mmCPTP: A Cross-Layer Pull based Transport Protocol for 5G mmWave Networks

Prasad Netalkar, Jiachen Chen and Dipankar Raychaudhuri
WINLAB, Rutgers University, NJ, USA
{pnetalka, jiachen, ray}@winlab.rutgers.edu

NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium | 978-1-6654-7716-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/NOMS56928.2023.10154364

***Abstract*—This paper presents mmCPTP, a cross-layer end-to-end protocol for fast delivery of data over mmWave channels associated with emerging 5G services. Recent measurement studies of mmWave channels in urban micro cellular deployments show considerable fluctuation in received signal strength along with intermittent outages resulting from user mobility. This results in significant impairment of end-to-end data transfer throughput when regular TCP is used to transport data over such mmWave channels. To address this issue, we propose mmCPTP, a novel cross-layer end-to-end data transfer protocol that sets up a transport plug-in at or near the base station and uses feedback from the lower layer (RLC/MAC) to opportunistically *pull* data at the mobile client without the slow start and probing delays associated with TCP. The system model and end-to-end protocol architecture are described and compared with TCP and Indirect-TCP (I-TCP) in terms of achievable data rate. The proposed mmCPTP protocol is evaluated using NS3 simulation for 5G NR (New Radio) considering a high-speed mobile user scenario. The system is further validated using a proof-of-concept prototype which emulates the high-speed mmWave/NR access link with traffic shaping over Gbps ethernet. Results show significant performance gains for mmCPTP over TCP and I-TCP (2.5x to 17.2x, depending on the version).**

***Index Terms*—wireless access, 5G, mobile data, mmWave, TCP, transport proxy, pull protocol**

I. Introduction

The recent emergence of 5G services [1] raises the prospect of data transfer to mobile wireless devices at ultra-high speeds of the order of Gbps. These higher speeds are expected to enable a range of new mobile applications such as augmented reality [2], virtual reality [3], autonomous vehicles [4] and machine-to-machine communications [5]. In this paper, we address the design of an end-to-end (E2E) transport protocol capable of harnessing the fast radio link layers associated with 5G. This is an important design issue because existing transport protocols such as TCP [6] do not perform well with mmWave link layers. While mmWave radio channels have access to abundant bandwidth, they suffer from rapid fluctuations in signal strength, which can lead to a significant slow-down in the actual data transfer speed available to applications running on 5G mobile devices. These fluctuations in received mmWave signal strength are caused by blockage due to buildings or human body movements and various environmental factors such as oxygen/rain absorption [7], [8]. Previous measurement studies [9] confirm the fact that mmWave bands are highly susceptible to blockages and mobile users may experience rapid and intermittent interruptions in application layer services [10].

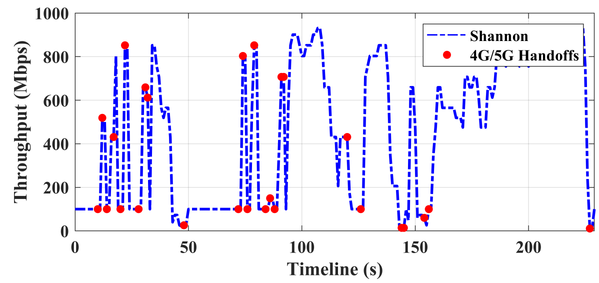


Fig. 1: 4G and 5G NR Shannon throughput performance.

These unique PHY layer characteristics of mmWave represent a design challenge to the upper layers of the network protocol stack, motivating a redesign of E2E transport protocols [11]. In particular, several studies have shown that the most commonly used reliable transport protocol, TCP, performs poorly over mmWave links [8], [12]. Using TCP for mmWave results in frequent timeout followed by slow growth of congestion window [9], [13] due to frequent switching and handoffs between the line of sight (LoS), non line of sight (nLoS) and sub 6 GHz paths. Such behavior often confuses the application layer logic [14], [15]. For example, bit-rate adaptation of video streams can further decrease the overall quality of service (QoS) and throughput. The authors of [9], [16] evaluated the performance of a commercial 5G mmWave non-standalone (NSA) Verizon network and found that the handovers due to signal fluctuations is a major performance issue that needs to be addressed in mmWave systems. A user frequently switches between various 4G/5G paths as shown in Fig. 1, an example Shannon bit-rate trace of a mobile user experiment conducted in downtown Minneapolis street with 5G capable phone [16]. The traces were obtained by considering LTE/4G bandwidth of 10 MHz and NR/5G bandwidth of 150 MHz, sampled every 1s. These events causes TCP to react as if the channel were congested, by timing out and gradually increasing the bit-rate following the AIMD (additive increase, multiplicative decrease) procedure.

In this paper, we overcome the problem of TCP probing and wastage of bandwidth during this phase by proposing ***mmCPTP, a novel cross-layer E2E transport protocol*** which uses concepts from information centric networks (ICN) [17], [18] to fully utilize the bandwidth of the mmWave access link. We use a pull-based mechanism on the fluctuating radio link to opportunistically pull the packets based on the resource availability at the base station (BS) or access point (AP). The goal is to ensure that data is always available to send to the mobile client when the radio PHY speed is high. To further reduce control latency associated with the transport protocol, we introduce a transport proxy (cache) closer to the user. The PHY aware cross-layer pull mechanism is only used between the BS and proxy, and if the content is not present at the proxy, traditional protocols which use AIMD mechanisms, such as TCP can still be used to retrieve the content from the file-server to the proxy. Our solution eliminates frequent probing as in TCP and thus makes it possible for data transfer speeds to match available 5G link bandwidth as resources become available. The major contributions of this work are as follows:

- We propose mmCPTP, a novel cross-layer pull-based transport solution to work with mmWave channel, whilst achieving high throughput efficiency and resiliency from outages.
- The proposed scheme is implemented in 5G NR (New Radio) stack and is validated using NS3 system simulation as well as on ORBIT/COSMOS testbeds [19], [20] against various TCP versions over intermittent mmWave channel.

The remainder of this paper is structured as follows. In section II, we provide benchmarks for TCP performance with 5G channels. Section III outlines design considerations for a new transport protocol such as mmCPTP. Section IV and V present design concepts and design details of our proposed mmCPTP protocol. In section VI and VII, we discuss protocol overview and implementation details. Section VIII describes our evaluation methodology and performance results. Section IX summarizes related work. Section X concludes the paper with a brief note on future work.

II. TCP Performance with mmWave

Short wavelength mmWave bands are subjected to attenuation and blockages [21] resulting in higher and more variable pathloss than in sub-6 Ghz bands. Frequent switching between LoS and nLoS links in the mmWave causes the buffers at the intermediate router to overflow [22] due to the high difference in the data rate triggering TCP fast retransmit mechanisms and timeouts required for adjusting to the new available bandwidth. Upon re-transmission timeout (RTO, usually set as 1s), the low slow start threshold causes TCP to take longer time to achieve optimum capacity due to high round-trip-time (RTT). On each successful handover, TCP probing takes a considerable amount of time and sometimes it is not even possible to reach the peak mmWave capacity [13]. Here, we present some simple evaluation results, in order to get an idea of potential throughput gains that could be achieved with a redesigned protocol. We start with an off-the-shelf mmWave transceiver (InterDigital EdgeLink [23] - 60 GHz, 802.11ad) on the COSMOS sandbox [19], [24], and then move to an emulated mobile user scenario implemented on the ORBIT [20] radio grid testbed.

The first set of lab measurements was carried out with the EdgeLink mmWave transceivers. The Tx/Rx devices were placed 25ft apart and the link was subjected to physical blockage at a distance 15ft from the Tx for over 5s interval.

Experimentation Setup	
Tx-Rx distance	25 ft
Blockage interval	5 s
Ping test (RTT)	0.295 ms
Throughput (Mbps, 100s interval)	
LoS	811.00
$\pm 10°$ Tx shift	451.00
$\pm 30°$ Tx shift	8.52
During blockage	Zero

TABLE I: InterDigital EdgeLink TCP performance on COSMOS.

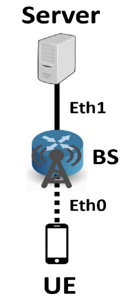
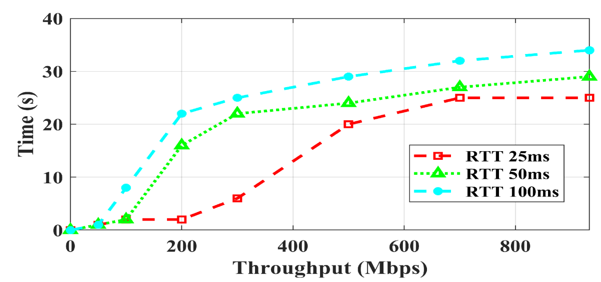


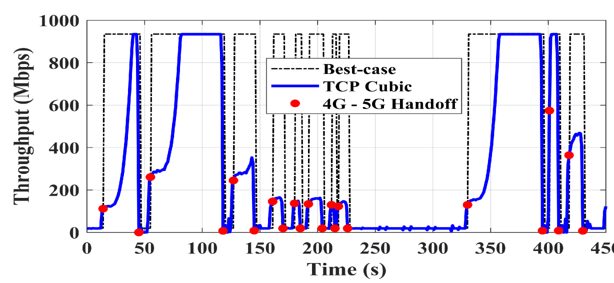
Fig. 2: Topology setup.

From Table I we see that with a slight shift in Tx orientation, the throughput drops as much as 10x when compared to the LoS scenario having a max throughput of 811Mbps. Also, the mmWave link is highly susceptible to blockages as the throughput drops to zero during the blockage interval.

We also evaluated the recovery time from the outages under different RTT settings using a 3 node experimental setup on ORBIT testbed (shown in Fig. 2). We measured the time required by the rate adaptation algorithm to switch from the lowest setting of 20Mbps to the peak value of 1Gbps and as shown in Fig. 3a, the TCP recovery time from 20Mbps to 1Gbps can be as high as 34s for RTT 100ms (mimicking distant server location). As observed, higher the RTT, the more time it takes to reach optimal capacity. This is mainly due to the presence of a long feedback loop in TCP potentially resulting in the starvation of data in the critical radio link.



(a) Recovery from nLoS to LoS



(b) Low mobility under mmWave channel

Fig. 3: TCP Cubic performance on ORBIT testbed.

As the next step, we considered the performance of TCP (Cubic) for a mobile user scenario via emulation of the available handoff model with traffic shaping of an Ethernet link in the testbed setup. The emulation switches between a maximum of 1Gbps and a minimum of 20Mbps (RTT set to 50ms), based on the mobile device mobility traces obtained from [9]. The standard Linux tool *tc* was used to limit the bandwidth. As observed from Fig. 3b, after each handoff event, TCP reacts slowly taking several seconds to reach possible peak mmWave speeds. This is clearly observed at time 15s, 50s, and 330s in the figure. Running the experiment over the entire 450s mobility trace, the achievable data transfer with TCP is 16 GB, while the theoretical maximum is 29.14 GB. This means that the use of TCP for transport incurs a 1.82x penalty over *best-case* transport protocol considering all PHY layer channel limitations, motivating our work on a new transport layer design that approaches the upper limit of achievable performance. From the above experiments, we can infer that there is a serious performance penalty associated with the use of TCP over a high bandwidth mmWave link. Our proposed transport scheme uses a novel pull mechanism to receive packets with the main objective of efficient utilization of high bandwidth-delay product (BDP) links when available.

III. Design Objectives

In this section, we present below the set of design objectives for a transport protocol comprising a heterogeneous network, having mmWave and sub 6 GHz links, with the following set of assumptions- user stationary or mobile, long-lived traffic flow, and sufficient resource availability at the BS and core.

Support mmWave intermittency and capacity: Considering TCP, the probing phase takes a considerable amount of time [9], [13] and bandwidth to reach optimal capacity. This phase often underutilizes the network since there are frequent fluctuations between various available paths and probing for all these short term paths consumes valuable time and resources. The desired protocol should reach link layer capacity as early as possible.

Support seamless mobility: Frequent handoffs occurring between sub 6 GHz and mmWave paths [10] causes TCP to reset the connection with each handover (due to the change in the IP address assignment if managed by different entities). This, in-turn causes timeout and retransmission of packets from the end host due to the packets being dropped as they cannot be forwarded. The protocol should provide seamless mobility among various available networks and moreover, it should also be capable of supporting in-network rebinding/re-routing of data packets based on the current user location.

Congestion control and packet loss: The intermittent wireless link having high bit-error-rate (BER) and packet loss, degrades the performance of transport protocol (in TCP) as it conflates loss due to link failure as congestion invoking rate limiting mechanisms to minimize congestion. Also, providing E2E Gbps throughput over high RTT is challenging in mmWave due to its propagation characteristics and intermittency [7]. A better mechanism for congestion control and packet loss is needed to achieve high data transfer speeds over mmWave access channels.

IV. mmCPTP Design Concepts

Based on the above objectives, the key design principles behind mmCPTP are described in this section. The high level architecture of mmCPTP is shown in Fig. 4. The link between BS and proxy uses the concept of receivers pulling the data based on an interest profile or globally unique identifier depending on the specific ICN technique used.

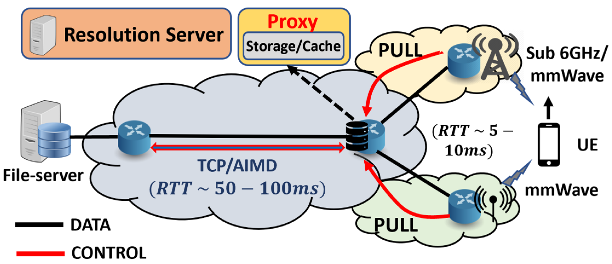


Fig. 4: High level mmCPTP protocol overview.

Avoid probing: To reduce the effect of fluctuations in mmWave band, we introduce *a novel pull mechanism* to retrieve packets from the file-server present closer to the user. There is no probing involved as the packets are pulled based on the feedback from the lower layer (RLC/MAC) buffer information. This particularly helps in mmWave scenarios since there are frequent outages and probing/reacting to all these temporary paths causes the channel to remain underutilized right after signal outages. Our solution sits on top of the last hop bottleneck link thus allowing us to use pull mechanism.

Split connection: If the content (file-server) has significant RTT to the user, to provide low latency and better throughput efficiency, we break the link into wired and wireless parts [25] with the help of a *proxy*. Such an approach can perform better in a highly dynamic scenario where the intermediate proxy node acts like a temporary cache/anchor to store packets received from the distant file-server and subsequently transfer temporarily stored data to the mobile user. In doing so, we fully utilize the potentially high BDP backhaul network by preventing the sender from limiting the rate due to frequent interruptions, thus, avoiding E2E retransmissions and minimizing latency.

Separate packet loss and congestion control: To overcome the issues faced by TCP in handling packet loss events and also to prevent any adverse impact because of this on the higher layers, we use a *separate mechanism* (only between proxy and user) for packet loss recovery and congestion control. The loss over the wireless access link is taken care by the lower layers of the stack such as link/MAC so that the transport layer [26] only needs to be concerned with flow and congestion control.

V. Design Details

In this section, we present further design details of our proposed mmCPTP transport protocol.

Flow and congestion control: *1) PULL based congestion and flow control:* The pull based data transfer is subjected to pull buffer occupancy on a per-user/UE basis at the transport layer of BS or AP. The data packets are pulled from the file-server/proxy by the BS depending on specific pull mechanisms as described next and the received packets are further sent to the lower RLC/MAC layer based on their buffer status report. The reason we use pull is to initiate traffic on the bottleneck link where we know how much to send every time or alternatively how much to pull every time. The described pull process eliminates probing and allows for fast fluctuations between LoS and nLoS paths, thus keeping the radio channel better utilized than TCP (please refer to section VI and VII for more details).

2) PULL mechanisms: Multiple pull techniques for data retrieval can be used at the BS, for example; a) *Periodic pull:* packets are periodically pulled based on current pull buffer occupancy. The period T can operate at a much larger timescale and ideally, the buffer size should be the product of *max-last-hop BW * periodicity* (in bytes or packets); b) *Threshold based:* pull buffer occupancy is periodically monitored and if it is less than a certain predetermined threshold, packets are pulled to fill the buffer. Here, the size can be very low but at the cost of an increased number of pull requests.

3) AIMD based flow control: TCP which extensively relies on AIMD, is used between the file-server and proxy provided the content is not present at the proxy, for a guaranteed fair share of bottleneck link capacity. The main rationale behind this is, TCP performs fairly well in the core and the throughput reduction issue only exists in the wireless medium [6], [27].

Error recovery: Reliable delivery of data packets is always ensured between the components of our transport protocol. TCP takes care of reliably delivering packets between the file-server and proxy, whereas between the BS and proxy, *SACKs* are used to selectively acknowledge proxy in the event of any packet loss. Furthermore, *NACK* (Negative Acknowledgement) is used to indicate handover along with the id of the last packet sequence successfully delivered from the source BS to UE, and *ACKs* (Acknowledgement) between UE and BS are used to indicate any loss in the wireless medium. In addition to this, lower layer error recovery techniques such as HARQ, also help in reliable packet delivery.

Cross-layer plug-in: Our scheme is designed and implemented as a cross-layer BS plug-in with no major changes at the sender and mobile client stacks. This new functionality is mostly at the BS as it needs to have access to the L2 information. Since our protocol involves a cross-layer design (involving L2 and L4), it can work with different L3 options (IP + 3GPP or NDN [17] or MF [18]) allowing for incremental as well as backward compatible deployment. We also provide an abstraction layer for unified access over WiFi, LTE, NR, or any future L2 protocol.

VI. Protocol Overview

To illustrate the operation of mmCPTP over IP network, we consider a UE (X), BS (B), proxy (P), and file-server (S) having respective IP addresses and port numbers as shown in Fig. 5. At first, the client UE resolves Y.URL (address of content Y) to a nearby proxy using the resolution server. Next, the UE requests for a content Y from the proxy. If the content is present locally at the proxy, it is retrieved using pull, or else an `HTTP_REQUEST()` is made to the file-server by the proxy requesting for content Y. A TCP session is setup between the file-server and proxy to receive data packets. The proxy immediately notifies UE for any

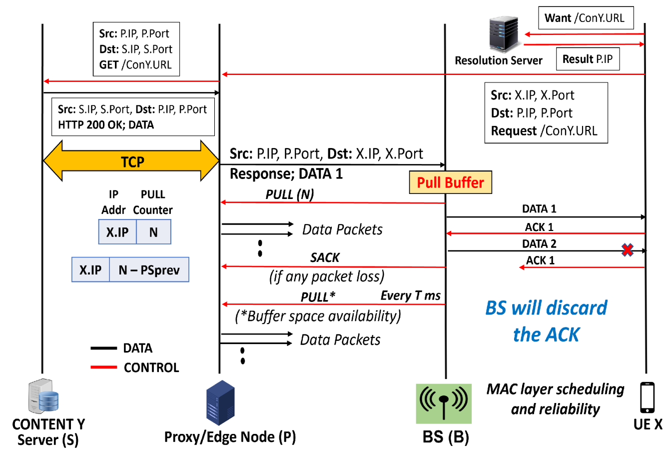


Fig. 5: Protocol overview - Timing diagram.

failure in `HTTP_REQUEST()`, and if required the UE can retry requesting the same content again. As the packets start arriving, it is cached locally at the proxy and a packet is sent to the UE (via BS) initiating data transfer. The BS on receiving this packet allocates a per-UE pull buffer (if it is the first time and no buffer has been initialized for that UE before) and requests data packets from the proxy using pull. On receiving the pull, a counter is initialized at the proxy and it starts sending requested data packets to the UE. The pull packet, issued either periodically (every T ms) or threshold based, depends on the packets being successfully delivered to the UE during the previous pull interval. The BS receives these data packets and selectively acknowledges (SACK) the proxy if there is any packet loss. Otherwise, based on the MAC layer scheduler, the BS will deliver these packets to the UE. The UE acknowledges (ACK) the received data packets and the BS will discard these ACKs, perform re-transmissions if required and clear the buffer based on the received ACKs. As the proxy receives the request for N packets, it keeps sending N data packets (N is considered as window size). The proxy will decrement the pull counter based on the number of data packets being sent and will increment with every new pull request. Also, with every new pull request the proxy can clear the corresponding number of bytes from its cache.

In the event of handovers between the BSs, proxy helps in performing the switch and ensures the reliable transfer of data. The source BS issues NACK to the proxy, indicating unsuccessful delivery of packets (and also handover) along with their IDs. Based on this feedback message the proxy will reroute the packets from the previously undelivered sequence to a new user location (target BS). Since the proxy is located at an infrastructure end, it expects less failure [25]. In any case, the states are always stored and if required the proxy will re-initiate a new session with the file-server and subsequently transfers data to the UE.

VII. Implementation Details

This section discusses the mmCPTP implementation details considering service layer abstraction, integration with 5G NR stack, and design details involving Click [28] software.

Pull based abstraction layer. The purpose of this layer is to provide a unified pull based transport access across different technologies ranging from WiFi, LTE, NR, or any future L2 protocol. It consists of a pull buffer of predetermined size and various interfaces to access the sockets required to send/receive packets to/from the lower radio layers and file-server/proxy respectively. This abstraction layer is only present at the BS since it does the pulling based on pull buffer occupancy and lower layer buffer information. As shown in Fig. 6 (considering the NR stack), the `Send_Pull()` is used to receive data packets to fill the pull buffer. The `Report_Buffer_Status()` is used to get information on the lower layer buffer irrespective of the radio access technology being used and is called for every MAC scheduling interval. In the case of LTE/NR stack, the RLC buffer is being reported whereas, in the case of WiFi, it is the status of the MAC queue that is being reported periodically to the abstract layer. Based on this buffer report, packets are sent out using `Push_Data_Packets()` from the pull buffer to lower RLC or MAC queues. Later, from these queues, the packets are dequeued to the UEs based on MAC scheduling decisions.

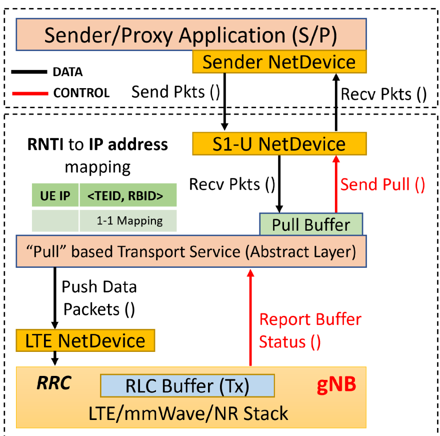


Fig. 6: Integration of mmCPTP with 5G NR stack.

Integration with 5G NR stack. We modified the 5G NR stack to support mmCPTP as shown in Fig. 6. For our implementation, we leveraged most parts of the code and support from the NS3 LTE stack, since NR will be using the same RRC, PDCP, and RLC stack as in LTE but different PHY/MAC layers [29]. Our transport service sits on top of the PDCP layer and there are various socket interfaces for sending and receiving packets as follows. Sockets *S1-u NetDevice* and *LTE NetDevice* are used to send/receive packets to/from the remote server and lower radio layer respectively. Between UE and packet gateway (PGW) there exists a default GTP/UDP/IP tunneling procedure where information on tunnel endpoint identifier (TEID), radio network temporary identifier (RNTI), and radio bearer identifier (RBID) are required to setup the tunnel with appropriate QoS identifier. A one-to-one mapping between the UE IP address and these identifiers is present which is then required by our abstract layer to allocate appropriate pull buffer size and access RLC buffer status. An RLC service access provider (SAP) is used by the transport

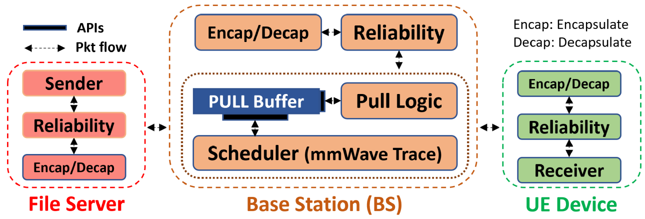


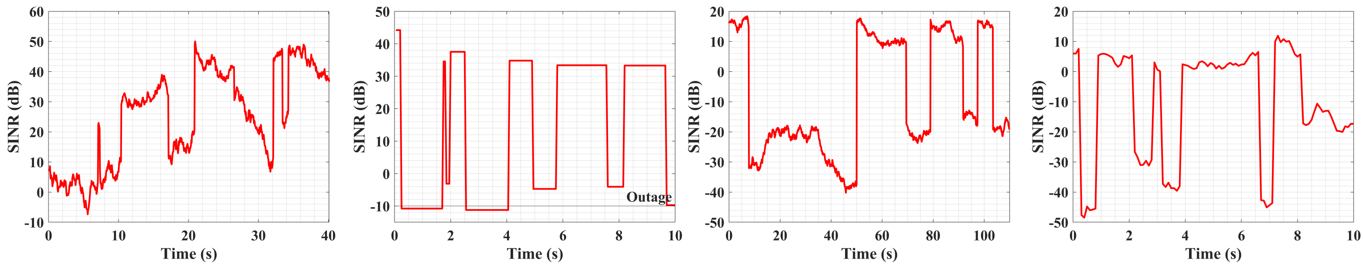
Fig. 7: Click element graph implementation of mmCPTP.

layer to periodically access the RLC buffer status, which is required to push in data packets from the pull buffer to the RLC for further delivery to the user. Also, note that mmCPTP is independent of any MAC layer scheduling decisions.

Click based protocol modules. We implement mmCPTP on ORBIT using Click modular software [28]. Click provides an easy to configure modular schema individually for each of the components as shown in the element graph (Fig. 7). The *FromDevice* and *ToDevice* elements are configured with their respective Ethernet addresses. At the BS, a trace driven *Scheduler* is implemented which schedules bytes in terms of packets, based on trace reporting interval. The pull buffer periodically pulls the data based on buffer occupancy (*PullLogic*) from the file-server and uses the *Scheduler* element to push data to the UE. The *Reliability* element ensures reliable transfer of data in a hop-by-hop fashion. Finally, the *Encap* and *Decap* elements are respectively configured to encapsulate and decapsulate the packets with IP headers to enable routing.

VIII. Performance Evaluation

We evaluate the performance of the proposed mmCPTP transport scheme in 5G NR and also trace driven emulation over Gbps Ethernet channel using NS3 simulation (v3.33) and ORBIT testbed. The mmWave channel traces were obtained from NYU mmWave simulator [29]. The deployment includes a mobile UE surrounded by buildings modeled as commercial type having concrete walls and height uniformly distributed between 30m and 40m, deployed within 500m [X] * 100m [Y] area having a certain minimum distance between them to avoid overlapping. We consider four mobility models: 1) UE moving horizontally (along X axis, towards origin) with a speed of 15m/s, 2) UE moving vertically (along Y axis) with a speed of 15m/s, 3) Random walk mobility model, where every 10s the UE changes its speed and direction based on a normal distribution with mean 2m/s and variance 0.04m/s and 4) an emulated channel model, where outages are emulated by moving UE far away (nLoS) from the BS for a time uniformly distributed between 0.1s to 2s and vice versa. For models 1 and 3, the BS is deployed at (25m, 120m, 10m), whereas for models 2 and 4, the BS is deployed at (25m, 50m, 10m). The UE originates at (550m, -20m, 1.6m) for mobility models 1 and 2 whereas, for a random walk, the UE chooses a random coordinate within the deployment area. The channel traces for all these mobility models are shown in Fig. 8. In the case of NR based evaluations, the underlying channel used is LTE having BuildingsChannelCondition model (see Fig. 8a and Fig. 8b) whereas, for Ethernet based emulation, mmWave channel (see Fig. 8c and Fig. 8d) of DL bandwidth 1 GHz



(a) Mobile UE (horizontal-X) (b) Emulated UE outages (c) UE random walk (d) Mobile UE (vertical-Y)

Fig. 8: SINR traces - a) and b) underlying channel is LTE; c) and d) underlying channel is mmWave (DL bandwidth 1 GHz).

is considered. More details on the simulation parameters and assumptions can be found in Table II.

TABLE II: System Simulation Parameters.

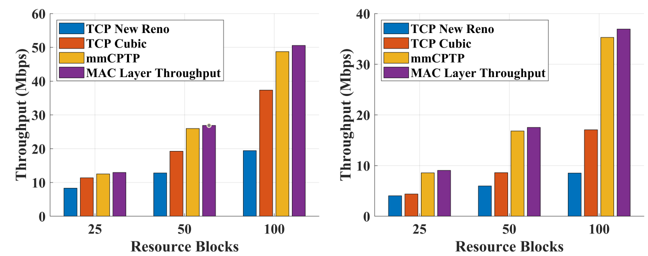
Simulation Parameter	Value
Deployment scenario	Single mmWave BS & UE in downtown area
Path loss model	3GPP Uma Propagation & Hybrid Buildings (Fig. 8b)
Channel	Buildings Channel Condition Model (3GPP)
Deployment area	500m x 100m
Number of buildings	15
Buiding dimension	Variable (height: 30 - 40m)
Carrier frequency	2 GHz (NR/LTE) & 28 GHz (Emulation)
DL Bandwidth	RBs: 25, 50, 100 (NR/LTE) 1 GHz (Emulation)
Tx power	BS: 30 dBm, UE: 30 dBm
BS height	10m
UE height	1.6m
UE mobility model	Constant velocity (15m/s) & outdoor 2-d random walk
Congestion control	TcpNewReno and Cubic
Sampling interval	100ms

A. Evaluation with 5G NR stack

1) System setup: In NR/LTE, the evaluations were carried out using topology consisting of a file-server, evolved packet core (EPC), BS, and a UE (RTT∼25ms). The mmCPTP module at the BS periodically pulls data packets from the file-server.

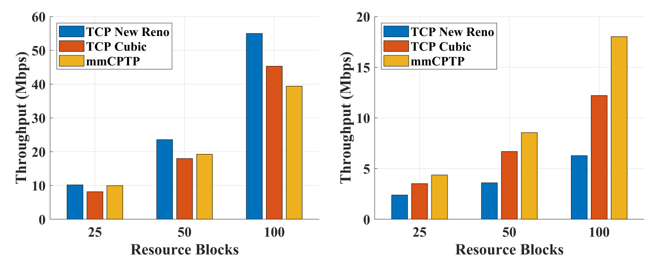
2) Single UE: We evaluate the performance of mmCPTP for a single user considering channel traces as shown in Fig. 8a and Fig. 8b. The simulation is run for 40s and 10s respectively for each of these traces and from Fig. 9a, we see that, for high speed mobile UE (moving along X axis), mmCPTP achieves a max throughput gain 2.5x times as compared to TCP New Reno and 1.3x times as compared to TCP Cubic congestion control mechanism. The gain increases with the increase in resource block size or bandwidth. With the emulated UE outage model, the gain observed is fairly high as compared to the mobile UE model since we are randomly injecting outages and our protocol is meant to optimize these temporary outages. From Fig. 9b, we see that mmCPTP achieves a gain close to 4x and 2x as compared to New Reno and Cubic respectively.

3) Multiple UEs and fairness: The impact of having multiple users is studied using the same simulation setup as before.



(a) Using Channel Fig. 8a (b) Using Channel Fig. 8b

Fig. 9: Performance of mobile UE in 5G NR.



(a) UE 1 - Stationary (LoS) (b) UE 2 - Mobile

Fig. 10: Multi user NR throughput performance.

UE 1 is stationary (*loc*: 0, -20, 1.6) and remains in the LoS region all the time, whereas UE 2 is having temporary outages as shown in Fig. 8b. The eNB MAC scheduler allocates appropriate transport block size (TBS) on a per subframe basis to these users. We evaluate the throughput performance of mmCPTP against New Reno and Cubic for a 10s simulation interval. From Fig. 10, we see that mmCPTP performs well among other considered protocols for the mobile user (UE 2), but the performance deteriorates for a stationary user (UE 1). This behavior is expected and it is because of the Proportional Fair scheduler. The radio resource blocks are allocated proportionally to these UEs and since our protocol is suitably designed for UEs having frequent outages or blockages, it achieves a max throughput gain of 2.8x and 1.5x times as compared to New Reno and Cubic schemes respectively for UE 2 (see Fig. 10b). We also compute the Jain's Fairness Index (JFI) [30] to validate the fairness of our protocol over other TCP variants. The JFI is given in equation 1, where x_i characterizes throughput proportional to the maximum achievable MAC throughput per user and n represents the total number of users.

$$J(x_1, ..x_n) = \frac{(\sum_{i=1}^{n} x_i)^2}{n.\sum_{i=1}^{n} x_i^2} \quad , x_i = \frac{APP.\,tpt\,of\,UE_i}{MAC\,tpt\,of\,UE_i} \qquad (1)$$

As seen in Table III, mmCPTP provides a better fairness index as compared to other TCP versions, since it operates closely with the MAC scheduler and lower layer buffer status, thus ensuring better resource utilization.

TABLE III: JFI under multi user setting.

Transport Schemes	Resource Blocks (RBs) 25	50	100
TCP New Reno	0.9123	0.8320	0.7828
TCP Cubic	0.9998	0.9983	0.9765
mmCPTP	0.9999	0.9978	0.9889

B. End-to-end simulation

1) System setup: In addition to the evaluation of mmCPTP over NR stack, evaluations were also carried out using trace driven emulation implemented with a traffic shaped ethernet link emulating radio link's channel traces. The system topology is shown in Fig. 11, consisting of a file-server, BS, UE, and a proxy between the end hosts. The SINR channel traces were mapped to their respective bit-rate based on the TBS and are injected into the simulator sampled every 100ms. A TCP session between the file-server and proxy is considered in our simulation. The performance comparison of our scheme with respect to TCP and Indirect-TCP (I-TCP) [25] is evaluated against a large content transfer application.

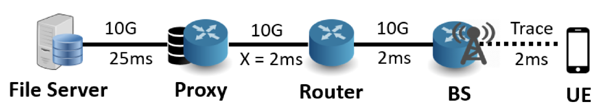


Fig. 11: Simulation setup for trace driven mmWave emulation.

2) Content delivery over mmWave channel: We evaluate the performance of mmCPTP using the outdoor random walk mobility setting. The SINR corresponding to its trajectory is shown in Fig. 8c respectively for 110s of simulation duration. From the traces, we see that during the LoS path, the SINR can be as high as 20dB, but in the event of NLoS i.e. when the links get blocked by buildings (between 10-50s interval) the SINR drops to -40dB leading to the low data rate. Fig. 12 presents the file transfer size vs. time for various transport schemes, and we observe that mmCPTP is quickly able to reach optimum capacity as compared with I-TCP and TCP. As observed with mmCPTP, a gain of 31% can be obtained against I-TCP considering Cubic. TCP performs poorly both with Cubic and New Reno schemes as only 970MB and 800MB of data are respectively transferred during the entire user trajectory. In the congestion avoidance phase, due to the presence of high RTT, the recovery time for TCP during frequent interruptions is very high, leading to low utilization of the mmWave channel.

3) Impact of RTT and loss rate on throughput: To evaluate the performance of mmCPTP under different RTT and loss rates, we perform a similar experiment as before considering channel as shown in Fig. 8d. We repeat this experiment for different network parameters by varying RTT (X set as 5ms, 10ms, 15ms) and introducing loss at the last-hop mmWave link (0 or 0.1% loss). From Fig. 13, we observe that the throughput is relatively high and remains fairly stable in all the cases

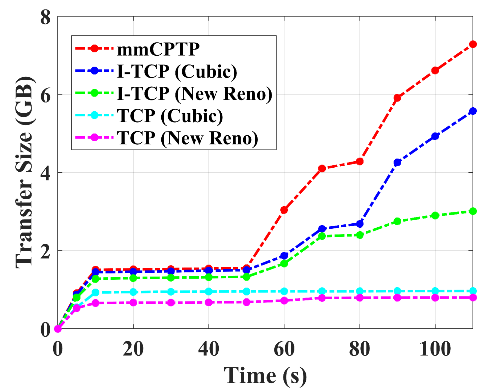


Fig. 12: File transfer for random walk (channel: Fig. 8c).

considered for mmCPTP, mainly because there is no feedback loop as in the case of TCP or I-TCP and any packet loss is recovered locally. The throughput of I-TCP drops significantly in the presence of high RTT and packet loss. A gain of 4.5x is observed with our scheme as compared to I-TCP (Cubic), having X set to 15ms and a loss ratio 0.1%.

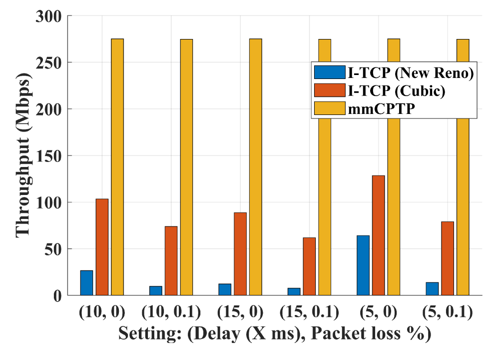


Fig. 13: Throughput for different delay and loss rate (High speed mobile UE, channel: Fig. 8d).

C. Real-time ORBIT emulation

To further validate the performance of mmCPTP on a real system, we implement it in Click software (as described in section VII) and compare its performance against various TCP versions on ORBIT, using the channel as shown in Fig. 1 and topology as given in Fig. 2. A standard Linux package tool *tc* is used to limit the bandwidth and introduce any additional delays between the nodes. From Table IV, we see that having a low RTT of 0.3ms (default), all the TCP schemes perform fairly similar, and a total of 11.80GB of data is transferred on average along the entire user trajectory. But the performance degrades with the increase in RTT due to the presence of a longer feedback loop. A performance gain of 17.2x and 5.7x is observed with mmCPTP as compared to New Reno and Cubic, respectively with RTT 50.3ms. The proposed mmCPTP is also very close to the theoretical maximum transfer size of 13.27GB. When compared against other existing TCP versions supporting high BDP links such as TCP HighSpeed [31], Yeah [32] (Yet Another HighSpeed TCP) and also BBR [33] (Bottleneck Bandwidth and Round-trip propagation time; model based), mmCPTP still outperforms these schemes in all the considered scenarios. These results demonstrate that the proposed mechanisms such as pull, cross-layer feedback support, separation of packet loss recovery, and congestion control from the transport layer, help highly intermittent mmWave links to reach optimal bottleneck capacity as early as possible and improve goodput.

TABLE IV: File transfer size in GB over different RTTs using channel as shown in Fig. 1 on the ORBIT testbed.

Transport Schemes	**Round Trip Time (RTT)**			
	0.3ms	**5.34ms**	**10.3ms**	**50.3ms**
TCP New Reno	11.80	3.34	2.31	0.75
TCP Cubic	11.80	4.07	3.20	2.24
TCP HighSpeed	11.80	5.90	3.78	1.46
TCP Yeah	11.80	10.70	10.20	6.96
TCP BBR	11.70	11.60	11.50	11.20
mmCPTP	12.96 *(average)*			
Max	13.27 *(theoretical limit)*			

IX. Related work

TCP [34] being the de-facto standard transport protocol for reliable transmission performs poorly in certain wireless scenarios [35]. Various MAC and RLC [12], [36] layer schemes have been proposed to mask the last hop losses from the TCP layer, especially for mmWave networks. Yet, this causes increase in E2E latency as the packets are queued temporally during NLoS scenarios. Proxies or middlebox based solutions have also been proposed for TCP, specifically targeting wireless scenarios to improve performance. In [37], a milliProxy is described following TCP semantics. The milliProxy allows controlling maximum segment size (MSS) and congestion window separately between wired and wireless parts thus improving goodput and reduction in latency for high bandwidth delay product (BDP) mmWave links. The authors in [38] proposed a proxy based TCP architecture called mm-PEP, breaking TCP E2E semantics. The module installed at the base-station helps in improving the packet delivery ratio by maintaining the sending rate during LoS/NLoS switches. Various other schemes such as I-TCP [25], M-TCP [39] adopt a split TCP approach by breaking the TCP session between wired and wireless parts of the network for boosting throughput and improving BER. However, all the above discussed schemes have the same underlying TCP session governed by probing, which often complicates the overall process.

Several other congestion control algorithms have been proposed particularly targeting high BDP links such as HighSpeed TCP [31], TCP Yeah [32], H-TCP [40], etc. Yet, they perform poorly over mmWave channel [6], [41] resulting in high latency and slow recovery in the event of handovers. TCP BBR, recently proposed by Google [33] operates by estimating bottleneck bandwidth and RTT. It enters a state called "probeRTT" having a smaller congestion window (several kB) periodically to measure real-time RTT. However, this is unfriendly for streaming applications and traffic over the Internet since the traffic might choose different routes and get different RTTs. The protocol is still under development [42] and also has issues on coexistence with other loss-based algorithms. QUIC [43] also proposed by Google deals better with packet loss and achieves faster connection establishment. Nevertheless, the problem of probing also exists in QUIC since it uses the same congestion control mechanism as in Reno or Cubic [44]. Hence, due to various implementation challenges, we restrict our study to New Reno and Cubic for the most part of the work. A detailed evaluation of these newer end-to-end transport protocols against mmCPTP, with more complex topologies and multiple users is left for future work.

ICN based architectures such as Named data networking (NDN) [17], use a receiver driven content retrieval model based on named content. Since ICN protocols are receiver driven, interest shaping can proactively control the congestion. The classical AIMD algorithm used by TCP for congestion control is used by NDN receivers to control interest rates. In addition to the consumer based congestion control, Hop-by-hop transport control [26] at routers have been well studied for FIAs such as NDN and MF [18]. Yet, the above described transport solutions still depend on AIMD for probing the bottleneck bandwidth and perform poorly as in TCP when there is high fluctuation in bandwidth and switching between alternate paths.

X. Conclusion

In this work, a novel cross-layer assisted pull based transport protocol is presented to overcome the deficiencies of TCP in mmWave networks. Fluctuating bit-rate and frequent handoffs between various available paths often underutilizes the mmWave links due to the gradual probing and additive increase multiplicative decrease behavior of TCP. The proposed scheme "mmCPTP" is based on an in-network proxy where the BS periodically fetches data based on cross-layer information from the lower layer (RLC/MAC) of the stack. In doing so, we avoid the slow start probing phase required to probe the available bandwidth. The evaluation of the proposed protocol was carried out using NS3 simulation considering high speed mobile UE and random walk user traces. The results show that when compared against conventional protocols, a gain of 4.5x is observed with respect to I-TCP (Cubic) in the presence of high RTT and packet loss. The protocol performs quite well in the presence of multiple users considering the NR stack, achieving a performance gain of 2.8x and 1.5x relative to New Reno and Cubic respectively. Also, when evaluated with a real-time implementation on the ORBIT [20] testbed, a gain of 17.2x and 5.7x is observed with mmCPTP, as compared to New Reno and Cubic respectively in scenarios with high RTT. The simulation results and the experimental validation demonstrate the feasibility of our cross-layer assisted transport protocol which has been shown to maintain better channel utilization in mmWave access scenarios. Future work includes extending the mmCPTP design to multi-homing scenarios with user mobility. Field trials using the outdoor COSMOS [19], [24] testbed are also under consideration.

XI. Acknowledgements

This work was supported in part by NSF PAWR COSMOS grant CNS-1827923 and OAC-2029295.

References

[1] S. E. Elayoubi, M. Fallgren, P. Spapis, G. Zimmermann, D. Martín-Sacristán, C. Yang, S. Jeux, P. Agyapong, L. Campoy, Y. Qi *et al.*, "5G service requirements and operational use cases: Analysis and metis ii vision," in *2016 European conference on networks and communications (EuCNC)*. IEEE, 2016.

[2] R. T. Azuma, "A survey of augmented reality," *Presence: teleoperators & virtual environments*, 1997.

[3] G. Burdea and P. Coiffet, "Virtual reality technology," 2003.

[4] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, 2015.

[5] K.-C. Chen and S.-Y. Lien, "Machine-to-machine communications: Technologies and challenges," *Ad Hoc Networks*, 2014.

[6] M. Zhang, M. Polese, M. Mezzavilla, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi, "Will TCP work in mmwave 5G cellular networks?" *IEEE Communications Magazine*, 2019.

[7] Y. Niu, Y. Li, D. Jin, L. Su, and A. V. Vasilakos, "A survey of millimeter wave communications (mmwave) for 5G: opportunities and challenges," *Wireless networks*, 2015.

[8] S. Rangan, T. S. Rappaport, and E. Erkip, "Millimeter-wave cellular wireless networks: Potentials and challenges," *Proceedings of the IEEE*, 2014.

[9] 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," in *Proceedings of The Web Conference 2020*, 2020.

[10] 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.

[11] X. Wang, L. Kong, F. Kong, F. Qiu, M. Xia, S. Arnon, and G. Chen, "Millimeter wave communication: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, 2018.

[12] M. Polese, R. Jana, and M. Zorzi, "TCP and MP-TCP in 5G mmwave networks," *IEEE Internet Computing*, 2017.

[13] D. Xu, A. Zhou, X. Zhang, G. Wang, X. Liu, C. An, Y. Shi, L. Liu, and H. Ma, "Understanding operational 5G: A first measurement study on its coverage, performance and energy consumption," in *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*, 2020.

[14] J. Jiang, V. Sekar, and H. Zhang, "Improving fairness, efficiency, and stability in http-based adaptive video streaming with festive," in *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, 2012.

[15] T.-Y. Huang, N. Handigol, B. Heller, N. McKeown, and R. Johari, "Confused, timid, and unstable: picking a video streaming rate is hard," in *Proceedings of the 2012 internet measurement conference*, 2012.

[16] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R. A. Fezeu, U. K. Dayalan, S. Verma, P. Ji, T. Li *et al.*, "Lumos5g: Mapping and predicting commercial mmwave 5G throughput," in *Proceedings of the ACM Internet Measurement Conference*, 2020.

[17] L. Zhang, A. Afanasyev, J. Burke, V. Jacobson, K. Claffy, P. Crowley, C. Papadopoulos, L. Wang, and B. Zhang, "Named data networking," *ACM SIGCOMM Computer Communication Review*, 2014.

[18] D. Raychaudhuri, K. Nagaraja, and A. Venkataramani, "Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet," *ACM SIGMOBILE Mobile Computing and Communications Review*, 2012.

[19] D. Raychaudhuri, I. Seskar, G. Zussman, T. Korakis, D. Kilper, T. Chen, J. Kolodziejski, M. Sherman, Z. Kostic, X. Gu *et al.*, "Challenge: COSMOS: A city-scale programmable testbed for experimentation with advanced wireless," in *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, 2020.

[20] D. Raychaudhuri, I. Seskar, M. Ott, S. Ganu, K. Ramachandran, H. Kremo, R. Siracusa, H. Liu, and M. Singh, "Overview of the orbit radio grid testbed for evaluation of next-generation wireless network protocols," in *IEEE Wireless Communications and Networking Conference, 2005*. IEEE, 2005.

[21] T. Bai, R. Vaze, and R. W. Heath, "Analysis of blockage effects on urban cellular networks," *IEEE Transactions on Wireless Communications*, 2014.

[22] 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 *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2016.

[23] "Interdigital, edgelink solution," https://www.interdigital.com/presentations/edgelink-solution, 2022.

[24] "Cloud enhanced open software defined mobile wireless testbed for city-scale deployment (COSMOS)," https://cosmos-lab.org/, 2022.

[25] A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for mobile hosts," in *Proceedings of 15th International Conference on Distributed Computing Systems*. IEEE, 1995.

[26] K. Su, F. Bronzino, K. Ramakrishnan, and D. Raychaudhuri, "MFTP: A clean-slate transport protocol for the information centric mobilityfirst network," in *Proceedings of the 2nd ACM Conference on Information-Centric Networking*, 2015.

[27] K. Liu and J. Y. Lee, "On improving TCP performance over mobile data networks," *IEEE transactions on mobile computing*, vol. 15, no. 10, pp. 2522–2536, 2015.

[28] E. Kohler, R. Morris, B. Chen, J. Jannotti, and M. F. Kaashoek, "The click modular router," *ACM Transactions on Computer Systems (TOCS)*, 2000.

[29] 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*, 2018.

[30] R. K. Jain, D.-M. W. Chiu, W. R. Hawe *et al.*, "A quantitative measure of fairness and discrimination," *Eastern Research Laboratory, Digital Equipment Corporation, Hudson, MA*, 1984.

[31] S. Floyd, "Rfc3649: Highspeed TCP for large congestion windows," 2003.

[32] A. Baiocchi, A. P. Castellani, and F. Vacirca, "Yeah-TCP: yet another highspeed TCP," in *Proc. PFLDnet*, 2007.

[33] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control: Measuring bottleneck bandwidth and round-trip propagation time," *Queue*, 2016.

[34] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM computer communication review*, 1988.

[35] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: problems and solutions," *IEEE Communications Magazine*, 2005.

[36] M. Polese, R. Jana, and M. Zorzi, "TCP in 5G mmwave networks: Link level retransmissions and MP-TCP," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2017.

[37] M. Polese, M. Mezzavilla, M. Zhang, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi, "milliproxy: A TCP proxy architecture for 5G mmwave cellular systems," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2017.

[38] M. Kim, S.-W. Ko, and S.-L. Kim, "Enhancing TCP end-to-end performance in millimeter-wave communications," in *2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2017.

[39] K. Brown and S. Singh, "M-TCP: TCP for mobile cellular networks," *ACM SIGCOMM Computer Communication Review*, 1997.

[40] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long-distance networks," in *Proceedings of PFLDnet*, 2004.

[41] J. Lorincz, Z. Klarin, and J. Ožegović, "A comprehensive overview of tcp congestion control in 5G networks: Research challenges and future perspectives," *Sensors*, 2021.

[42] Y.-J. Song, G.-H. Kim, I. Mahmud, W.-K. Seo, and Y.-Z. Cho, "Understanding of BBRv2: Evaluation and comparison with BBRv1 congestion control algorithm," *IEEE Access*, 2021.

[43] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar *et al.*, "The quic transport protocol: Design and internet-scale deployment," in *Proceedings of the conference of the ACM special interest group on data communication*, 2017, pp. 183–196.

[44] J. Iyengar and I. Swett, "QUIC Loss Detection and Congestion Control," RFC 9002, May 2021. [Online]. Available: https://www.rfc-editor.org/info/rfc9002