This paper has been accepted for publication in
the IEEE Communication Magazine

Will TCP work in mmWave 5G Cellular Networks?

Menglei Zhang[†], Michele Polese[*], Marco Mezzavilla[†],
Jing Zhu[◇], Sundeep Rangan[†], Shivendra Panwar[†], Michele Zorzi[*]
[†]NYU Wireless, New York University, NY, USA - e-mail: {menglei, mezzavilla, srangan, panwar}@nyu.edu
[*]Department of Information Engineering, University of Padova, Italy - e-mail: {polesemi, zorzi}@dei.unipd.it
[◇]Intel Corporation - e-mail: jing.z.zhu@intel.com

***Abstract*—The vast available spectrum in the millimeter wave (mmWave) bands offers the possibility of multi-Gbps data rates for fifth generation (5G) cellular networks. However, mmWave capacity can be highly intermittent due to the vulnerability of mmWave signals to blockages and delays in directional searching. Such highly variable links present unique challenges for adaptive control mechanisms in transport layer protocols and end-to-end applications. This paper considers the fundamental question of whether TCP – the most widely used transport protocol – will work in mmWave cellular systems. The paper provides a comprehensive simulation study of TCP considering various factors such as the congestion control algorithm, including the recently proposed TCP BBR, edge vs. remote servers, handover and multi-connectivity, TCP packet size and 3GPP-stack parameters. We show that the performance of TCP on mmWave links is highly dependent on different combinations of these parameters, and identify the open challenges in this area.**

***Index Terms*—TCP, Congestion Control, BBR, mmWave, 5G, Cellular, Blockage, ns-3**

I. Introduction

End-to-end connectivity over the internet largely relies on transport protocols that operate above the network layer. The most widely used transport protocol is the Transmission Control Protocol (TCP), designed in the 1980s [1] to offer reliable packet delivery and sending rate control to prevent congestion in the network. Reliability is accomplished with receiver's acknowledgments (ACKs) fed back to the sender, which retransmits packets if needed, while rate control is achieved by dynamically adjusting the congestion window, i.e., the maximum amount of data that the sender can transmit without receiving ACKs. Several Congestion Control (CC) algorithms have been proposed in order to improve the goodput (defined as the application layer throughput) and latency of TCP over different types of networks [2].

However, the next generation of cellular networks will present new challenges for TCP[1], specifically related to mmWave links in the radio access network, which exhibit an erratic propagation behavior. This technology is seen as a promising enabler for the 5G targets of multi-gigabit/s data rates and ultra-low latency [3], but the end-to-end performance perceived by the user will eventually depend on the interaction with transport protocols such as TCP. Some recent studies [4], [5] have highlighted that the extreme variability of the signal quality over mmWave links yields either a degraded TCP goodput and a very low utilization of the resources at mmWave frequencies, or, in the presence of link-layer retransmissions, high goodput at the price of high latency. Moreover, in [4] it is shown that the bufferbloat phenomenon (i.e., the increase in latency that is caused by excessive buffering) emerges as a consequence of the presence of large buffers in the network.

Our goal is to answer the question: *Will TCP work in mmWave 5G cellular networks?* To this aim, we compare the performance of different TCP congestion control algorithms over simulated 5G end-to-end mmWave networks considering (1) a high speed train and (2) an urban macro 3GPP deployment, as further described in Sec. II. Our detailed simulation study demonstrates that the performance of TCP over mmWave depends critically on several aspects of the network:

1) **Edge vs. Remote Server:** By comparing the end-to-end performance at varying server's location, we show that for a shorter control loop, i.e., when the server is placed at the cellular network edge, TCP can react faster to link impairments.
2) **Handover and Multi-Connectivity:** Due to unreliability of individual mmWave links, dense deployments of small cells with fast handover protocols are critical in maintaining stable connections and avoiding TCP timeouts.
3) **CC Algorithms:** With remote servers, we observe higher performance variations across different congestion control algorithms, while the difference is almost negligible with edge servers. Overall, BBR outperforms loss-based TCP in terms of both rate and latency.
4) **TCP Packet Size:** We quantitatively compare the benefits of transmitting larger TCP packets in Long-Term Evolution (LTE) versus mmWave networks, and show that, given the fluctuating Gbps data rates offered at mmWave frequencies, a larger packet size provides a faster growth of the congestion window and higher achievable rate.
5) **Radio Link Control (RLC) Buffer Size:** We analyze TCP performance over small and large buffers. While the TCP goodput degradation caused by buffer overflow in undersized buffers is difficult to mitigate, the problem of bufferbloating, i.e., large buffer occupancy leading to delays, can be approached by appropriately designing cross-layer algorithms [6].

The rest of the article is organized as follows. We first describe the scenarios of interest in Sec. II. Then, we list the main features of the CC algorithms considered in this study in Sec. III. We report the main results and observations in Sec. IV, and draw our conclusions in Sec. V.

[1]In this work, we focus on TCP since it is the dominant transport protocol in use today. One possible avenue for future work is to consider the UDP protocol, that, however, shifts the burden of retransmissions and flow control to a higher layer, introducing similar problems as those related to TCP.

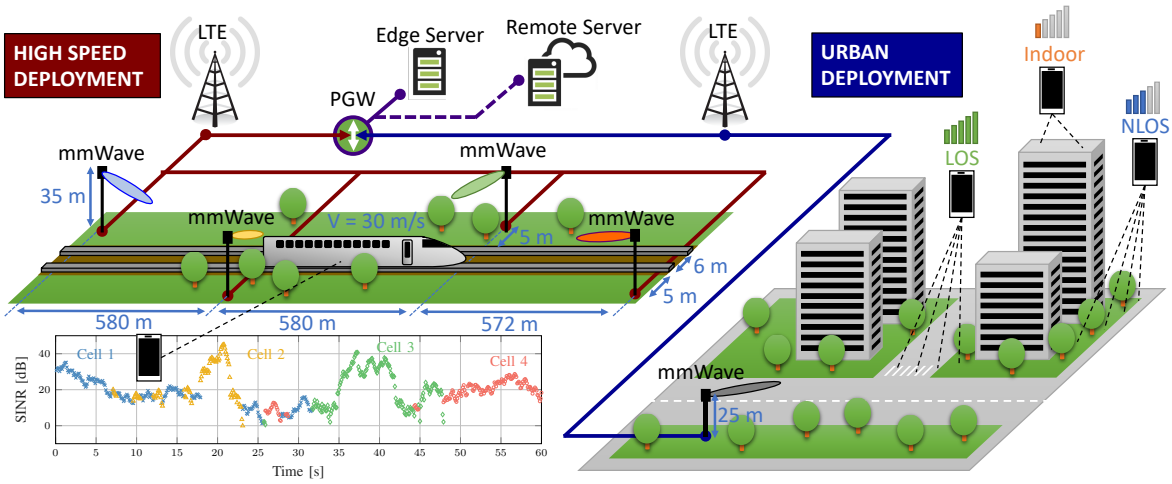


Figure 1: High speed and urban deployment scenarios

II. 5G Deployment Scenarios

In order to assess how TCP will perform in mmWave cellular networks, we consider two of the most challenging scenarios among those specified by the 3GPP in [7], i.e., a high speed train and a dense urban scenario, represented in Fig. 1. They were studied using the ns–3-based mmWave end-to-end simulation framework described in [8], which models radio access, the core network, and the 3GPP channel for the mmWave band with spatial correlation in mobility scenarios. Moreover, the protocol stack simulated by [8] also features retransmissions at both the MAC layer, with Hybrid Automatic Repeat reQuest (HARQ), and the RLC layer, using the acknowledged mode option.

High speed scenario: In this scenario, shown on the left side of Fig. 1, we test the performance of TCP over a channel that varies frequently in time and under realistic mobility conditions. Multiple Next Generation Node Bases (gNBs) provide coverage to the railway, which is mostly Line of Sight (LOS): even if the current gNB is blocked by obstacles placed between gNBs 2 and 3, the User Equipment (UE) can quickly perform a handover to another LOS gNB. The gNBs are at a height of 35 meters, with an intersite distance of 580 meters. The train moves at a speed of 108 km/h, and, as a result, the channel experienced by the UE varies very quickly because of severe fading and the Doppler effect, and, on a longer time scale, due to obstacles, as shown in the Signal to Interference plus Noise Ratio (SINR) plot of Fig. 1. We use the channel tracking and mobility scheme described in [9], which features fast and locally coordinated handovers for devices that are dual-connected to a mmWave gNB and a sub-6 GHz gNB (e.g., an LTE base station).

Dense urban scenario: In this deployment, shown on the right side of Fig. 1, we study the fairness of TCP flows over multiple UEs with different channel conditions. A single mmWave gNB placed at a height of 25 meters serves a group of ten users moving at walking speed. They are located in different positions, in order to account for a mixture of channel conditions: four UEs are in LOS, thus perceiving a very high SINR, four are in Non-Line of Sight (NLOS) and the last two are inside a building, so that the received power is additionally attenuated by the building penetration loss.

For both scenarios we consider two deployments of the TCP server which acts as the endpoint of the connection. The first is a traditional setup in which the server is hosted in a remote data center, with a minimum Round Trip Time (RTT) in the order of 40 ms, accounting for the latencies of both the core network and the public internet. The second is a Mobile Edge Cloud (MEC) scenario [10], in which the server is located close to the gNBs with smaller latency (of the order of 4 ms).

III. TCP Congestion Control Protocols

In this section, we will describe the congestion control protocols and the TCP performance enhancement techniques considered in this paper.

A. TCP Congestion Control Algorithms

We study four most commonly used CC algorithms.

TCP NewReno has been the default algorithm for the majority of communication systems. In the congestion avoidance phase, the congestion window `cwnd` is updated after the reception of every ACK. The update is based on the Additive Increase Multiplicative Decrease (AIMD) design: `cwnd` is increased by summing a term $\alpha/\texttt{cwnd}$ for each received ACK, and divided by a factor β for each packet loss. For NewReno these parameters are fixed to $\alpha = 1$ and $\beta = 2$.

HighSpeed TCP is designed for high Bandwidth-Delay Product (BDP) networks, in which NewReno may exhibit a very slow growth of the congestion window. HighSpeed behaves the same as NewReno when the congestion window is small, but when it exceeds a predefined threshold the parameters α, β become functions of the congestion window, in order to maintain a large `cwnd`. Moreover, the window growth of NewReno and HighSpeed depends on the ACK reception rate, thus a shorter RTT increases the ACK frequency and further speeds up the window growth.

TCP CUBIC, instead, increases the congestion window over time, without considering the ACK reception rate but

rather capturing the absolute time since the last packet loss and using a cubical increase function for `cwnd`. It has been designed to increase the ramp-up speed of each connection while maintaining fairness with other users.

TCP BBR, recently presented by Google [11], measures bottleneck bandwidth and round-trip propagation time, or BBR, to perform congestion control. It strives to match the sending rate to the estimated bottleneck bandwidth by pacing packets and setting the congestion window to `cwnd` gain $\times$ BDP, where the `cwnd` gain is a factor ($\leqslant 2$) that is used to balance the effects of delayed, stretched and aggregated ACKs on bandwidth estimation.

B. TCP Performance Enhancement Techniques

The performance of TCP has been the object of many studies over the last decades, and, besides new CC algorithms, many other techniques have been proposed and deployed either at the endpoints of the connection (TCP sender and receiver) or inside the network.

In case of multiple packet losses, the TCP Selective Acknowledgment (SACK) option [12] allows the receiver to inform the sender which packets were received successfully, so that the sender can retransmit only those which were actually lost. This dramatically improves the efficiency of the TCP retransmission process.

Active Queue Management (AQM) schemes [13], instead, are deployed in network devices (e.g., routers, gateways, gNBs), to control the behavior of their queues and buffers. The size of these buffers plays an important role in the end-to-end performance. If the buffer is too small, many packets may be dropped when the buffer is full, according to the drop tail policy. Conversely, if the buffer is too large, then the bufferbloat phenomenon occurs [13]. AQM techniques can be deployed at the buffers to drop packets before the queue is full, so that the TCP sender can proactively react to the congestion that could arise in the near future.

Finally, there are some techniques that are typically used in combination with wireless links. The first is the usage of link-layer retransmissions between the gNB and the UE, so that the losses on the channel are masked from TCP. This helps increase the goodput, however the end-to-end latency also increases, as shown in [5]. Another technique which is often used in wireless networks is proxying [2], i.e., the connection is split into two at some level in the mobile network (e.g., at the gateway with the internet, at the gNB, etc), and different CC techniques are deployed over the two parts of the connection.

IV. TCP Performance in the 3GPP Scenarios

In the following paragraphs we will report the performance of the TCP congestion control algorithms presented in Sec. III over the 5G mmWave deployment scenarios described in Sec. II, focusing on both goodput and latency. The results are averaged over multiple independent simulation runs, so that the confidence intervals are small (they are however not shown to make the figures easier to read). In all the simulations, we use full buffer traffic with the TCP SACK option and disable the TCP delayed ACK mechanism, thus each received packet will generate an ACK. The minimum retransmission timeout is set to 200 ms.

A. High Speed Deployment Scenario

In this scenario we compare different combinations of the RLC buffer size B and the Maximum Segment Size (MSS) P with a single TCP connection from the UE. For both the remote and the edge server deployments the RLC buffer is 10% or 100% of the BDP computed considering the maximum achievable data rate (3 Gbit/s) and the minimum latency, i.e., B equals 1.5 or 15 MB for the remote server deployment, and 0.15 or 1.5 MB for the edge server. We also consider two different MSS, i.e., a standard MSS of 1400 bytes (1.4 KB) and a large MSS of 14000 bytes (14 KB). The goodput of saturated UDP traffic is also provided as a reference for the maximum achievable rate, as shown in Fig. 2.

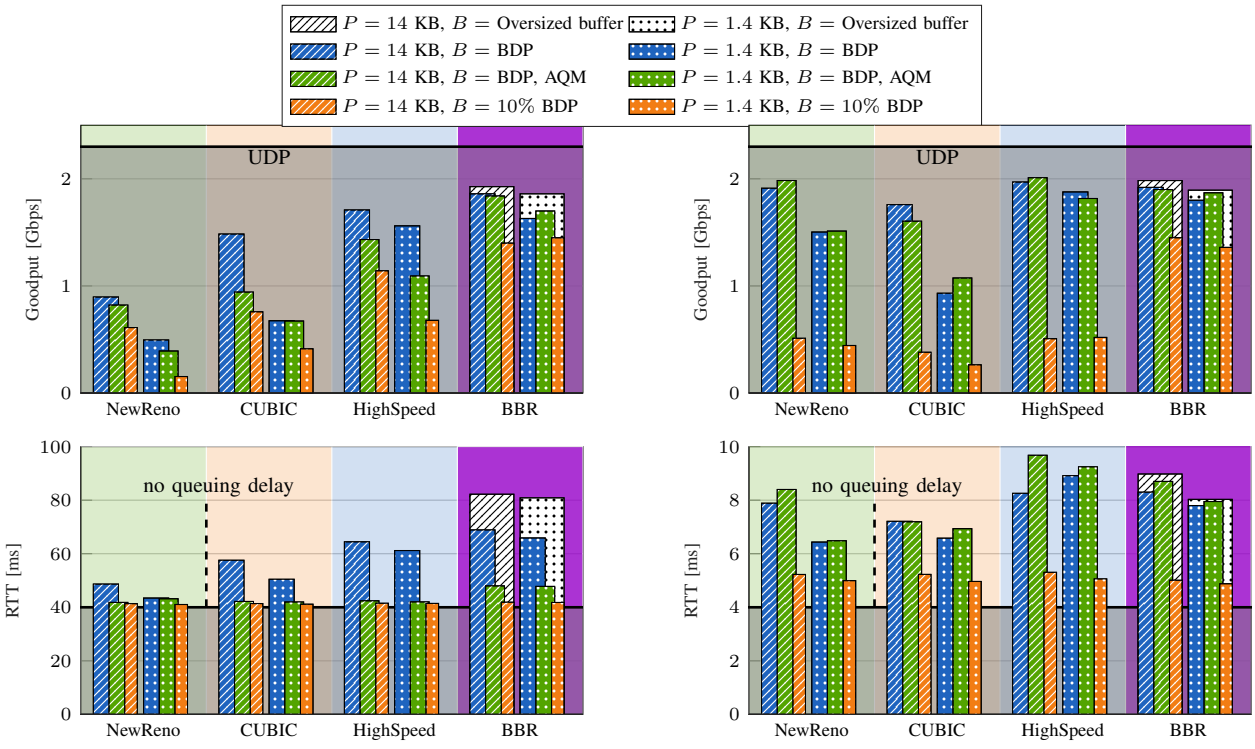
Notice that, thanks to the mobility management scheme based on dual connectivity and fast secondary cell handover, and despite the high mobility of the scenario, we never observed a TCP connection reset due to an outage, i.e., even if the closest two base stations are blocked, the UE is still capable of receiving signals from other nearby gNBs. Therefore, even if blockage events are still possible, in a scenario with a dense deployment (according to 3GPP guidelines), it is possible to provide uninterrupted connectivity to the final user [14].

In the following paragraphs we will provide insights on the effects of the different parameters on TCP performance over mmWave at high speed.

*1) **Impact of the server deployment**:* Loss-based TCP benefits from the shorter control loop related to an edge server deployment, as shown by comparing Figs. 2a and 2b. With the latter, indeed, the differences between the maximum goodput of the loss-based TCP versions are less marked, since the faster reaction time makes up for the differences among them. Moreover, the RTT difference between the large and the small RLC buffer is lower in absolute terms (milliseconds with edge server versus tens of milliseconds with remote server), but the ratio is approximately the same. However, for CUBIC and HighSpeed with the smallest buffer configuration, the goodput is lower with the edge than with the remote server, i.e., there is a 30% loss with the smallest MSS, and of 50% with the largest one. In this case, indeed, the buffer size is very small (i.e., $B = 0.15$ MB), thus incurring buffer overflows[2], which reduce the sending rate.

*2) **Impact of the congestion control algorithm**:* The congestion control algorithm has a stronger impact in the remote server scenario. The best performance, in terms of goodput, is achieved by BBR with large buffer size, but it is still 400 Mbps lower than the maximum achievable rate. Moreover, as observed in [4], [5], high goodput values also correspond to higher end-to-end latency. However, with small buffers, BBR produces the highest goodput (especially in the edge server scenario), with a latency comparable to loss-based TCP. BBR, indeed, regulates its sending rate to the estimated bandwidth

[2]With large MSS just 11 packets are enough to cause a buffer overflow.



(a) Remote server.

(b) Edge server.

Figure 2: Goodput and RTT for the high speed train scenario, with the remote and the edge server for different combinations of the buffer size and the MSS.

and is not affected by packet loss, i.e., the congestion window dynamics of BBR, presented in Fig. 3a, matches the SINR plot in Fig. 1.

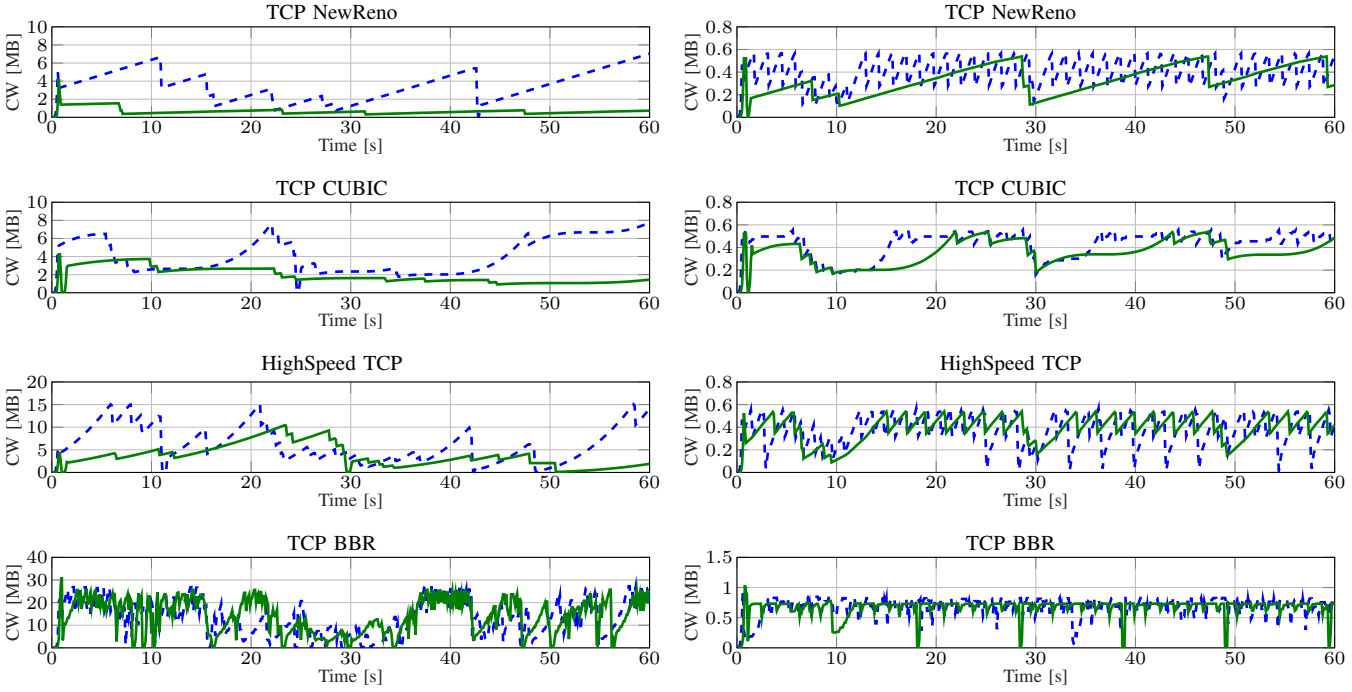
However, the loss-based versions of TCP cannot adjust their congestion window fast enough to adapt to the channel variations and perform worse than BBR, especially with small buffer, as seen in Fig. 3a. Among them, TCP HighSpeed provides the highest goodput because of the aggressive window growth in the high BDP region. TCP CUBIC performs better than NewReno in the remote server case, but worse in the edge server case. This is because CUBIC's window growth is not affected by the ACK rate, and therefore is more reliable over long RTT links.

*3) **Impact of the MSS**:* The MSS does not affect the performance of BBR, which probes the bandwidth with a different mechanism, whereas, for loss-based TCP, the impact of the MSS on the goodput is remarkable.[3] The standard MSS of P =1.4 KB exhibits much worse performance compared to a larger MSS of P =14 KB. This happens because, in congestion avoidance, the congestion window increases by MSS bytes every RTT, if all the packets are received correctly and delayed acknowledgment is not used, so the smaller the MSS the slower the window growth. Hence, the MSS dictates the congestion window's growth, which is particularly critical in mmWave networks for two main reasons: *(i)* The mmWave peak capacity is at least one order of magnitude higher than in LTE, so that the congestion window will take a much longer time to reach the achievable link rate. In this case, we can gain in performance by simply using a larger MSS, as depicted in Fig. 2. *(ii)* In addition, the channel fluctuations in the mmWave band will result in frequent quality drops, thus often requiring the congestion window to quickly ramp up to the link capacity to avoid underutilizing the channel.

Large MSS – mmWave vs. LTE: Aimed at better illustrating why larger packets are particularly important in 5G mmWave networks, we also provide a performance comparison against LTE in the same scenario[4], and report in Table I and Fig. 3 detailed results focusing on the impact of the TCP MSS on the congestion window growth and, consequently, on the goodput of the system. Only a single user is placed in the high-speed train scenario, thus the drops in the congestion window are due to the worsening of the channel quality and not to contention with other flows. Fig. 3 shows that the loss-based TCP congestion window with a small MSS grows very slowly in congestion avoidance, and consequently loss-based TCP does not fully exploit the available bandwidth during the intervals in which the received signal has a very high SINR (i.e., at $t = 20$ s and $t = 40$ s, as shown in Fig. 1). The large MSS helps speed up the congestion window's growth, which translates into higher goodput. Conversely, the

[3]Typically, TCP segments are mapped to multiple MAC/PHY data units, which complicates the dependence between a larger value of the TCP MSS and the correspondingly higher packet error probability over the wireless link. This non-trivial relationship, which would deserve a study by itself, has been properly captured in our numerical results.

[4]For the LTE setup the small buffer represents 50% of the BDP (i.e., 0.08 and 0.2 MB for edge and remote server, respectively), because a 10% BDP buffer would be too small to protect from random fluctuations of the channel.



(a) TCP congestion window with mmWave

(b) TCP congestion window with LTE

Figure 3: Congestion window evolution over time for different CC algorithms. The scenario is configured with remote servers and small RLC buffers

goodput degradation associated with small packets is less relevant in LTE networks, given that the goodput is limited by the available bandwidth and not by the congestion window increase rate. These trends are reflected in Table I. Among all loss-based TCP versions, only HighSpeed increases its congestion window fast enough even when transferring small packets. As a consequence, the goodput gain obtained with large MSS values is much smaller.

Large packets introduce an additional benefit: due to (1) a reduced TCP/IP header overhead and (2) a reduced number of TCP ACKs, there will be more available downlink/uplink resources, resulting in higher goodput values.

This solution may not be practical in an end-to-end network in which the Maximum Transmission Unit (MTU) is not entirely in control of the mobile network provider and is typically dictated by the adoption of Ethernet links (i.e., an MTU of 1500 bytes). By contrast, in a MEC scenario, in which the whole network is deployed by a single operator, it is possible to support a large MSS thanks to Ethernet jumboframes [15].

*4) **Impact of the buffer size and AQM**:* The buffer size is also critical for the performance of TCP. As shown in Fig. 2, large buffers generally yield higher goodput, because the probability of buffer overflow is smaller, and they offer a more effective protection against rapid and temporary variations of the mmWave channel quality. However, when a large buffer is coupled with loss-based TCP, the latency inevitably increases. Conversely, smaller buffers provide lower latency at the price of lower goodput.

For loss-based TCP, an intermediate solution is provided by applying AQM to the largest buffer, especially in the remote server scenario. Controlled Delay Management (CoDel) is used as the default AQM in our simulation because of its simple configuration. It controls the buffering latency by dropping packets when the minimum queuing delay within an interval, starting from 100 ms, is larger than 5 ms. CoDel is picked as an example to show the trade-off between latency and goodput by using AQM. Our goal for this paper is not to select the best AQM scheme or optimize AQM, which in itself is a very interesting topic, and could be considered for future research. As shown in Fig. 2a, the goodput with the AQM option is larger than that with the smallest buffer, and in some cases (i.e., for the smallest packet size) is comparable to that of the BDP buffer without AQM, which in general yields the highest goodput. However, the latency is equivalent to the one associated with the small buffer, which is the lowest. In the edge server scenario the TCP control loop is short (the RTT is 4 ms) and the reaction to congestion is quick. Hence, its performance is indeed equivalent to having BDP buffers

		Remote Server		Edge Server	
	Buffer	BDP	10% BDP	BDP	10% BDP
TCP NewReno	LTE	1.06	1.17	0.80	0.65
	mmWave	1.81	3.96	1.27	1.15
TCP CUBIC	LTE	1.06	1.15	1.03	0.89
	mmWave	2.2	1.83	1.89	1.44
HighSpeed TCP	LTE	1.08	0.9	0.94	0.95
	mmWave	1.09	1.69	1.05	0.98
TCP BBR	LTE	1.00	0.96	1.02	0.82
	mmWave	1.14	0.97	1.06	1.06

Table I: Ratio between the goodput achieved with $P = 14$ KB and with $P = 1.4$ KB, for different configurations of the simulated scenario.

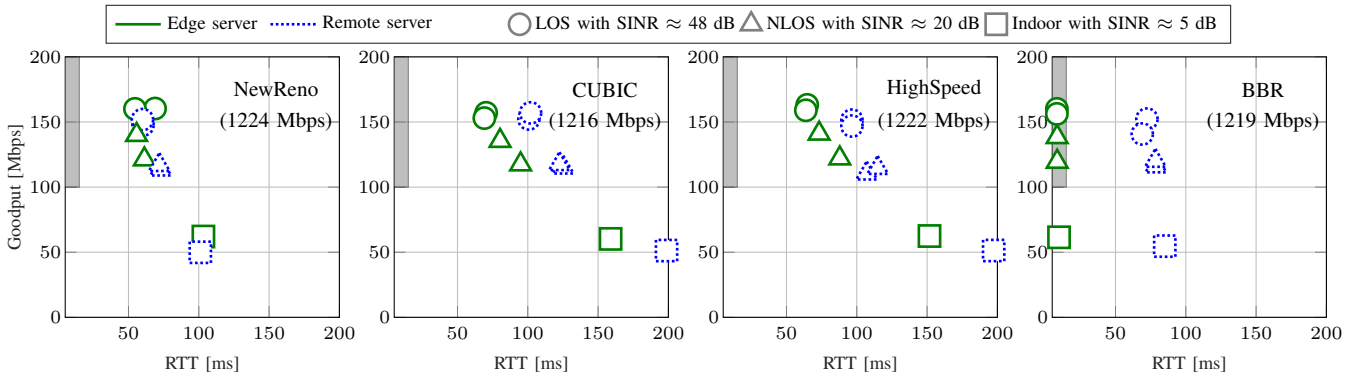


Figure 4: Goodput vs RTT for ten UEs in the Urban Scenario, for different choices of the CC algorithm.

without AQM.

BBR tries to solve this problem without modifying the buffers in the routers by maintaining a congestion window equal to twice the BDP regardless of packet loss, as shown by Fig. 3. As a consequence, latency is only doubled in large buffers, and the goodput is slightly reduced in small buffers. These behaviors are also observed in the oversized and 10% BDP buffer cases of Fig. 2.

B. Urban Deployment Scenario

In this scenario we consider ten UEs attached to a single mmWave gNB. In particular, we position four UEs in LOS conditions, four in NLOS and two inside a building. The average SINR for each channel condition is provided in Fig. 4. Notice that, with low blockage density and walking speed, the channel condition is relatively stable over time. For each UE pair one is connected to an edge server, and the other is connected to a remote server. In this way, it is possible to test the performance of TCP over a mixture of different conditions. The gNB uses a Round Robin (RR) scheduler, so that the resource management at the base station does not have an impact on the fairness among different flows. All the UEs use the same TCP version. We consider a standard MSS of 1400 bytes and an RLC buffer size of 1.5 MB for each UE.

Fig. 4 shows the average cell goodput (labeled in parentheses) and the goodput-latency trade-off for each type of user, separately, and for each CC algorithm, in order to evaluate the fairness and the overall performance of different TCP versions with respect to different user channel conditions.

All CC algorithms achieve the same average cell goodput, and similar goodput per UE. However, the RTT varies a lot among the CC algorithms. The reason is that all UEs use the same buffer size regardless of their channel conditions and network latency. As a consequence, the RLC buffer size may be large for some UEs, such as those at the edge. Therefore, the CC algorithms that adopt a more aggressive window growth policy, such as CUBIC and HighSpeed, yield much higher latency. For the loss-based TCP, NLOS and indoor UEs suffer from a higher latency: given the same buffer size and backhaul data rate, a reduced bottleneck bandwidth results into an increased queueing delay in the buffers, until TCP settles to a steady state phase. BBR, instead, limits the congestion window to twice the estimated BDP, and results in a maximum latency of 2 $\times$ minimum RTT. We also draw a gray area in the plot representing a typical 5G application requirement, i.e., goodput greater than 100 Mbps and delay lower than 10 ms. In this scenario, among all CC algorithms, only BBR meets this requirement for the UEs connected to an edge server, and only under good channel conditions.

V. Conclusions

The massive but intermittent capacity available at mmWave frequencies introduces new challenges for all layers of the protocol stack, including TCP, the most widely used transport protocol. The interplay between congestion control algorithms and mmWave channel quality fluctuations makes the topic particularly complex, and represents the key driver behind this work. We have carried out a thorough simulation campaign, based on ns-3, across 3GPP-inspired scenarios, whose results are summarized in Table II. The main findings and some relevant research questions are listed as follows: (1) TCP benefits from a shorter control loop, where the server is placed at the cellular network edge and can react faster to link impairments. *Should we (re)consider splitting TCP at some point?* (2) Moreover, when the RTT is high, loss-based TCP underutilizes the mmWave capacity, while those based on congestion (e.g., BBR) show an improved performance by estimating the bandwidth of mmWave links. This means that new approaches based on *more refined abstractions of the end-to-end network* can be studied for highly-variable and high-data-rate mmWave links. (3) Multi-connectivity and smart handovers, supported by advanced beamtracking and beamswitching techniques, will result in robust TCP connections. *How densely should we deploy mmWave cells? How to support backhaul for densely deployed mmWave cells?* (4) We show very clearly how loss-based TCP over mmWave bands can greatly benefit from using larger datagrams. *Has the time come to break the legacy MTU value*, by natively supporting larger packets in a wider set of networks? (5) Finally, it is well known that buffer size must scale proportionally to BDP to achieve maximum TCP goodput. However, it is very challenging to properly dimension the buffers for mmWave links, given the rapid bandwidth variations between LOS and NLOS conditions, and to protect from link losses without introducing bufferbloat. Given the low latency requirement

	Loss-based	MSS impacts goodput	Summary	Considerations over 5G
TCP NewReno	yes	yes	remote server: lowest goodput	need to move servers to the edge
TCP CUBIC	yes	yes	edge server: lowest goodput	need to increase MSS
HighSpeed TCP	yes	only remote server	big buffer: high goodput and high latency	need to mitigate latency with AQM
TCP BBR	no	no	big buffer: high goodput and high latency *small buffer: small rate reduction and low latency*	small buffer is preferred performs well over mixed UE conditions

Table II: Results of the CC algorithms over 5G deployments

and massive available bandwidth, *is it beneficial to trade bandwidth for lower latency,* for example by running BBR over small RLC buffer configurations?

We believe that these insights will stimulate further exploration of this important topic, which is essential to fully exploit the true potential of mmWave communications. Moreover, the observations provided by this initial simulation-based study can be used to guide the design of experimental activities, which are necessary to further validate the challenges that mmWave links pose to TCP, and to test novel techniques to improve the end-to-end user experience in mmWave cellular networks.

REFERENCES

[1] J. Postel, "Transmission control protocol," RFC 793, Sep. 1981.

[2] K. Liu and J. Y. Lee, "On Improving TCP Performance over Mobile Data Networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 10, pp. 2522–2536, Oct. 2016.

[3] M. Xiao, S. Mumtaz, Y. Huang, L. Dai, Y. Li, M. Matthaiou, G. K. Karagiannidis, E. Bjrnson, K. Yang, C. L. I, and A. Ghosh, "Millimeter Wave Communications for Future Mobile Networks," *IEEE J. on Sel. Areas Commun.*, vol. 35, no. 9, pp. 1909–1935, Sept 2017.

[4] M. Zhang, M. Mezzavilla, R. Ford, S. Rangan, S. Panwar, E. Mellios, D. Kong, A. Nix, and M. Zorzi, "Transport layer performance in 5G mmWave cellular," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, April 2016, pp. 730–735.

[5] M. Polese, R. Jana, and M. Zorzi, "TCP and MP-TCP in 5G mmWave Networks," *IEEE Internet Computing*, vol. 21, no. 5, pp. 12–19, Sept 2017.

[6] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan, and S. Panwar, "TCP dynamics over mmwave links," in *IEEE 18th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, July 2017.

[7] 3GPP, "TR 38.913, Study on Scenarios and Requirements for Next Generation Access Technologies, V14.1.0," 2017.

[8] 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, third quarter 2018.

[9] 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, Sept 2017.

[10] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1628–1656, third quarter 2017.

[11] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Queue*, vol. 14, no. 5, p. 50, 2016.

[12] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options," Internet Requests for Comments, RFC 2018, Oct. 1996.

[13] Y. Gong, D. Rossi, C. Testa, S. Valenti, and M. D. Täht, "Fighting the bufferbloat: on the coexistence of AQM and low priority congestion control," *Computer Networks*, vol. 65, pp. 255–267, June 2014.

[14] M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Mobility Management for TCP in mmWave Networks," in *Proceedings of the 1st ACM Workshop on Millimeter-Wave Networks and Sensing Systems 2017*, ser. mmNets '17, 2017, pp. 11–16.

[15] M. Mezzavilla, D. Chiarotto, D. Corujo, M. Wetterwald, and M. Zorzi, "Evaluation of Jumboframes feasibility in LTE access networks," in *IEEE International Conference on Communications (ICC)*, June 2013, pp. 5964–5968.

Menglei Zhang received the B.S. degree in electrical engineering from Nanjing University of Science and Technolodge, Nanjing, China, in 2013, and the M.S. degree in electrical engineering in 2015 from New York University Tandon School of Engineering, Brooklyn, NY, USA, where he is currently working toward the Ph.D. degree in electrical engineering with Prof. Rangan. His research interests include wireless communications, channel modeling, congestion control, and system level simulation.

Michele Polese [S'17] received his B.Sc. (2014) and M.Sc. (2016) in Telecommunication Engineering from the University of Padova, Italy. Since October 2016 he has been a Ph.D. student at the Department of Information Engineering of the University of Padova. He visited New York University (NYU) and AT&T Labs in Bedminster, NJ. His research interests focus on the analysis and development of protocols and architectures for 5G mmWave networks.

Marco Mezzavilla is a research scientist at the NYU Tandon School of Engineering. He received his B.Sc. (2007), M.Sc. (2010) in telecommunications engineering, and Ph.D. (2013) in information engineering from the University of Padova, Italy. He is serving as reviewer for IEEE conferences, journals, and magazines. His research interests include design and validation of communication protocols and applications of 4G/5G wireless technologies, multimedia traffic optimization, radio resource management, spectrum sharing, convex optimization, cognitive networks, and experimental analysis.

Jing Zhu [M'04-SM'12] received B.S. and M.S. degrees from Tsinghua University, China in 1999 and 2001 respectively, and a Ph.D. in 2004 from University of Washington, Seattle, all in electrical engineering. He is currently a Principal Engineer at Intel Corporation. His main research interests are system design, performance optimization, and applications for heterogeneous wireless networks, including 4G/5G cellular systems, high-density wireless LANs, and mobile ad hoc networks.

Sundeep Rangan [F'15] is an associate professor of electrical and computer engineering at NYU and Associate Director of NYU WIRELESS. He received his Ph.D. from the University of California, Berkeley in electrical engineering. In 2000, he co-founded (with four others) Flarion Technologies, a spinoff of Bell Labs that developed Flash OFDM, the first cellular OFDM data system. It was acquired by Qualcomm in 2006, where he was a director of engineering prior to joining NYU in 2010.

Shivendra Panwar [S'82-M'85-SM'00-F'11] is a Professor of Electrical and Computer Engineering at New York University. He is the Director of the New York State Center for Advanced Technology in Telecommunications (CATT), a member of NYU WIRELESS, and the Faculty Director of the NY City Media Lab. He was co-awarded the IEEE Communication Societys Leonard G. Abraham Prize, and a co-author of the IEEE Multimedia Communications and ICC Best Papers for 2011 and 2016, respectively.

Michele Zorzi [F'07] is with the Information Engineering Department of the University of Padova. His present research interests focus on various aspects of wireless communications. He was Editor-in-Chief of IEEE Wireless Communications from 2003 to 2005, IEEE Transactions on Communications from 2008 to 2011, and, at present, IEEE Transactions on Cognitive Communications and Networking. He served as a Member-at-Large of the ComSoc Board of Governors from 2009 to 2011, and as Director of Education and Training from 2014 to 2015.