Analysis of TCP Performance in 5G mm-wave Mobile Networks

Pablo Jiménez Mateo, Claudio Fiandrino, and Joerg Widmer
IMDEA Networks Institute, Madrid, Spain
E-mail: {firstname.lastname}@imdea.org

***Abstract*—Millimeter-wave (mm-wave) bands will play an essential role in 5G mobile networks in supporting the increasing demand for higher data rates. Communications at mm-wave frequencies pose unique challenges. The high propagation loss and unfavorable atmospheric absorption make the channel quality highly variable – short communication ranges and blockage through obstacles may prevent communication altogether. The use of directional antennas helps to achieve higher communication ranges and provides better spatial reuse and lower interference compared to omni-directional communications. At the same time, this introduces the problem of beam misalignment. Mm-wave research has primarily focused on the PHY and MAC layers, whereas the transport layer aspects of mm-wave systems require further attention. In this article, we analyze the behavior of TCP in mm-wave networks and study its impact on system-level performance. Through extensive simulations, we show the effect of different types of blockages on the behavior of the congestion control in the presence of handovers, and when small, medium and long flows coexist. Protocols like CUBIC that target high throughput benefit significantly when jointly optimizing link layers buffers and timeouts. While the optimization fosters prompt reaction to short-term blockages, the performance of such protocols significantly decreases when obstacles degrade the channel quality for longer time periods. Hybrid-designs like TCP YeAH are more robust to blockage, but fail to recover quickly and to ramp up to the link capacity after timeouts.**

I. Introduction

Millimeter-wave (mm-wave) communications systems have emerged as a key wireless technology in fifth-generation (5G) networks and beyond [1]. The limited spectrum available below 6 GHz makes it difficult for current systems to achieve the data rates required for 5G. Mm-wave bands use frequencies above 10 GHz and due to the massive available spectrum, they can provide orders of magnitude higher data rates than systems operating at lower frequencies. However, higher propagation loss and unfavorable atmospheric absorption make mm-wave systems highly susceptible to blocking. The use of highly directional antennas partially overcome such issues, but complicates link establishment and maintenance especially under user mobility. Consequently, changes in position and relative orientation between the device and Access Points (APs), and presence of obstacles and reflections lead to high variability in the channel quality. Users can either connect with single- or multi-connectivity modes, depending on the number of base stations they are simultaneously attached to [2]. From the network perspective, the former case is the most challenging scenario as a fallback to lower frequencies is not possible.

The unique dynamics of mm-wave communications, i.e., higher achievable data rates with high variability in channel quality, present particular challenges to the design of the physical (PHY) and medium access control (MAC) layers. To this date, the majority of the research has been devoted to investigate propagation issues [3], [4], beamforming procedures [5], and MAC layer design aspects [6]. However, the extreme propagation conditions of mm-wave links also impact the transport layer and especially the congestion control mechanisms. The role of congestion control is to regulate the amount of injected traffic in the network according to its congestion state. However, in wireless communications, traditional congestion control protocols like TCP NewReno are unable to differentiate between losses attributed to congestion and those attributed to transmission errors caused by a decay in channel quality. This well known problem is vastly exacerbated in mm-wave networks because of the magnitude of rate variations that sudden transitions from Line of Sight (LoS) to Non-LoS (NLoS) cause. At the transport layer, the sender is not notified about such short-term changes and might not decrease immediately the congestion window. Instead, lower layers react immediately by lowering the Modulation and Coding Scheme (MCS) used for subsequent data transmissions to increase robustness against adverse channel conditions. Preliminary works in this area analyzed the system-level implications of mm-wave channels on transport protocols [7], and have focused on understanding the performance of Multipath-TCP and the impact of link level retransmissions [8]. Additionally, Azzino et al. [9] proposed a cross-layer optimization of the congestion window, that is set according to the bandwidth-delay product by considering the latency estimated without buffering delays.

Throughout this article, we provide a systematic study of end-to-end performance of TCP congestion protocols in mm-wave 5G mobile networks. TCP is the de facto transport protocol used by the majority of the applications such as HTTP, Skype, file transfer and email [10]. Our contribution is to identify critical design aspects of congestion control protocols and highlight transport layer performance trade-offs particular to mm-wave networks. Compared to previous research in [7]–[9], the objective of this work is provide a comprehensive analysis, taking into consideration several loss-, delay-based and hybrid congestion protocols and studying their performance for different applications and scenarios: i) a single user served by one AP (Section III-A), ii)

a single user performing handover between multiple APs (Section III-B), and iii) multiple users exploiting different applications (FTP, video streaming and instant messaging) served by multiple APs (Section III-C).

II. Congestion Control Protocols over mm-wave Links

Channel quality in cellular networks is unpredictable for several reasons, including device mobility and handovers, frame scheduling algorithms which create burstiness and the transitions of the Radio Resource Control (RRC) state machine [11]. Congestion control protocols probe the channel to exploit the available rate by injecting packets until losses occur, or proactively adapt the injection rate to match delay- or rate-based measurements. Legacy congestion control protocols were primarily designed for wired networks assuming fixed capacity links, and hence suffer in the presence of the short term link quality variations common in cellular networks. These issues are further exacerbated in high-speed mm-wave networks. Large and persistently full buffers are typical in cellular networks to help maintain a high throughput, but they lead to the bufferbloat problem, which significantly increases latency [12]. The following paragraphs revisit congestion control mechanisms, highlight challenges and identify critical issues specific to mm-wave networks.

A. Fundamentals of TCP

TCP ensures that all the data packets are correctly received and in order. To this end, TCP requires each packet to be acknowledged by the receiver (ACK). TCP relies on a Congestion Window (CW) which determines the maximum quantity of data that can be sent within one round trip time (RTT). The CW adaptation procedure depends on the link status. During TCP slow start, the CW increases by one packet per received ACK until: (i) the *sstresh* threshold is reached, or (ii) there is a packet loss. Upon reaching *sstresh*, TCP enters the congestion avoidance phase and the CW grows, for example for the NewReno protocol, by one packet whenever a whole congestion window worth of data has been acknowledged. When the receiver obtains a packet that has a sequence number higher than the expected one, a duplicate ACK is triggered. This informs the sender that the packet with the expected sequence number has been lost and requires retransmission. If three duplicate ACKs are received, TCP assumes a packet loss: it enters in fast retransmit mode by halving the CW and continues in congestion avoidance phase. If after a given period the Retransmission Time Out (RTO), commonly set to 1 s, no ACKs were received for a packet, TCP reduces the CW to one packet and recovers from the slow start phase.

B. Congestion Control Protocols

Congestion control protocols differ in the mechanisms utilized to adapt to the available bandwidth. According to the methodology employed to detect congestion, protocols can be attributed to three main categories: loss-, delay-based and hybrid. To obtain insightful results, a preliminary study narrowed down the initial set of protocols from twelve (BBR, CUBIC, HighSpeed, HTCP, Hybla, Illinois, New Reno, Scalable TCP, Vegas, Veno, Westwood, WestwoodPlus, and YeAH [13]) to seven. Preference to widely-known protocols and those implemented in current networks were the selection criteria. For example, Hybla was discarded being primarily used in high-latency terrestrial links or satellite links. The seven different protocols which encompass all the aforementioned categories loss-, delay-based and hybrid are: NewReno, Scalable TCP, CUBIC, Vegas, Westwood, YeAH and BBR. NewReno relies on packet losses as indication of congestion, halving the size of the CW whenever a loss is detected. Compared to the older Reno TCP, the Fast Recovery mechanism is designed to increase robustness to multiple losses in a single window. While NewReno follows a Additive Increase Multiplicative Decrease strategy, Scalable TCP implements a Multiplicative Increase Multiplicative Decrease strategy to growth the rate more quickly. The rate increase is proportional to the estimated spare bandwidth of the link. CUBIC increases the CW according to a cubic function by computing the absolute time since the last dropped packet. In the first part, the function is concave and CUBIC attempts to quickly reach the CW size registered before the last dropped packet. In the second part, the function is convex and CUBIC conservatively probes for additional bandwidth. Vegas implements congestion control on the basis of delay-measurements to proactively estimate the buffer status of the bottleneck router. The CW size increases or decreases with an Additive Increase Additive Decrease strategy by computing the difference between the actual rate and the ideal one that the TCP flow would achieve without congestion. Westwood adjusts parameters like *sstresh* and CW according to estimations of the available network bandwidth based on measurements of the average rate of received ACK packets. YeAH is a hybrid approach, which determines the CW on the basis of both losses and delay that makes it more robust to LoS-NLoS transitions. This is because YeAH employs two modes, the *fast* and the *slow* modes. The *fast* mode uses an aggressive rule in the spirit of Scalable TCP to quickly grow the CW, while the *slow* mode acts like NewReno, hence the state is determined by the estimated number of packets in the bottleneck queue. When YeAH estimates that packet buffering and queuing delays in the network are below or above predefined thresholds, it switches between *fast* and *slow* mode. The presence of NLoS enforces the *slow* mode in YeAH, which better copes with the low available bandwidth. Similarly to YeaH, TCP Bottleneck Bandwidth and RRT (BBR) also continuously measures RTT. Unlike the previous congestion control protocols, BBR is model-based. Specifically, BBR builds a model of the network by continuously measuring the available bandwidth at the bottleneck link and the two-way propagation delay. In this way, it estimates the bandwidth-delay product and accordingly adjusts the sending rate through pacing.

C. Effects of mm-wave Channel Properties on Upper Layers of the Protocol Stack

Under LoS conditions, a mm-wave link can offer several Gbps of throughput [6]. During the congestion avoidance phase, TCP tries to take advantage of all of the available bandwidth. However, if the channel becomes blocked by an obstacle, the data rate drastically drops. Sudden transitions between LoS and Non-LoS make the intermediate router buffers to fill very rapidly due to the high data rate, until the fast retransmit phase starts or a RTO is triggered. When entering the fast retransmit phase, the value of the CW is halved. However, this might not be sufficient to adapt to the new low link capacity and might in turn cause a RTO. TCP then resumes with slow start to correctly adjust to the available bandwidth. Upon a RTO, the slow start threshold is reduced, which causes TCP to take longer in achieving the optimal value for the CW and obtain high throughput, especially when the channel quality improves from NLoS to LoS.

Retransmissions at lower layers of the mobile network stack, i.e., the Radio Link Control (RLC) and the MAC layer, play an essential role in maintaining throughput as they hide channel losses to TCP [14]. The RLC in Acknowledged Mode (AM) splits the TCP Protocol Data Units (PDUs) into smaller chunks (RLC PDUs) that are acknowledged by the RLC layer at the sender side. The Automatic Repeat Request (ARQ) on the RLC layer is in charge of asking for the retransmission of corrupted RLC PDUs. Hence, augmenting the RLC buffer size is a solution to compensate for high number of losses. Although the maximum number of retransmission attempts per RLC PDU is limited, this operation increases RTTs and, in turn, might trigger TCP timeouts. To reduce the number of retransmissions on the RLC layer, a Hybrid ARQ (HARQ) mechanism at the MAC layer can recover corrupt packets by soft combining two or more corrupted RLC PDUs stored in its buffer. The minimum duration of the RTO timer typically employed in current wireless networks is large and not suitable for mm-wave systems. While in presence of short blockages entering in fast retransmit phase is beneficial, during extensive NLoS periods TCP is likely to trigger multiple RTOs. Thus, TCP can significantly benefit from the reduction of the waiting period before it recovers from the slow start phase.

III. TCP Performance Analysis

This section analyzes the performance of multiple congestion control protocols. For the simulations we use the mm-wave module of ns-3 [15], which implements the complete protocol stack and accurately represents the underlying mm-wave channel according to the 3GPP channel model.

A. Single Flow Analysis

A mm-wave AP working at 1 GHz bandwidth, 30 dBm maximum transmit power and a carrier frequency of 28 GHz serves one user equipment (UE) at a distance of 50 m. The UE remains static for 2 s, then moves following a straight path parallel to the AP location at a walking speed of 1.5 m/s for 21 s and finally it stops and remains static for another 2 s.

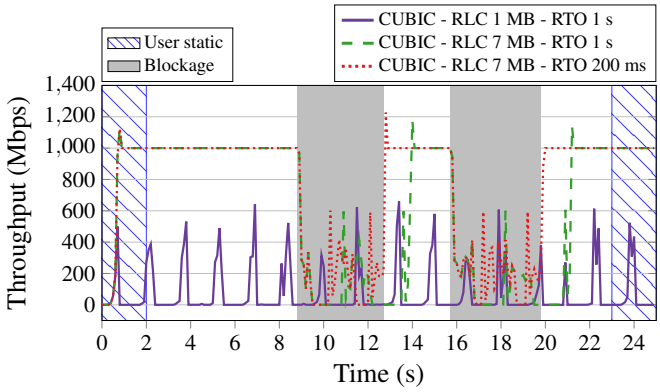


Fig. 1. Throughput comparison with different configurations of CUBIC

Along the path, the connection is interrupted. Three blockage scenarios are considered: i) a building creating an extensive blockage of 13 s, ii) two small buildings creating medium blockages of 4 s each, and iii) 6 small obstacles creating multiple short blockages of 0.25 s each. Scenario i) and ii) permit to analyze the performance of congestion protocols when the penetration loss remains high for a prolonged period of time. Finally, Scenario iii) extends the analysis of Scenario ii) augmenting the number of LoS-NLoS transitions and shortening their duration. In the presence of medium or extensive blockage, congestion control protocols can detect the link capacity reduction while in the presence of temporary short-term blockage this is not possible as the blockage ends too rapidly.

While in traditional mobile technologies like 3G/4G, RLC buffer sizes on the order of 1 MB were sufficient to compensate for wireless losses, in mm-wave networks the size needs to be adjusted for the higher data rates. A prior analysis through experimentation in different scenarios and for all TCP protocols showed that a 7 MB RLC buffer size achieves maximum throughput and limits the bufferbloat problem. Fig. 1 shows the results of this analysis. Note that the dimension is conservative with respect to the set-up of [9] and is in line with the 3GPP 36.306 TS Release 14 specification for a category 12 UE. Fig. 1 shows throughput of CUBIC comparing the joint RLC buffer size and RTO timer with modified setting, e.g., with an RLC buffer of 7 MB and and RTO value 200 ms, an RLC buffer of 7 MB and RTO of 1 s, and the default settings of an RLC buffer of 1 MB and RTO of 1 s. The limited RLC buffer size of the baseline setting causes CUBIC to suffer considerably from wireless losses and the protocol persistently fails to reach the congestion avoidance phase. Additionally, the duration of the RTO timer prevents CUBIC from reacting quickly. Hence, the achieved throughput in LoS phases is sub-optimal. The importance of the timer becomes evident when comparing RLC versus RLC+RTO with modified setting: in NLoS periods, the latter provides a significant advantage to recover faster from NLoS-LoS transitions.

Fig. 2 shows performance of the different protocols after having updated the setting of RLC+RTO as previously discussed for the considered scenarios. The graph is a throughput-RTT plot, where for each scenario we take the average results

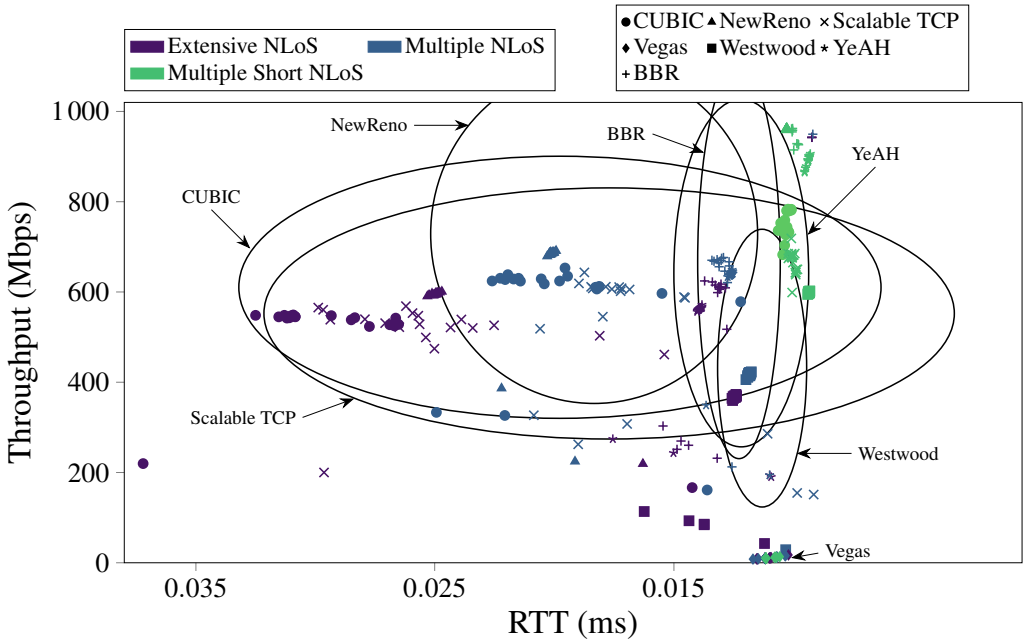


Fig. 2. Performance of TCP congestion protocols for different scenarios. The elliptic contour defines the confidence interval with a $2-\sigma$ precision, hence the outliers are those samples outside the 95% confidence interval region.

and compute the $2-\sigma$ elliptic contour of the maximum-likelihood 2D Gaussian distribution. The $2-\sigma$ expression defines the level of confidence, i.e., the projection on a one-dimensional sub-space of the elliptic contour is the 95% confidence interval. The plot shows the results for individual scenarios and the overall congestion control protocol performance with different markers. On the x-axis, lower (better) RTTs are to the right. Hence, best performing protocols are on the top-right. Throughput-RTT plots highlight the variability and relative performance between protocols. The more narrow the ellipses in the axis dimension, the more stable is the protocol in consistently achieving similar throughput or RTT performance. On the other hand, wider ellipses indicate higher variability. The ellipses' orientations define the relationship between throughput and RTT.

Both CUBIC and Scalable TCP aggressively increase their rate, with Scalable TCP implementing exponential increase/decrease dynamics, while the congestion window in CUBIC grows as a cubic function of the time elapsed since the last congestion event (see Section II-B). Specifically, CUBIC's congestion window growth is fast or slow when the current window is respectively far and close to the target one. This mechanism enables prompt recovery and is particularly suitable for short NLoS periods. In presence of a LoS to medium or extensive NLoS transition, both Scalable TCP and CUBIC at first attempt to recover to the sending rate achieved during LoS. However, due to the high channel quality degradation, both protocols repeatedly attempt recovery from slow start. As a result, their performance differs considerably from one scenario to another and the protocols are very susceptible to LoS-NLoS transitions in RTT metrics.

NewReno and Westwood exhibit less variability in RTT than CUBIC and Scalable TCP. Westwood, upon detection of congestion, computes the current bandwidth-delay product and sets the slow start threshold accordingly. NewReno does not terminate the fast recovery phase until complete recovery of multiple losses and thus take longer to achieve the full link capacity. In the congestion avoidance phase, both protocols gently probe for additional bandwidth.

YeAH congestion control is based on both packet loss and RTT measurements and its objective is to limit the amount of buffering in the network (see Section II). Hence, YeAH does not increases the rate further once determines that the sending rate is sufficiently high. As queuing delay computations are based on the minimum of the recently measured RTT and not the average RTT, in presence of blockage, YeAH quickly switches to slow mode which prevents recovery from slow start. BBR congestion protocol has a very similar performance to YeAH as it adapts its rate to minimize queuing over the whole network, but also achieves a lower RTT among all the scenarios. Finally, Vegas consistently achieves the lowest throughput and exhibits little variability in RTT. Indeed, the Vegas congestion control mechanism sets the data rate so that all transmitted packets can be acknowledged within the minimal RTT. Upon finding such a rate, Vegas does not attempt to increase the rate to use a higher fraction of the link capacity under-utilizing the high bandwidth at disposal in LoS significantly.

For performance evaluation, in the next sections CUBIC and YeaH are selected because of the different level of robustness to blockage. While Scalable TCP provides similar performance than CUBIC, the latter is the de-facto protocol implemented from Linux kernel v2.6.19.

B. Single Flow with Handover Analysis

Handovers transfer the ongoing data session from the current AP to another, to ensure that a UE remains connected while it is moving. While in traditional cellular networks handovers are mainly triggered by user movements out of the operational range of an AP, in mm-wave networks also blockage is a probable cause for handovers. There exist several methods to trigger a handover [2]. The threshold method estimates the Signal to Interference to Noise Ratio (SINR) degradation. The *fixed* and *dynamic* Time to Trigger (TTT) generate a handover event if the SINR is below a certain threshold during a predetermined or variable time window, respectively. The objective of the timers is to reduce the ping pong effect, i.e., a UE switching between two APs in a rapid succession. The following analysis employs a fixed TTT with 200 μs time window as per 3GPP TS 36.331 V8.4.0.

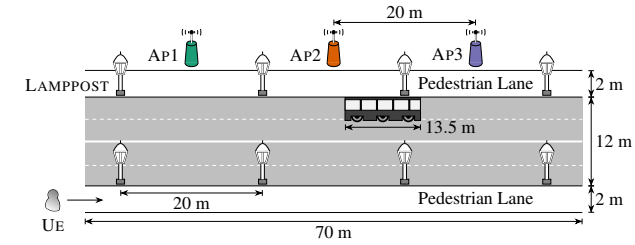
Fig. 3(a) shows the geometry of the scenario, which mimics a typical street. Lamp posts are distributed at regular intervals of 20 m and there are two double lanes for cars next to which are pedestrian lanes of 2 m each. Three APs, deployed at a regular distance of 20 m, ensure full coverage. The UE moves at a constant speed of 1.5 m/s on the bottom pedestrian lane from left to right. The lampposts create multiple short blockages similar to the multiple NLoS scenario analyzed in Section III-A. Additionally, a 13.5 m wide bus parked on the opposite lane of the user generates extensive blockage for AP2 and AP3.

Fig. 3(b) shows the SINR as the UE moves along the pedestrian lane. The colored backgrounds represent NLoS periods to the corresponding APs. The color of the SINR profile represents the AP to which the UE is currently attached. When an obstacle blocks the current AP, a handover event is automatically triggered. However, in the presence of short-term blockage, this decision is suboptimal as the connectivity rapidly moves from one AP to the other causing a throughput decrease (see Fig. 3(c)). The graph shows results for CUBIC and YeAH protocols that exhibit different levels of robustness towards blockage (see Section III-A). Unsurprisingly, during rapid handover events CUBIC drops its rate and recovers quickly from slow start. YeAH, instead, switches to slow mode and reaches the link capacity slowly.

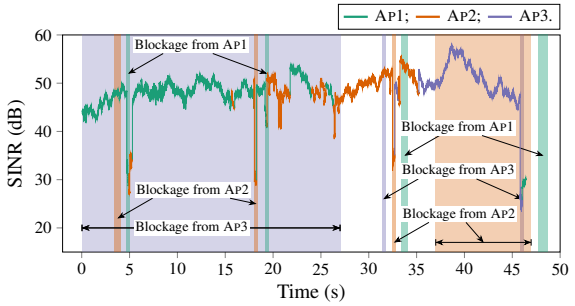
C. Multiple Concurrent Flows Analysis

Previous research showed that long flows in the background can significantly reduce the responsiveness of short flows [12]. This section specifically analyzes performance of TCP over mm-wave channel dynamics and rates when short flows and background traffic coexist.

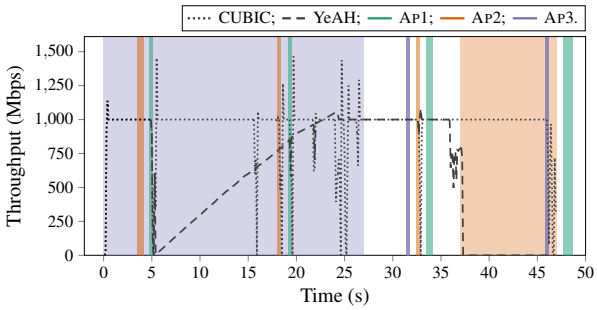
Fig. 4 shows a more complex scenario where 10 users with different applications are served by 4 APs in a square. The scenario represents a dense small cell deployment with a statue in the center of the square and lampposts placed all around the statue that create blockages. UE0 has a throughput of 500 Mbps mimicking a FTP download. This flow starts at the beginning of the simulation and lasts until the end. The rest of the users are either streaming a video (UE2, UE4



(a) Scenario



(b) SINR



(c) Throughput

Fig. 3. Handover analysis: scenario (a), SINR (b) and throughput (c)

and UE5) or using instant messaging applications. The video applications buffer data at 4 Mbps for 30 s and then stop their flow for 2 s before starting again. The rest of the users are using instant messaging applications or social networks, sending a few packages every few seconds. These applications are common for smartphones. Except for UE6−UE9, all UEs experience handovers while the user is walking, and eventually move the active connection to an AP that is already serving other flows. CUBIC and YeAH are the congestion control protocols used for the comparison.

Impact of Retransmissions: Fig. 5 compares the number of successfully received packets and retransmissions. UE0's application is the one generating the heaviest flow, hence the experiment allows to verify the performance of medium-small flows with heavy background traffic. The complexity of the scenario also allows to verify the effect of medium flows on small ones. As UE0 moves through the environment, its connection changes from one AP to another due to handovers. Consequently, the rates of the UEs that were already connected to this AP are degraded to accommodate the incoming long flow, or their connection is moved to other APs. The combined effect of blockage and presence of the long flow is different for medium and small flows and for the two congestion control protocols. With the sole exception of UE4, YeAH consistently

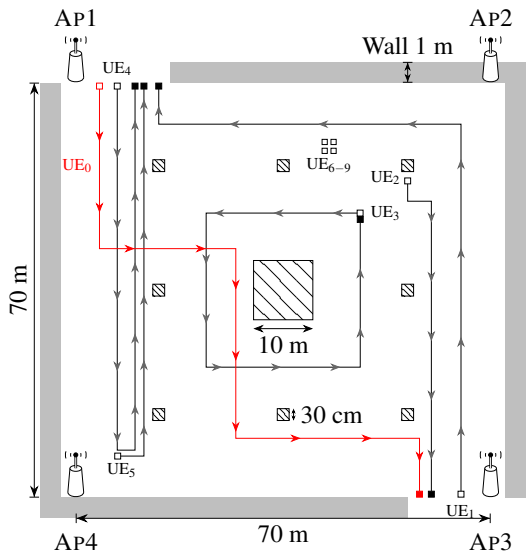


Fig. 4. System scenario

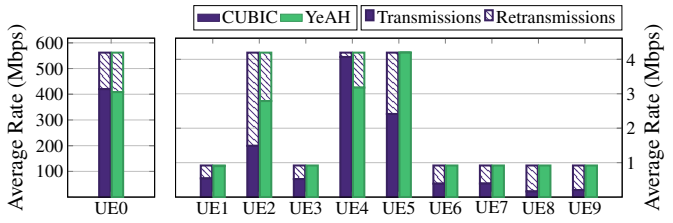


Fig. 5. Composition of successful transmissions and retransmissions with CUBIC and YeAH

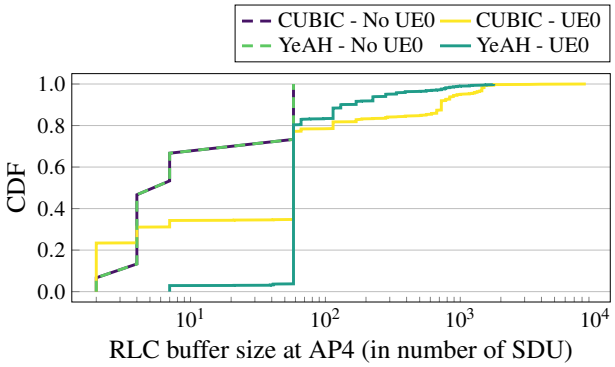


Fig. 6. CDF of AP4 RLC buffer occupancy for CUBIC and YeAH with and without UE0 traffic in background

achieves a lower number of retransmissions. Recalling the result in Fig. 2, by design YeAH reacts to congestion based on both packet loss and RTT measurements. Limiting the sending rate on this basis, it achieves higher robustness than CUBIC, whose performance is strictly dictated by the duration of blockage causing packet losses. Nonetheless, for UE0 YeAH achieves data rates up to 500 Mbps similar to CUBIC. For all the others UEs and for both congestion control protocols, the achieved data rates are lower because the amount of data to be transmitted is small.

UE2 has comparatively the highest number of retransmissions with both protocols. Indeed towards the end of its movement, it competes with UE0 and the network makes UE2 to frequently handover between AP3 and AP4. UE4, although following a similar path as UE0, is not severely affected by UE0 and is served by AP4 for the entire period. Hence, YeAH outperforms CUBIC because the retransmissions are solely due to errors. Unlike UE4, UE5 is served by multiple APs during the movement and, consistent with the results obtained in Subsection III-B, CUBIC is more affected than YeAH. The performance of static users UE6–UE9 is similar. They are scheduled concurrently with the medium-sized flow of UE1 and are partially affected from the long flow. YeAH provides consistent results and does not incur retransmissions, while CUBIC is more susceptible to blockages and requires more retransmissions. Specifically, CUBIC uses up to five times the RLC buffer space compared to YeAH. Unlike UE6–UE9, UE3's connectivity is persistently blocked by small obstacles, however its performance is comparable to that of static users in NLoS.

Analysis of RLC Buffer Occupancy: Fig. 6 analyzes the effect of the long flow on a medium-sized one. The plot shows the CDF of the contribution from UE4 to the RLC buffer occupancy of AP4 for both CUBIC and YeAH. Note that UE4 is constantly in LoS and served by AP4 during its movement. Interestingly, in absence of the long flow from UE0, both congestion control protocols lead to the same buffer occupancy. Hence, the hybrid ARQ mechanism at the MAC layer is sufficient to hide wireless errors from the RLC layer and the transport layer. In presence of the long flow in the background, this is no longer true and AP4 provides connectivity to both long and medium flows from UE0, UE4 and UE5 respectively. Consequently, the contribution of UE4 to the RLC buffer occupancy increases. Unsurprisingly, CUBIC and YeAH performance differs and CUBIC causes higher buffer occupancy on average, while achieving higher rates.

IV. Discussion and Open Challenges

Proper setting of RLC buffer and RTO timers is essential to alleviate the impact of losses on throughput and foster prompt recovery from slow start. RLC buffer size can not be arbitrarily augmented to prevent delay increase and bufferbloat. Given the short term variations of link quality, quick recovery from the slow start phase is crucial. This can be achieved by shortening the timeout value. Further investigation on optimizing buffer size and timers is an interesting direction for future research, especially to foster ultra-low latency and reliable communications. Another promising research direction involves cross-layer optimization of transport, MAC and PHY layers. For example, more advanced solutions for directional antennas and beamforming may potentially benefit from machine learning techniques to learn the properties of the environment and optimize resource allocation.

Congestion control protocols that aim at maximizing throughput through quick recovery like CUBIC suffer with longer NLoS periods while they perform comparatively well in the presence of short NLoS. On the other hand, the performance variations of protocols that detect congestion through hybrid mechanisms (packets loss and RTT measurements) like

YeAH is minimal. This tradeoff is especially evident in presence of handovers: while CUBIC resumes from slow start upon any handover event, YeAH probes bandwidth more gently and prevents short-term throughput variations. When long, medium and small flows coexists, CUBIC always requires more transmissions than YeAH to successfully complete the data transmission. Hence, YeAH is suitable for transmission of small flows over mm-wave channels. However, CUBIC outperforms YEAH in terms of achieved throughput.

V. Conclusions

This article studied the behavior of multiple congestion control protocols over mm-wave networks. In contrast to traditional mobile networks, mm-wave links have unique features, including high available bandwidth, high variability in channel quality and sensitivity to blockage because of high propagation and penetration loss and atmospheric absorption. Through extensive simulations, the paper has shown aspects of packet loss-, delay-based and hybrid congestion control protocols by analyzing the impact on throughput and latency of various environments with blockages. Furthermore, the paper has presented open challenges and outlined promising future research directions.

Acknowledgment

This work was supported by the ERC project SEARCHLIGHT, grant no. 617721, the Ramon y Cajal grant RYC-2012-10788, and the Madrid Regional Government through the TIGRE5-CM program (S2013/ICE-2919).

References

[1] T. S. Rappaport, S. Sun, R. Mayzus, H. Zhao, Y. Azar, K. Wang, G. N. Wong, J. K. Schulz, M. Samimi, and F. Gutierrez, "Millimeter wave mobile communications for 5G cellular: It will work!" *IEEE Access*, vol. 1, pp. 335–349, May 2013.

[2] M. Polese, M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Improved handover through dual connectivity in 5G mmwave mobile networks," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 9, pp. 2069–2084, Sep 2017.

[3] T. S. Rappaport, G. R. MacCartney, M. K. Samimi, and S. Sun, "Wideband millimeter-wave propagation measurements and channel models for future wireless communication system design," *IEEE Trans. on Communications*, vol. 63, no. 9, pp. 3029–3056, Sep 2015.

[4] M. V. Perić, D. B. Perić, B. M. Todorović, and M. V. Popović, "Dynamic rain attenuation model for millimeter wave network analysis," *IEEE Trans. on Wireless Communications*, vol. 16, no. 1, pp. 441–450, Jan 2017.

[5] D. D. Donno, J. P. Beltran, and J. Widmer, "Millimeter-wave beam training acceleration through low-complexity hybrid transceivers," *IEEE Trans. on Wireless Communications*, vol. 16, no. 6, pp. 3646–3660, Jun 2017.

[6] H. Shokri-Ghadikolaei, C. Fischione, G. Fodor, P. Popovski, and M. Zorzi, "Millimeter wave cellular networks: A MAC layer perspective," *IEEE Trans. on Communications*, vol. 63, no. 10, pp. 3437–3458, Oct 2015.

[7] 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 *Proc. IEEE INFOCOM WKSHPS*, Apr 2016, pp. 730–735.

[8] 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, Sep 2017.

[9] T. Azzino, M. Drago, M. Polese, A. Zanella, and M. Zorzi, "X-TCP: a cross layer approach for TCP uplink flows in mmwave networks," in *Proc. Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, Jun 2017, pp. 1–6.

[10] J. F. Kurose and K. W. Ross, "Computer networking: a top-down approach," *Addison Wesley Computing*, 2013.

[11] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, "Adaptive congestion control for unpredictable cellular networks," in *Proc. ACM SIGCOMM*, 2015, pp. 509–522.

[12] S. Alfredsson, G. D. Giudice, J. Garcia, A. Brunstrom, L. D. Cicco, and S. Mascolo, "Impact of TCP congestion control on bufferbloat in cellular networks," in *Proc. IEEE WoWMoM*, Jun 2013, pp. 1–7.

[13] A. Afanasyev, N. Tilley, P. Reiher, and L. Kleinrock, "Host-to-host congestion control for TCP," *IEEE Communications Surveys Tutorials*, vol. 12, no. 3, pp. 304–342, Third Quarter 2010.

[14] R. Kumar, A. Francini, S. Panwar, and S. Sharma, "Dynamic control of RLC buffer size for latency minimization in mobile RAN," in *Proc. IEEE WCNC*, Apr 2018.

[15] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-end simulation of 5g mmwave networks," *IEEE Communications Surveys Tutorials*, vol. 20, no. 3, pp. 2237–2263, Third Quarter 2018.