4, pp. 153–165, 2015, doi: 10.1145/2829988.2787489.

[96] C. Lee, S. Song, H. Cho, G. Lim, and J.-M. Chung, ''Optimal multipath TCP offloading over 5G NR and LTE networks,'' *IEEE Wireless Commun. Lett.*, vol. 8, no. 1, pp. 293–296, Feb. 2019, doi: 10.1109/LWC.2018.2870595.

[97] R. Stewart, *Stream Control Transmission Protocol*, document RFC 4960, Sep. 2007. [Online]. https://datatracker.ietf.org/doc/html/rfc4960

[98] J. R. Iyengar, P. D. Amer, and R. Stewart, ''Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths,'' *IEEE/ACM Trans. Netw.*, vol. 14, no. 5, pp. 951–964, Oct. 2006, doi: 10.1109/TNET.2006.882843.

[99] G. Sarwar, R. Boreli, E. Lochin, A. Mifdaoui, and G. Smith, ''Mitigating receiver's buffer blocking by delay aware packet scheduling in multipath data transfer,'' in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2013, pp. 1119–1124, doi: 10.1109/WAINA.2013.80.

[100] F. Yang, Q. Wang, and P. D. Amer, ''Out-of-order transmission for in-order arrival scheduling for multipath TCP,'' in *Proc. 28th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, May 2014, pp. 749–752, doi: 10.1109/WAINA.2014.122.

[101] P. Hurtig, K.-J. Grinnemo, A. Brunstrom, S. Ferlin, O. Alay, and N. Kuhn, ''Low-latency scheduling in MPTCP,'' *IEEE/ACM Trans. Netw.*, vol. 27, no. 1, pp. 302–315, Feb. 2019, doi: 10.1109/TNET.2018.2884791.

[102] Q. De Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, ''Observing real smartphone applications over multipath TCP,'' *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 88–93, Mar. 2016, doi: 10.1109/MCOM.2016.7432153.

[103] S. Ferlin, O. Alay, O. Mehani, and R. Boreli, ''BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks,'' in *Proc. IFIP Netw. Conf. (IFIP Netw.) Workshops*, May 2016, pp. 431–439, doi: 10.1109/IFIPNetworking.2016.7497206.

[104] B. Han, F. Qian, S. Hao, and L. Ji, ''An anatomy of mobile web performance over multipath TCP,'' in *Proc. 11th ACM Conf. Emerg. Netw. Exp. Technol.*, Dec. 2015, p. 8, doi: 10.1145/2716281.2836090.

[105] S. Ferlin, T. Dreibholz, and O. Alay, ''Multi-path transport over heterogeneous wireless networks: Does it really pay off?'' in *Proc. IEEE Global Commun. Conf.*, Dec. 2014, pp. 4807–4813, doi: 10.1109/GLOCOM.2014.7037567.

[106] Y.-S. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, ''ECF: An MPTCP path scheduler to manage heterogeneous paths,'' in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 147–159, doi: 10.1145/3143361.3143376.

[107] S. Deng, R. Netravali, A. Sivaraman, and H. Balakrishnan, ''WiFi, LTE, or both?: Measuring multi-homed wireless Internet performance,'' in *Proc. Conf. Internet Meas. Conf.*, Nov. 2014, pp. 181–194, doi: 10.1145/2663716.2663727.

[108] S.-I. Sou and Y.-T. Peng, ''Performance modeling for multipath mobile data offloading in cellular/Wi-Fi networks,'' *IEEE Trans. Commun.*, vol. 65, no. 9, pp. 3863–3875, Sep. 2017, doi: 10.1109/TCOMM.2017.2713378.

[109] H. Kim and K. Chung, ''Design of adaptive offloading framework for video streaming over 5G NR with multipath TCP,'' in *Proc. IEEE 8th Global Conf. Consum. Electron. (GCCE)*, Oct. 2019, pp. 727–728, doi: 10.1109/GCCE46687.2019.9015350.

[110] S. K. Saha, S. Aggarwal, R. Pathak, D. Koutsonikolas, and J. Widmer, ''MuSher: An agile multipath-TCP scheduler for dual-band 802.11ad/AC wireless LANs,'' in *Proc. 25th Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2019, pp. 1–6, doi: 10.1145/3300061.3345435.

[111] *Netgear Nighthawk X10 Smart WiFi Router*. Accessed: Dec. 2021. [Online]. https://www. netgear.com/landings/ad7200/

[112] *TP-Link Talon AD7200 Multi-Band WiFi Router*. Accessed: Dec. 2021 [Online]. http://www.tp-link.com/us/products/details/cat-5506_AD7200.html

[113] K. Nguyen, M. G. Kibria, K. Ishizu, and F. Kojima, ''Feasibility study of providing backward compatibility with MPTCP to WiGig/IEEE 802.11ad,'' in *Proc. IEEE 86th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2017, pp. 1–5, doi: 10.1109/VTCFall.2017.8288189.

[114] S. Sur, I. Pefkianakis, X. Zhang, and K.-H. Kim, ''WiFi-assisted 60 GHz wireless networks,'' in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 28–41, doi: 10.1145/3117811.3117817.

[115] S. K. Saha, R. Shyamsunder, N. M. Prakash, H. Assasa, A. Loch, D. Koutsonikolas, and J. Widmer, ''Poster: Can MPTCP improve performance for dual-band 60 GHz/5 GHz clients?'' in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, Oct. 2017, pp. 573–575, doi: 10.1145/3117811.3131248.

[116] M. M. Roselló, ''Multi-path scheduling with deep reinforcement learning,'' in *Proc. Eur. Conf. Netw. Commun. (EuCNC)*, Jun. 2019, pp. 400–405, doi: 10.1109/EuCNC.2019.8802063.

[117] Q. De Coninck and O. Bonaventure, ''Multipath QUIC: Design and evaluation,'' in *Proc. 13th Int. Conf. Emerg. Netw. Exp. Technol.*, Nov. 2017, pp. 160–166, doi: 10.1145/3143361.3143370.

[118] J. Iyengar and M. Thomson. (May 2021). *QUIC: A UDP-Based Multiplexed and Secure Transport. Work in Progress. Draft-IETF-QUIC-Transport-29*. [Online]. Available: https://datatracker.ietf.org/doc/draft-ietf-quictransport/

[119] *Programmable MultiPath TCP Scheduling*. Accessed: Sep. 2022. [Online]. Available: https://progmp.net/

[120] A. Frömmgen, A. Rizk, T. Erbshäußer, M. Weller, B. Koldehofe, A. Buchmann, and R. Steinmetz, ''A programming model for application-defined multipath TCP scheduling,'' in *Proc. 18th ACM/IFIP/USENIX Middleware Conf.*, Dec. 2017, pp. 134–146, doi: 10.1145/3135974.3135979.

[121] P. S. Kumar, N. Fatima, and P. Saxena, ''Performance analysis of multipath transport layer schedulers under 5G/B5G hybrid networks,'' in *Proc. 14th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2022, pp. 658–666, doi: 10.1109/COMSNETS53615.2022.9668483.

[122] B. Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, ''Alternative scheduling decisions for multipath TCP,'' *IEEE Commun. Lett.*, vol. 21, no. 11, pp. 2412–2415, Nov. 2017, doi: 10.1109/LCOMM.2017.2740918.

[123] K. Noda and Y. Ito, ''Proposal of multi-path TCP packet scheduler to adjust trade-off between QoS fluctuation and throughput for WebQoE improvement,'' in *Proc. IEEE 4th Int. Conf. Comput. Commun. Syst. (ICCCS)*, Feb. 2019, pp. 493–496, doi: 10.1109/CCOMS.2019.8821657.

[124] B. Felix, I. Steuck, A. Santos, S. Secci, and M. Nogueira, ''Redundant packet scheduling by uncorrelated paths in heterogeneous wireless networks,'' in *Proc. IEEE Symp. Comput. Commun. (ISCC)*, Jun. 2018, pp. 498–503, doi: 10.1109/ISCC.2018.8538641.

[125] J. Luo, X. Su, and B. Liu, ''A reinforcement learning approach for multipath TCP data scheduling,'' in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 276–280, doi: 10.1109/CCWC.2019.8666496.

[126] G. N. Vivekananda and P. C. Reddy, ''Critical analysis of cross-layer approach,'' in *Proc. Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Oct. 2015, pp. 12–16, doi: 10.1109/ICGCIoT.2015.7380419.

[127] T. Shreedhar, N. Mohan, S. K. Kaul, and J. Kangasharju, ''QAware: A cross-layer approach to MPTCP scheduling,'' in *Proc. IFIP Netw. Conf. (IFIP Networking) Workshops*, May 2018, pp. 1–9, doi: 10.23919/IFIPNetworking.2018.8696843.

[128] *5G.322, KT 5th Generation Radio Access, Radio Link Control Protocol (5G Pre-Specifications)*, document Korea Telecom (KT) 5G-SIG,. 2016.

[129] D. Kaspar, K. Evensen, A. F. Hansen, P. Engelstad, P. Halvorsen, and C. Griwodz, ''An analysis of the heterogeneity and IP packet reordering over multiple wireless networks,'' in *Proc. IEEE Symp. Comput. Commun.*, Jul. 2009, pp. 637–642, doi: 10.1109/ISCC.2009.5202249.

[130] A. Alheid, A. Doufexi, and D. Kaleshi, ''A study on MPTCP for tolerating packet reordering and path heterogeneity in wireless networks,'' in *Proc. Wireless Days (WD)*, Mar. 2016, pp. 1–7, doi: 10.1109/WD.2016.7461454.

[131] S. Floyd, J. Mahdavi, M. Mathis, and M. Podolsky, *An Extension to the Selective Acknowledgement (SACK)*, document RFC 2883, Jul. 2000. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc2883

[132] R. Ludwig and R. H. Katz, ''The eifel algorithm: Making TCP robust against spurious retransmissions,'' *ACM SIGCOMM Comput. Commun. Rev.*, vol. 30, no. 1, pp. 30–36, Jan. 2000, doi: 10.1145/505688.505692.

[133] F. Wang and Y. Zhang, ''Improving TCP performance over mobile ad-hoc networks with out-of-order detection and response,'' in *Proc. 3rd ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, Jun. 2002, pp. 217–225, doi: 10.1145/513800.513827.

[134] P. Sarolahti, M. Kojo, and K. Yamamoto, *Forward RTO-Recovery (FRTO): An Algorithm for Detecting Spurious Retransmission Timeouts With TCP*, document RFC 5682, Sep. 2009. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc5682

[135] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Commun. Mag.*, vol. 51, no. 2, pp. 114–119, Feb. 2013, doi: 10.1109/MCOM.2013.6461195.

[136] H. Hawilo, A. Shami, M. Mirahmadi, and R. Asal, "NFV: State of the art, challenges, and implementation in next generation mobile networks (vEPC)," *IEEE Netw.*, vol. 28, no. 6, pp. 18–26, Nov./Dec. 2014, doi: 10.1109/MNET.2014.6963800.

[137] S. Chung, S. Moon, and S. Kim, "The virtualized MPTCP proxy performance in cellular network," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2017, pp. 703–707, doi: 10.1109/ICUFN.2017.7993881.

[138] *MultiPath TCP Linux Kernel Implementation*. Accessed: Nov. 23, 2021. [Online]. Available: http://www.multipath-tcp.org/

[139] C. Raiciu, "How hard can it be? Designing and implementing a deployable multipath TCP," in *Proc. 9th USENIX Conf. Netw. Syst. Design Implement.*, Apr. 2012, pp. 29–42.

[140] C. Paasch, R. Khalili, and O. Bonaventure, "On the benefits of applying experimental design to improve multipath TCP," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol.*, Dec. 2013, pp. 393–398, doi: 10.1145/2535372.2535403.

[141] *MATLAB. 5G Library for LTE System Toolbox*. Accessed: Feb. 20, 2020. [Online]. Available: https://www.mathworks.com/products/5g.html

[142] C. Mehlführer, J. Colom Ikuno, M. Šimko, S. Schwarz, M. Wrulich, and M. Rupp, "The Vienna LTE simulators–Enabling reproducibility in wireless communications research," *EURASIP J. Adv. Signal Process.*, vol. 2011, no. 1, pp. 29–42, Jul. 2011, doi: 10.1186/1687-6180-2011-29.

[143] *Vienna LTE-A Simulators*, Accessed: Dec. 26, 2021. [Online]. Available: https://www.nt.tuwien.ac.at/research/mobile-communications/vccs/vienna-lte-a-simulators/

[144] M. K. Müller, F. Ademaj, T. Dittrich, A. Fastenbauer, B. Ramos Elbal, A. Nabavi, L. Nagel, S. Schwarz, and M. Rupp, "Flexible multi-node simulation of cellular mobile communications: The Vienna 5G system level simulator," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, p. 227, Sep. 2018, doi: 10.1186/s13638-018-1238-7.

[145] Institute of Telecommunications, T. U. Wien, *Vienna Cellular Communications Simulators Forum*. Accessed: Dec. 26, 2021. [Online]. Available: https://www.nt.tuwien.ac.at/research/mobile-communications/forums/

[146] S. Choi, J. Song, J. Kim, S. Lim, S. Choi, T. T. Kwon, and S. Bahk, "5G K-SimNet: End-to-end performance evaluation of 5G cellular systems," in *Proc. 16th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2019, pp. 1–6, doi: 10.1109/CCNC.2019.8651686.

[147] C.-K. Jao, C.-Y. Wang, T.-Y. Yeh, C.-C. Tsai, L.-C. Lo, J.-H. Chen, W.-C. Pao, and W.-H. Sheen, "WiSE: A system-level simulator for 5G mobile networks," *IEEE Wireless Commun.*, vol. 25, no. 2, pp. 4–7, Apr. 2018, doi: 10.1109/MWC.2018.8352614.

[148] *Requirements, Evaluation Criteria and Submission Templates for the Development of IMT-2020*, document ITU-R Report ITU-R M.2411, ITU-R WP 5D, Nov. 2017.

[149] T. Dominguez-Bolano, J. Rodriguez-Pineiro, J. A. Garcia-Naya, and L. Castedo, "The GTEC 5G link-level simulator," in *Proc. 1st Int. Workshop Link-Syst. Level Simulations (IWSLS)*, Jul. 2016, pp. 1–6, doi: 10.1109/IWSLS.2016.7801585.

[150] M. Liu, P. Ren, Q. Du, W. Ou, X. Xiong, and G. Li, "Design of system-level simulation platform for 5G networks," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Jul. 2016, pp. 1–6, doi: 10.1109/ICCChina.2016.7636796.

[151] N. Baldo, M. Miozzo, M. Requena-Esteso, and J. Nin-Guerrero, "An open source product-oriented LTE network simulator based on Ns-3," in *Proc. 14th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Oct. 2011, pp. 293–298, doi: 10.1145/2068897.2068948.

[152] CTTC. *LTE-EPC Network Simulator*. Accessed: Jan. 2020. [Online]. Available: http://networks.cttc.es/mobile-networks/software-tools/lena/

[153] M. Mezzavilla, S. Dutta, M. Zhang, M. R. Akdeniz, and S. Rangan, "5G mmWave module for the Ns-3 network simulator," in *Proc. 18th ACM Int. Conf. Modeling, Anal. Simulation Wireless Mobile Syst.*, Nov. 2015, pp. 283–290, doi: 10.1145/2811587.2811619.

[154] 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)*, Jul. 2017, pp. 1–6, doi: 10.1109/SPAWC.2017.8227746.

[155] 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 (WNS)*, Jun. 2017, pp. 71–78, doi: 10.1145/3067665.3067678.

[156] M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Improved handover through dual connectivity in 5G mmWave mobile networks," *IEEE J. Sel. Areas Commun.*, vol. 35, no. 9, pp. 2069–2084, Sep. 2017, doi: 10.1109/JSAC.2017.2720338.

[157] 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. Simul. Tools Techn. (SIMUTOOLS), Prague, Czech Republic, Aug.*, vol. 2016, pp. 118–123, doi: 10.5555/3021426.3021445.

[158] *Study on Channel Model for Frequency Spectrum Above 6 GHz, V14.2.0*, document TR 38.900, 3GPP, Sophia Antipolis, France, 2017.

[159] H. Tazaki, F. Uarbani, E. Mancini, M. Lacage, D. Camara, T. Turletti, and W. Dabbous, "Direct code execution: Revisiting library OS architecture for reproducible network experiments," in *Proc. 9th ACM Conf. Emerg. Netw. Exp. Technol.*, Dec. 2013, pp. 217–228, doi: 10.1145/2535372.2535374.

[160] *mmWave Cellular Network Simulator*. Accessed: Dec. 31, 2021. [Online]. Available: https://apps.nsnam.org/app/mmwave/

[161] *Ns3-mmWave*. Accessed: Dec. 31, 2021. [Online]. Available: https://github.com/nyuwireless-unipd/ns3-mmwave

**REZA POORZARE** (Member, IEEE) received the B.S. and M.S. degrees in computer engineering from the Azad University of Iran, in 2010 and 2014, respectively, and the Ph.D. degree in network engineering from Universitat Politècnica de Catalunya, Barcelona, Spain, in 2022. Currently, he is a Postdoctoral Researcher with the Data-Centric Software Systems (DSS) Research Group, Institute of Applied Research, Karlsruhe University of Applied Sciences, Karlsruhe, Germany. His research interests include 5G, mmWave, wireless mobile networks, TCP, MPTCP, congestion control, and artificial intelligence.

**OLIVER P. WALDHORST** (Member, IEEE) received the Dipl.-Inf. (comparable to M.Sc. degree in computer science) and Ph.D. degrees in computer science from the University of Dortmund, Germany in 2000 and 2005, respectively, and the Habilitation (postdoctoral lecture qualification) degree from the Computer Science Department, Karlsruhe Institute of Technology (KIT), Germany, in February 2011. During his Ph.D., he worked as a Research Assistant at the University of Dortmund, Germany, from April 2000 to December 2005, and then as a Postdoctoral Researcher at the University of Leipzig, Germany, from January 2006 to August 2006. He was a Senior Researcher and a Lecturer at the Institute of Telematics, KIT, from September 2006 to August 2013. From September 2009 to February 2010, he visited the ECE Department's Communication Group at the University of Toronto, ON, Canada, as a Researcher. From April 2011 to March 2012, he was a Visiting Professor at the Integrated Communications Systems Group, Technical University of Ilmenau, Germany. He developed connected car communication architectures for Daimler AG, Sindelfingen, Germany, from September 2013 to August 2016. In September 2016, he joined the Karlsruhe University of Applied Sciences, Germany, where he is currently a Professor of computer and communication networks with the Faculty for Computer Science and Business Information Systems and a member of the Data-Centric Software Systems (DSS) Research Group, Institute of Applied Research. His research interests include communication protocols, services, and distributed applications in mobile and heterogeneous environments and performance modeling and analysis of communication systems.

• • •