

Received August 31, 2020, accepted September 1, 2020, date of publication September 7, 2020, date of current version September 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3022284

# NexGen D-TCP: Next Generation Dynamic TCP Congestion Control Algorithm

MADHAN RAJ KANAGARATHINAM<sup>1</sup>, (Senior Member, IEEE),  
SUKHDEEP SINGH<sup>1</sup>, (Member, IEEE), IRLANKI SANDEEP<sup>1</sup>, HANSEOK KIM<sup>2</sup>,  
MUKESH KUMAR MAHESHWARI<sup>3</sup>, JAEHYUN HWANG<sup>4</sup>, ABHISHEK ROY<sup>5</sup>,  
AND NAVRATI SAXENA<sup>6</sup>

<sup>1</sup>Samsung R&D Institute India Bangalore, India (SRI-B), Bengaluru 560066, India

<sup>2</sup>Samsung Electronics, Suwon 443-743, South Korea

<sup>3</sup>Department of Electrical Engineering, Bahria University, Karachi 75260, Pakistan

<sup>4</sup>Department of Computer Science, Cornell University, Ithaca, NY 14850, USA

<sup>5</sup>Wireless Standards Team, MediaTek USA Inc., Irvine, CA 92618, USA

<sup>6</sup>Department of Computer Science, San Jose State University, San Jose, CA 95192, USA

Corresponding author: Sukhdeep Singh (sukh.sandhu@samsung.com)

**ABSTRACT** With the advancement of wireless access networks and mmWave New Radio (NR), new applications emerged, which requires a high data rate. The random packet loss due to mobility and channel conditions in a wireless network is not negligible, which degrades the significant performance of the Transmission Control Protocol (TCP). The TCP has been extensively deployed for congestion control in the communication network during the last two decades. Different variants are proposed to improve the performance of TCP in various scenarios, specifically in lossy and high bandwidth-delay product (high-BDP) networks. Implementing a new TCP congestion control algorithm whose performance is applicable over a broad range of network conditions is still a challenge. In this article, we introduce and analyze a Dynamic TCP (D-TCP) congestion control algorithm over mmWave NR and LTE-A networks. The proposed D-TCP algorithm copes up with the mmWave channel fluctuations by estimating the available channel bandwidth. The estimated bandwidth is used to derive the congestion control factor  $N$ . The congestion window is increased/decreased adaptively based on the calculated congestion control factor. We evaluated the performance of D-TCP in terms of congestion window growth, goodput, fairness and compared it with legacy and existing TCP algorithms. We performed simulations of mmWave NR during LOS <-> NLOS transitions and showed that D-TCP curtails the impact of under-utilization during mobility. The simulation results and live air experiment points out that D-TCP achieves 32.9% gain in goodput as compared to TCP-Reno and attains 118.9% gain in throughput as compared to TCP-Cubic.

**INDEX TERMS** TCP, congestion control, bandwidth estimation, throughput, friendliness, fairness.

## I. INTRODUCTION

Mobile networks have been making significant progress over the last decade and will be performing an important role in access networks. The next generation mobile networks (5G) are expected to provide Gbps uplink/downlink speeds, ubiquitous availability and massive connectivity. With the evolution of smart phone technology and mobile networks, many real time application has emerged demanding high throughput and low end-to-end delay, which may cause congestion

The associate editor coordinating the review of this manuscript and approving it for publication was Angelos Antonopoulos<sup>1</sup>.

at the network. The 5G networks exploit mmWave to meet the higher bandwidth and low latency requirements of different services. However, mmWave wireless links experience excessive path and penetration losses [1]. Moreover, with the introduction of next generation mobile network services like Device to Device (D2D) communications, Internet of Things (IoT), Internet of Vehicles (IoV), etc., the mobile network will change dynamically. The mobile networks are usually interconnected using wired backbone networks. The afore-said network changes will affect the transport and network layer protocol. The Transmission Control Protocol (TCP) has been adopted widely since last two decade for reliable and

error-checked delivery of data. Many applications on the network work over the Transmission Control Protocol/Internet Protocol (TCP/IP) e.g. HyperText Transfer Protocol (HTTP), streaming media, peer-to-peer file sharing, telnet, file transfer, simple mail transfer protocol, secure shell and the like [2].

Over the last decade, TCP has been extensively studied for mobile wireless networks mostly sub-6 GHz LTE/LTE-A networks. However, the impact of 5G mmWave links on TCP is much more disruptive than LTE/LTE-A networks. This is due to propagation characteristics of mmWave like: (i) The blockage sensitivity causing outage leading to slow growth of Congestion Window (CWND) and multiple retransmission timeouts; (ii) Highly variable channel fluctuations i.e. Non-line of sight (NLOS)/Line of sight (LOS) transitions. Particularly in NLOS conditions channel losses are masked with the help of re-transmissions at link level but TCP remains unaware of such variable channel conditions. Due to such characteristics of mmWave, the packets keep on buffering in different layers due to an increase in CWND. This in turn leads to bufferbloat phenomenon causing a significant impact on end-to-end performance [3]–[5]. To avoid the same, Active Queue Management (AQM) methodologies and cross-layer approach for downlink have been proposed for mmWave channel [6], [7]. There are several techniques are proposed to detect packet loss either due to congestion or due to disconnection caused by mmWave, which can be prevented with the help of retransmission timeouts. Existing research works on TCP mmWave New Radio (NR) does not explore mobility scenarios where high channel fluctuations (NLOS/LOS transitions) with multiple gNBs.

The analysis of TCP performance on live air and in more realistic scenarios with multiple base stations are not explored yet. Timely adaptation of serving gNBs is beneficial for TCP, however considering the frequent handover case in mmWave based 5G NR, it can negatively affect the throughput and end-to-end system performance [8]. The mmWave enabled 5G networks are high Bandwidth-Delay (BDP) networks. This is because of the availability of Gbps order bandwidth at the physical layer. Hence, high BDP network conditions has to be considered while designing congestion control for 5G mmWave. Moreover, while designing backhaul and core networks for mmWave based 5G NR, it is crucial to curtail NLOS <-> LOS fluctuations and end to end system performance. Therefore, it is necessary to adapt mobility techniques that minimize the delay caused due to switching between different gNBs or different radio access technologies in case of 5G HetNet scenarios. The existing TCP algorithms consider either fluctuation in bandwidth or high BDP conditions and none of them addresses both simultaneously.

In LTE/LTE-A, during edge scenarios, the standard congestion control algorithms (like TCP Reno [9], TCP NewReno [10], TCP Tahoe [11]) force to unnecessary drops in Congestion Window (CWND), which significantly decrease the efficiency of TCP mobile receivers/senders. On the other hand, few algorithms such TCP BIC [12] increase the CWND aggressively to overcome the losses, but does

not maintain inter and intra fairness. To tackle this, TCP Cubic [13] was developed to improve the scalability of TCP over high BDP conditions and maintain the fairness [14]. Since, Wi-Fi networks experiences high bandwidth fluctuation due to high loss rate. Authors in [2], [15] introduce TCP Veno, TCP Westwood to address the bandwidth fluctuation scenarios, but these variants failed to attain higher goodput as compare to TCP CUBIC. Hence, Google proposed TCP BBR [16], which mitigate the losses in high BDP scenario and adapts the fluctuating bandwidth. Some recent studies [4], [17]–[19] analyze the performance of TCP in 5G mmWave Networks. They have highlighted that high variability of the signal quality over mmWave links and blockage results in low utilization of resources and degrades the TCP goodput. Motivated from these limitations posed by 5G mmWave, we designed a dynamic congestion control algorithm to achieve the higher goodput as compared to existing algorithms while maintaining friendliness and fairness.

*Contributions:* In this paper, we propose Dynamic TCP (D-TCP) congestion control algorithm for next generation mobile networks, which not only considers the dynamics of mmWave but also adapts the High BDP and low lossy rate conditions. More precisely, the contributions of this work are summarized as follows:

- 1) In mmWave NR, the available bandwidth oscillates from a few Mbps to Gbps, as channel conditions are highly variable (transition between LOS/NLOS). D-TCP understands the varying dynamics and adapts to the mobility.
- 2) To model the D-TCP, we first estimate the available bandwidth. We apply Continuous to Discrete (C2D) time variant filter using Tustin approximation to extract the network attributes (like: traffic intensity on the link, packet transmission rate, signal conditions) and to estimate the available bandwidth.
- 3) We derive the congestion control factor ( $N$ ), using estimated bandwidth. The calculated  $N$  is utilized to increase/decrease the congestion window during the loss detection and Round Trip Time (RTT) update. Moreover, the congestion window is increased/decreased adaptively.
- 4) We derive the expression for throughput using calculated bandwidth and congestion control factor  $N$ . Subsequently, we analyze the efficiency, friendliness and fairness.
- 5) The D-TCP performance is analyzed in 5G NR and LTE-A network. The simulations are carried out using NS3-mmWave module and NS3 LTE-A module. The simulation results of D-TCP are compared with legacy TCP algorithms BBR, CUBIC, RENO, Westwood, Tahoe and the existing CLTCP algorithm [20].
- 6) Additionally, we experimented D-TCP on a live network using two Samsung Galaxy S8 configured with proposed D-TCP and TCP-Cubic algorithms. The live experiment results show that D-TCP achieve 124.92%

gain in goodput as compared to TCP-Cubic when the server is located in Seoul (135 ms RTT).

The initial version of this work is published in [21]. In [21], we proposed a Dynamic TCP congestion control algorithm (D-TCP) for LTE-A networks. D-TCP estimates the channel bandwidth and adjusts the congestion window dynamically. The performance analysis carried out using NS3 based simulation experiments. However, this work lacks in analysis and performance evaluations of D-TCP over mmWave NR. This paper extends the work [21] in several aspects, as mentioned below:

- 1) This work is extended considering the varying channel conditions of 5G mmWave and mobility, which are major challenges of TCP congestion control in 5G.
- 2) Section II is significantly extended. Section II presents an overview of different TCP algorithms. We categorize the TCP algorithms into (i) loss-based, (ii) delay-based and (iii) hybrid algorithms. Moreover, we have summarized the recent studies on the performance of TCP in mmWave enabled 5G networks.
- 3) We added a new analysis in section III. The analysis is extended in terms of throughput model, efficiency, friendliness and fairness of the proposed model.
- 4) In section IV, we had added new simulation results using the NS3-mmWave module for 5G New Radio (NR). Moreover, Table. 1, Fig. 3, Fig. 4, Fig. 5, Fig. 6, Fig. 7, Fig. 8 and Fig. 9 are new and unpublished.

The rest of the paper is organized as follows. In Section II we review the existing TCP congestion control algorithms. Section III introduces the D-TCP proposal, bandwidth estimation, congestion control factor and congestion window adjustment. The D-TCP throughput model, efficiency, friendliness and fairness are also presented in section III. Section IV demonstrates the performance evaluations based on simulation studies conducted using NS3 (mmWave 5G NR and LTE-A) followed by real time experiment results. Finally, we conclude the paper in Section V.

## II. BACKGROUND

Several studies have been carried out to improve the performance of TCP congestion control algorithms. All the proposals employ the approach of the entire usage of bandwidth to improve the goodput while minimizing the packet drop. Authors in [20] proposed an adaptive TCP congestion control algorithm CLTCP. The CLTCP uses congestion level to control the number of virtual flow numbers in a TCP connection. This could prevent the delay measurements error observed in the TCP-FIT [22]. Jingyian Wang *et al.* [23] proposed CUBIC-FIT, that simultaneously maintains the fairness among widely deployed TCP CUBIC servers and improved the performance over a large range of network conditions. The above TCP variant does not consider energy efficiency. Authors in [24] propose ecMTCP, a multipath TCP congestion control algorithm. The ecMTCP achieves load-balancing and energy-saving. The energy efficiency is achieved by

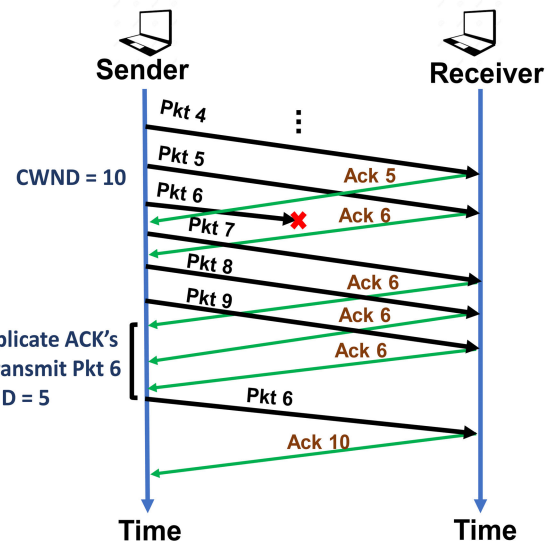


FIGURE 1. Pkt transmission, ACK receiving timeline showing 3 duplicate ACK's.

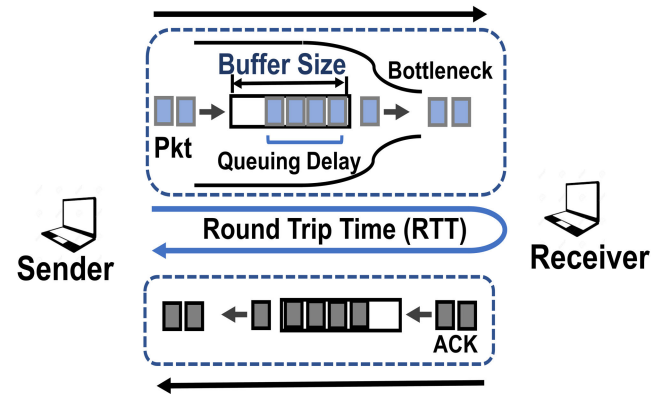


FIGURE 2. RTT between two hosts [30].

shifting the traffic from high energy cost path to the low energy cost path. Moreover, the load-balancing is attained by moving the traffic from the most congested paths to the more lightly loaded path. For improving the steaming services, authors in [25] proposed a congestion control algorithm based on MAC layer contention state. Packets of each flow are marked to estimate the contention in each state. The estimated contention is used to control the transmission at each flow.

Since the original TCP, different algorithm has been presented in the literature for estimating the congestion. These algorithms are estimating congestion based on checking the buffer (when the buffer is not empty) or calculating Round Trip Time (RTT) and detect the congestion if a dramatic increase in RTT is noticed. These algorithms are categorized into three groups: (1) Loss-based (2) Delay-based and (3) Hybrid algorithm considers both packet loss and queuing delay.

*Loss-based algorithms:* The Loss-Based algorithm considers packet loss as congestion and the packet is lost only when the network is overloaded. TCP Reno was first that widely

TABLE 1. Related works in TCP congestion control.

Algo.	Ref.	Work Summary	Objective(s)
TCP Reno	[26]	<ul style="list-style-type: none"> <li>• Congestion window increase by 1 in case of ACK</li> <li>• Reduce to half in case of 3 duplicate ACK's</li> </ul>	<ul style="list-style-type: none"> <li>• Congestion Control Algorithm</li> </ul>
TCP VenO	[2]	<ul style="list-style-type: none"> <li>• Monitors the congestion level and decide packet loss are owing to congestion or random bit error</li> <li>• Refines the multiplicative decrease and linear increase of TCP Reno</li> </ul>	<ul style="list-style-type: none"> <li>• Congestion control mechanism</li> <li>• Efficient in handling random packet loss</li> </ul>
TCP Cubic	[13]	<ul style="list-style-type: none"> <li>• Modifies linear window growth to a cubic function</li> </ul>	<ul style="list-style-type: none"> <li>• Congestion Control Algorithm</li> <li>• Equal bandwidth allocations between flows with different RTTs</li> </ul>
TCP Vegas	[31]	<ul style="list-style-type: none"> <li>• New time out mechanism operation</li> <li>• Use RTT estimate to decide for retransmission</li> <li>• Overcomes the problem of duplicate ACKs</li> </ul>	<ul style="list-style-type: none"> <li>• Proactive congestion detection</li> <li>• Better throughput</li> </ul>
FAST TCP	[34]	<ul style="list-style-type: none"> <li>• Each source tries to keep a constant number of packets in the queue</li> <li>• Packets in queues are calculated using observed RTT and base RTT</li> </ul>	<ul style="list-style-type: none"> <li>• Congestion algorithm for high-speed long-latency network</li> </ul>
CLTCP	[20]	<ul style="list-style-type: none"> <li>• Control number of virtual parallel flow using congestion level signal</li> <li>• High utilization when packet loss rate varies</li> </ul>	<ul style="list-style-type: none"> <li>• Congestion level based TCP for lossy and high-BDP networks</li> </ul>

deployed. In TCP Reno congestion window is increased on receiving of Acknowledgment (ACK). The congestion window is set to half in case of 3 duplicate ACK's [26]. Fig. 1 shows the simplified packet (Pkt) transmission, ACK's reception and 3 duplicate ACK's. In TCP Tahoe if a packet is lost, the connection is restarted and it should go through a procedure call 'slow-start'. During the slow-start the sender set the congestion window to 1. The congestion windows is increased by 1 after each ACK is received. The setting of CWND to half in TCP Reno [26] and a slow-start in TCP Tahoe [9] is a problem. TCP NewReno [11] can identify the multiple packet loss and overcome the problem of decreasing the CWND several times. The TCP CUBIC [13] is widely used and default algorithm in Linux kernel [27]. TCP CUBIC considers single CUBIC function for calculation of CWND size and enforce RTT-fairness through a congestion window function which is independent of RTT. It is less complex than High speed TCP [28] and Hamilton TCP [29] but cannot utilize 100% resources, as loss is utilized for the indication of congestion.

*Delay-based algorithms:* The delay-based algorithms are proactive and keep track of RTT. Fig. 2, shows the RTT between two hosts connected at the bottleneck [30]. An unaffected RTT indicates that there is no congestion. In contrast, an increase in RTT or packet loss indicates congestion. The RTT provides more timely feedback and the degree of network congestion. It is important for a large BDP network [30]. Moreover, the delay based algorithms are also efficient in a network environment where packet drop is high. When a packet drop occurs irrelevant to congestion the RTT time remains steady and packet delivery time remains the same. As a result, the delay-based algorithm has the capability to control the random packet loss and may perform better in

a lossy network. The delay-based congestion control algorithms can control the packet sending rate before the queue delay becomes high or packet loss occurs [30].

TCP dual was the first algorithm, which identifies the congestion using queuing delay. It calculates maximum, minimum RTT and uses a fraction of maximum RTT as a threshold for congestion detection. The problem with TCP dual is, in case of multiple flows the RTT time is overestimated and available resources are not efficiently utilized due to conservative nature [27]. The first improvement to this algorithm was TCP Vegas [31]. TCP Vegas uses CWND, RTT and  $RTT_{base}$  (derived from the minimum witnessed RTT) to calculate the number of in-flight bytes, that inhabit inside the bottleneck buffer and try to maintain this number smaller. Later studies [32], [33] demonstrated issues with TCP Vegas: (i) low fair share of bandwidth with loss based congestion control algorithms and (ii) low throughput based flow with sudden increase in RTT. Authors in [34] propose FAST TCP to utilize available resources in high-speed networks. The congestion window in Fast TCP algorithm is updated periodically after every  $x$  ms. TCP Fast include a parameter  $\zeta$  in Vegas congestion windows function to show the compromise between stability and high throughput. The throughput is high in case of high  $\zeta$  and behaves as Vegas for a low value of  $\zeta$ . Authors in [35] propose an extension of FAST, VFAST to minimize the oscillations in the queue and throughput. Elastic-TCP [36] is another algorithm in the literature that estimates the bottleneck bandwidth utilization with the help of RTT to achieve high average throughput.

*Hybrid algorithms:* The hybrid algorithms identify congestion using loss and delay both. The hybrid algorithm are designed to detect the congestion in loss based network well before queues is full while keeping throughput high in



presence of other variances. These algorithms are useful in high speed network, where resource utilization is very low and a technique with fast CWND value is required. TCP Veno proposed by Cheng Peng Fu *et al.* [2] was the first hybrid congestion control algorithm. TCP Veno detects the congestion or non-congestion state by integrating the congestion detection technique of Vegas TCP into Reno TCP and adjust the congestion window accordingly. Moreover, it extends an additive increase phase to exploit the available bandwidth and reduce the losses due to congestion.

Africa [37], Compound [38], YeAH [39] made similar modification on Vegas to identify the congestion and make efficient use of available bandwidth in a broadband network. Illinois [40], TCP-Adaptive Reno (AReno) [41] and Libra [42] modified the congestion window to follow both bottleneck link capacity and RTT. In 2016, Google developed a Bottleneck Bandwidth and Round Trip Time (BBR) congestion control algorithm [16]. BBR aims to improve TCP throughput and resolve the bufferbloat problem. BBR controls the congestion by sending the packets according to estimated bottleneck bandwidth and  $RTT_{min}$ . To align with varying network conditions and to examine the available bandwidth, BBR increases the packet sending rate periodically and inflight size to 125% for an RTT period. If there is no change in bandwidth the packet sending rate and inflight size are reduced to 75%. This process carried out every eight cycle for each lasing RTT [16], [27], [30]. BBR maintains one BDP for packets are inflight and guarantees the full utilization of resources with minimum queuing delay. The problem with BBR is the wrong estimation of  $RTT_{min}$  can affect the operation. Moreover, contingent with new flow time and RTT bandwidth can be shared unfairly. It is not clear how well this algorithm works with other congestion algorithms in the presence of long standing queues.

Besides, there exist several TCP studies related to mmWave communications. Authors in [4] demonstrated the performance of TCP and how high link variability of mmWave impacts the end-to-end performance. Zhang *et al.* [19] presented a comprehensive simulation study of TCP in mmWave cellular system, considering various factors and identified the open challenges in this area. Work in [17] presented the impact of the mmWave links on the performance of the transport layer and points out the challenges for reducing latency while maintaining high throughput. Authors in [43] proposed a novel MPTCP congestion control scheme named delay-equalized FAST (DEFT) to achieve high throughput and low end-to-end (E2E) delay in 5G networks. In Table 1, we presented the comparison of the different TCP algorithms. The above mentioned TCP algorithm does not consider the specious wireless losses while adjusting the congestion window and does not exploit the available bandwidth. This results in a reduction in goodput despite of the enough accessible bandwidth. This motivates us to propose a dynamic approach to control the congestion. We (a) utilize the currently available bandwidth for calculation of congestion control factor; (b) adjust the

congestion window adaptively; (c) demonstrate the goodput gain and intra/inter algorithm fairness.

### III. PROPOSED APPROACH: D-TCP

In this section, we first present the D-TCP proposal. We then estimate the bandwidth for D-TCP and calculate the congestion control factor ( $N$ ). We utilize the congestion control factor to adjust the CWND and presents the D-TCP throughput analysis. We prove the efficiency of D-TCP in terms of fairness, friendliness and network utilization. Table 2 lists the major notations used in this paper.

TABLE 2. List of symbols used.

Sym.	Description	Sym.	Description
$t_i$	Time instant	$d_i$	Data transferred
$\frac{1}{\tau}$	Cutoff frequency	$B_I$	Filtered available bandwidth
$W$	TCP window size	$g$	Clock granularity
$N$	Congestion control factor	$B_i$	Bandwidth at time $t_i$
$Q_i$	Queue length	$\eta$	Convergence factor
$P_i$	Packet received	$X_i$	Packet round
$p_l$	Packet loss due to congestion	$x_l$	Link Congestion Matrices
$R$	Routing Matrix	$q_l$	Queuing delay
$TP$	Throughput	$T_s$	Throughput for a session
$D_s$	end-to-end RTT	$P_s$	Random packet loss
$D_l$	RTT of link $l$	$\gamma_{DTCP}$	DTCP fairness
$m_l$	Price mapping function	$D_{DTCP}$	DTCP Delay

#### A. D-TCP PROPOSAL

In this article, we propose a TCP enhancement (D-TCP) for congestion control in next generation mmWave 5G network. D-TCP learn the available Bandwidth (BW) based on channel conditions. It is important to learn bandwidth as mmWave enabled wireless links suffer with excessive path and penetration losses [1], which result in high variations in Reference Signal Received Power (RSRP). In case when there is no line of sight path between transmitter and receiver the channel losses are masked with the help of transmission and TCP algorithm is unaware of such variations. We used discrete time filter to approximate the available bandwidth based on traffic intensity on link and channel conditions. After approximating the bandwidth, we calculate the congestion control factor ( $N$ ) which is used to control the congestion window. The congestion factor ( $N$ ) is calculated using window size, estimated bandwidth and queue length at the  $i^{th}$  period. The  $N$  is used to adjust the congestion window is Adaptive Increase/Adaptive Decrease (AIAD) paradigm, in place of traditional approach additive increase/multiplicative decrease. The congestion window is increased by  $N_i$  on the successful delivery of packet (received ACK) and congestion window is decreased by  $(1 - y)$  in case of packet loss (where  $y$  is given in section III-D). If a packet drop is detected, the D-TCP algorithm tries to take the congestion window to the previous level, with the help of calculated bandwidth. This results in better utilization of network resources particularly in case of high Bandwidth-delay (high-BDP) and lossy network conditions.

## B. BANDWIDTH ESTIMATION ( $BW_E$ )

Bandwidth estimation is a key attribute for congestion control algorithm design. The algorithms like TCP Westwood, TCP BBR estimate the bandwidth to control the CWND. However, they do not consider the varying 5G network conditions. D-TCP not only estimates the bandwidth using the packet flow but also adapts to the varying mmWave characteristics considering SINR fluctuations.

Let  $d_i$  be the data transferred during time period  $t_{i-1}$  and  $t_i$ . Then the Bandwidth  $B_i$  at time instance  $i$  is defined as the ration of amount data transferred in a given time interval and it is given as

$$B_i = \frac{d_i}{t_i - t_{i-1}} = \frac{d_i}{\Delta_i} \quad (1)$$

where  $\Delta_i$  is time difference between  $t_i - t_{i-1}$ . We model the predicted available bandwidth as a discrete time linear system and employ a discrete time filter. Using the Tustin approximation [15] the filtered available bandwidth  $B_I$  at interval  $\Delta_i$  is given as

$$B_I = \left[ \frac{\frac{2\tau}{\Delta_i} - 1}{\frac{2\tau}{\Delta_i} + 1} \times (B_{i-1}) \right] + \left[ \frac{B_i + B_{i-1}}{\frac{2\tau}{\Delta_i} + 1} \right] \quad (2)$$

where  $\frac{1}{\tau}$  is the cut off frequency and  $B_{i-1}$  is bandwidth at previous time instant. Let consider  $\kappa_i$  as a constant and it is given as

$$\kappa_i = \frac{2\tau - \Delta_i}{2\tau + \Delta_i} \quad (3)$$

Hence, by utilizing the  $\kappa_i$ , equation (2) can be simplified as

$$B_I = \kappa_i B_{i-1} + (1 - \kappa_i) \left( \frac{B_i + B_{i-1}}{2} \right) \quad (4)$$

The constant  $\kappa_i$  has high impact on filtered available bandwidth  $B_I$ . The higher value of  $\kappa_i$  imply that more weightage to previous learning and lower value of  $\kappa_i$  imply more weightage to current learning. As, we want to utilize the previous learning, so we considered  $\kappa_i$  as 0.90. Hence equation (4) can be rewritten as

$$B_I = 0.90 B_{i-1} + 0.10 B_i \quad (5)$$

As we are considering discrete time filter, packet aliasing may occur. To remove the packet aliasing effect, we normalize  $B_I$  using a low pass filter and obtained estimated bandwidth  $BW_E$ . According to Shannon Equation, the channel capacity is directly related to the Signal to Interference Noise Ratio (SINR). We estimate the variation in the SINR and feedback the difference in the normalization factor. The SINR information can be exposed to TCP layer through a cross-layer implementation (Section IV-C). In mmWave NR, the available bandwidth oscillates, as channel conditions are highly variable (transition between LOS/NLOS). Let,  $SINR_i$  and  $SINR_A$  is the current SINR and average SINR respectively. Hence, we normalize the fluctuations in the filtered

bandwidth as follows:

$$norm = (B_i - B_I) + (SINR_i - SINR_A) \quad (6)$$

$$BW_E = B_I + g \times norm \quad (7)$$

where  $norm$  is an error normalization parameter to cope up the ad-hoc network conditions and  $g$  is clock granularity. The clock granularity ( $g$ ) is set as 0.5 to match with the epoch timer. In the next subsection we calculate the congestion control factor  $N$ .

## C. CALCULATION OF CONGESTION CONTROL FACTOR ( $N$ )

To compute the Congestion Control Factor ( $N$ ), first we have to calculate the expected queue length. The queue length of a D-TCP flow is proportional to TCP window size  $W$ . Let,  $E[W_{SF}]$  is the expected window size of the single TCP flow, the expected queue length  $E(Q)$  is computed as:

$$E(Q) = \alpha \times E[W_{SF}] \quad (8)$$

where  $\alpha$  is TCP fairness factor explained in Section III-F3. As we are creating  $N$  different flow in D-TCP, the  $E[W_{SF}]$  for  $N$  different flows is given as  $\frac{E[W]}{E[N]}$ , where  $E[W]$  is expected window size and  $E[N]$  is expectation of the congestion control factor of  $N$  flows. By substituting  $E[W_{SF}]$  in equation (8), we have

$$E(Q) = \alpha \times \frac{E[W]}{E[N]} \quad (9)$$

Next we update the parameter  $N_{i+1}$ . Let  $N_i$ ,  $Q_i$  and  $W_i$  are  $N$ , queue length and window size of the  $i^{th}$  period, respectively, then using  $E(Q)$ , equation (9) during  $(i+1)^{th}$  time, the parameter  $N_{i+1}$  updated as

$$N_{i+1} \leftarrow \max \left( 1, N_i + 1 - \frac{Q_i}{\alpha \times W_i} N_i \right) \quad (10)$$

Let  $BW_C$  is the current bandwidth and by using estimated bandwidth  $BW_E$ ,  $Q_i$  can be calculated as:

$$Q_i = (BW_E - BW_C) \times \frac{W_i}{BW_E} \quad (11)$$

By combining equations (10) and (11), the updated congestion control factor ( $N$ ) is given as:

$$N_{i+1} \leftarrow \max \left( 1, N_i + 1 - \frac{BW_E - BW_C}{\alpha \times BW_E} W_i \right) \quad (12)$$

According to equation (12), the expectation of the  $N$  can be written as

$$E[N] = \max \left( 1, \frac{\alpha \cdot E[BW_E]}{\delta B} \right) \quad (13)$$

where  $\delta B = E[BW_E] - E[BW_C]$  is the over/under-utilized bandwidth estimation.

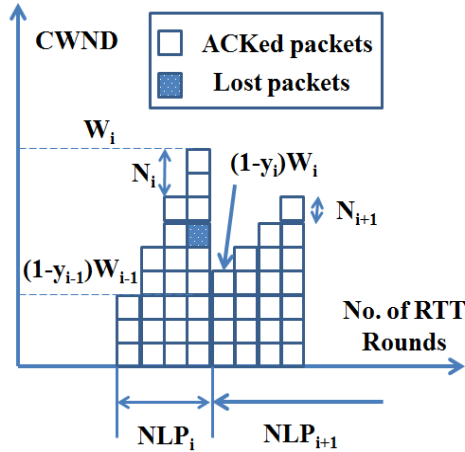


FIGURE 3. Congestion window growth in No Loss Period (NLP).

**D. CWND ADJUSTMENT**

In D-TCP algorithm, congestion window is adjusted using congestion control factor ( $N$ ). The congestion window is adjusted in adaptive increase or decrease paradigm depending on ACKed packets or loss packets. On each RTT, the congestion window is increased by  $N_i$  and in case of packet drop the congestion window is decrease by  $(1 - y)$ . The congestion window adjustment is given by:

$$\text{For each RTT: } CWND \leftarrow CWND + N_{i+1} \tag{14}$$

$$\text{For each Loss: } CWND \leftarrow CWND(1 - y) \tag{15}$$

$$y = \frac{\eta}{[(2\eta - 1)N] + 1} \tag{16}$$

where  $y$  is given by equation (16) and  $\eta$  is convergence factor for fairness. The convergence factor is set as  $2$  ( $\eta = 2$ ) in our trails to prevent the aggressiveness (explained in Section III-E). By substituting  $\eta = 2$ ,  $y = \frac{2}{3N+1}$ . For TCP Reno  $N = 1$ , hence  $y = \frac{1}{2}$ . The congestion window  $W$  is adjusted in adaptive increase adaptive decrease manner, as shown in Fig. 3. If two packets drop consecutively, we consider this period as No Loss Period (NLP). The first packet loss has happened (filled box) as delineated in the Fig. 3, the congestion window  $W$  is adjusted using equations (14) ~ (16). The overall D-TCP algorithm is delineated in Algorithm. 1.

**E. D-TCP THROUGHPUT MODEL**

In this subsection, we derive an approximate throughput model for D-TCP. We adopt most of our terminology from J. Padhye et al. [46] and assume that readers are familiar with TCP Reno Congestion control algorithm. As depicted in Fig. 3, the congestion window  $W$  is on Adaptive Increase or Adaptive Decrease paradigm. During time  $T_i$ ,  $P_i$  packets is being transferred for the  $i^{th}$  iteration. Hence, expected TCP throughput ( $TP$ ) during steady state can be written as

$$TP = \frac{E[P]}{E[T]} \tag{17}$$

where  $E[P]$  is expected packet transmitted and  $E[T]$  is expected packet transmission time. Since, in D-TCP algorithm during  $NLP_i$  for each RTT the congestion window  $W$  is increase by  $N_i$  and reduced by  $(1 - y)$  for each loss. Hence, the Packet received ( $P_i$ ) at the  $i^{th}$  duration can be computed as

$$\begin{aligned} P_i &= \sum_{K=0}^{X_{i-1}} N_i.K + (1 - y_{i-1}).W_{i-1}.X_i \\ &= \frac{X_i.(X_{i-1})}{2} * N_i + (1 - y_{i-1}).W_{i-1}.X_i \end{aligned} \tag{18}$$

And the congestion window ( $W_i$ ) at the end of  $NLP_i$  is calculate as

$$W_i = (1 - y_{i-1}).W_{i-1} + N_i.X_i \tag{19}$$

where  $X_i$  is the packet round. There would be multiple rounds of data transfer. We consider at round  $X_i$ , where packet loss happens. By substituting  $W_i$  equation (19) in equation (18), we have

$$P_i = X_i \left( \frac{(1 - y_{i-1}).W_{i-1} + W_i - N_i}{2} \right) \tag{20}$$

The packet expectation during the transmission is inversely proportional to the expected packet loss probability  $\rho$ , hence

**Algorithm 1** D-TCP Algorithm

- 1: **Initialization:**
- 2:  $\Upsilon \leftarrow$  Initial congestion window, init\_cwnd
- 3:  $CWND \leftarrow 2 \times \Upsilon$ , ssthresh  $\leftarrow$  65535 (slow start threshold)
- 4: D-TCP\_reset()
- 5: **Packet Loss:**
- 6:  $CWND \leftarrow CWND - CWND \times \frac{\eta}{[(2\eta - 1)N] + 1}$
- 7: ssthresh  $\leftarrow$  CWND
- 8: **D-TCP\_reset():**
- 9:  $N \leftarrow 1$ ,  $\alpha = 0.2$
- 10: epoch\_start  $\leftarrow$  tcp\_time\_stamp()
- 11: **Timeout:**
- 12:  $CWND \leftarrow 2 \times \Upsilon$
- 13: D-TCP\_reset()
- 14: **N\_update():**
- 15: **if** tcp\_time\_stamp() - epoch\_start > update\_epoch **then**
- 16: epoch\_start  $\leftarrow$  time\_stamp
- 17: **if** current BW < estimated BW **then**
- 18: temp  $\leftarrow$  max(1.0, preNValue)
- 19: nValue  $\leftarrow$  nValue + temp
- 20: preNValue  $\leftarrow$  nValue
- 21: **end if**
- 22: **else if** current BW > estimated BW **then**
- 23: nValue  $\leftarrow$  nValue + 1
- 24: preNValue  $\leftarrow$  nValue
- 25: **end if**

$E[P] = \frac{1}{\rho}$ . Using equation (20),  $\frac{1}{\rho}$  is given as

$$\frac{1}{\rho} = E[X] \left( \frac{(2 - E[y]).E[W] - E[N]}{2} \right) \quad (21)$$

where  $E[X]$  and  $E[y]$  is expectation of  $X_i$  and  $y_i$  respectively. Using equation (19), the expectation  $E[X]$  in round  $X_i$  can be given as

$$E[X] = \frac{E[y].E[W]}{E[N]} \quad (22)$$

Solving equations (21) and (22) for  $E[X]$

$$E[X] = \frac{E[y]}{2.(2 - E[y])} + \sqrt{\left( \frac{E[y]}{2.(2 - E[y])} \right)^2 + \frac{2.E[y]}{\rho.E[N](2 - E[y])}} \quad (23)$$

Assuming  $\rho$  to be very small, we can write equation (23) as

$$E[X] = \sqrt{\frac{2.E[y]}{\rho.E[N].(2 - E[y])}} + O\left(\frac{1}{\sqrt{\rho}}\right) \quad (24)$$

Assuming  $r^{i,j}$  as a randomly distributed value of  $j^{th}$  iteration of NLP, with  $P_i = \sum_{j=1}^{X_i} r^{i,j}$ . Following the assumption in [46], the value of  $r^{i,j}$  are independent of  $j$  and congestion window. Therefore, we can write  $E[T]$  as

$$E[T] = E[X].E[r] = E[X].RTT \quad (25)$$

By substituting equations (24) and (25) in equation (17), we derive throughput (TP) as

$$TP = \frac{1}{\rho.E[T]} = \frac{1}{\rho.E[X].RTT} = \frac{1}{\rho.RTT \left( \sqrt{\frac{E[y]}{\rho.E[N].(2 - E[y])}} + o\left(\frac{1}{\sqrt{\rho}}\right) \right)} \quad (26)$$

For D-TCP, when  $\eta$  is set to 2, then  $E[y] = \frac{2}{3E[N]-1}$ . Therefore, using equation (26) the approximate throughput model for D-TCP ( $TP_{DTCP}$ ) can be written as

$$TP_{DTCP} = \frac{E[N]}{RTT} \sqrt{\frac{3}{2.\rho}} \quad (27)$$

For TCP Reno,  $E[N] = 1$  as  $N$  is always 1. The throughput model for TCP Reno can be written as

$$TP_{RENO} = \frac{1}{RTT} \sqrt{\frac{3}{2.\rho}} \quad (28)$$

Hence, when  $\eta$  is selected as 2, then D-TCP flows is  $E[N]$  times TCP RENO flows. For multiple flows using different TCP sockets (multcp), the throughput model can be derived as

$$TP_{multcp} = \frac{1}{RTT} \sqrt{\frac{E[N](4E[N] - 1)}{2.\rho}} \quad (29)$$

which is not exactly  $E[N]$  times TCP Reno. Hence, D-TCP AIAD algorithm is stable when  $\eta$  is set to 2. By substituting

equation (13) into equation (27), we can write throughput of D-TCP as

$$TP_{DTCP} = \max \left( \frac{1}{RTT}, \frac{\alpha}{\delta B} \right) \sqrt{\frac{3}{2.\rho}} \quad (30)$$

where  $\delta B = E[BW_E] - E[BW_C]$  is the over/under-utilized bandwidth estimation.

### F. D-TCP EFFICIENCY

In this section, we prove the efficiency of D-TCP in terms of resource utilization, friendliness and fairness.

#### 1) EQUILIBRIUM AND STABILITY

We study a network model which consists of  $L$  links, where  $l \in (1, 2, ..L)$  each with bandwidth  $B_l$ . There are  $S$  TCP session flows in this link. The routing matrix for  $L \times S$  is denoted as  $R$ . For each TCP session  $s \in (1, 2, ..S)$ ,

$$R_{l,s} = \begin{cases} 1 & \text{if session } s \text{ traverse in link } l \\ 0 & \text{otherwise} \end{cases}$$

The link  $l$  has RTT  $D_l$  and packet loss rate  $P_l$  which is independent of the congestion and depends on the link  $l$ . Each link has congestion metrics  $x_l$  which is measured using the packet loss due to congestion  $p_l$  and queuing delay  $q_l$ . Hence  $x_l = m_l(p_l, q_l)$ , where  $m_l$  is price mapping function. The throughput for a session ( $T_s$ ) can be given as

$$T_s = f_s(x) \quad (31)$$

According to [47] the network reaches equilibrium when

$$RT \leq B, X(RT - B) = 0 \quad (32)$$

*Theorem 1:* The equilibrium defined in [48] exists if the network meets the following three conditions:

- $T_s = f_s(x_l, x_{-l})$  is a non-increasing function of  $x_l$  for any fixed  $x_{-l} = (x_1, \dots, x_{l-1}, x_{l+1}, \dots, x_L)^T$ ;
- $\forall \epsilon > 0$ , there exists a  $x_{max}$  such that if  $x_l > x_{max}$  for link  $l$ , then  $T_s(x_l) < \epsilon$ ,  $\forall s$  where  $R_{l,s} = 1$
- if  $(RT - B) = 0$  then D-TCP flows can reach equilibrium.

*Proof:* The packet loss and queuing delay due to congestion  $p_l$  and  $q_l$  increase as the network gets more congested. Hence we can make the following assumptions:

- 1) If there is no congestion then  $p_l = q_l = 0$ . Hence  $m(0, 0) = 0$
- 2)  $x_l = m_l(p_l, q_l)$  is a strictly increasing function of  $p_l$  and  $q_l$
- 3) For each link  $l$ ,  $p_l$  is non-decreasing as  $q_l$  increases and vice versa.

We define packet loss  $p_s$  and delay  $q_s$ , due to congestion in a particular TCP session  $s$  as

$$p_s = 1 - \prod_{l \in L} (1 - R_{l,s} p_l), \quad q_s = \sum_{l \in L} R_{l,s} q_l$$



TABLE 3. Throughput comparison of various TCP model ([22]).

TCP Variants	Throughput Models	Upper Bounds	Fairness Index ( $\gamma = T_i(D_i)/T_j(D_j)$ )
D-TCP	$T = \max\left(\frac{1}{RTT}, \frac{\alpha}{\delta B}\right) \sqrt{\frac{3}{2P}}$	$\times$	$\gamma = 1$ (when $E[N] > 1$ )
Reno	$T = \frac{1}{D+q} \sqrt{\frac{3}{2(P+p)}}$ [46]	$T \leq \frac{1}{D} \sqrt{\frac{3}{2p}}$	$\gamma = \frac{D_j+q}{D_i+q}$
CTCP	$T = \frac{1}{D+q} \frac{A}{(P+p)^{\frac{1}{2-k}}}, \Lambda = \frac{(1-(1-\beta')^{2-k})^{\frac{1-k}{2-k}}}{\alpha'^{\frac{1}{2-k}} (1-(1-\beta')^{1-k})}$ where, $\alpha', \beta'$ and $k$ are preset parameters. [38]	$T \leq \frac{1}{D} \frac{\Lambda}{p^{\frac{1}{2-k}}}$	$\gamma \leq \left[\frac{D_j+q}{D_i+q}\right]^2$ [38]
Veno	$T = \frac{1}{D+q} \sqrt{\frac{1+\gamma'}{2(1-\gamma')(P+p)}}$ , where $1/2 \leq \gamma' \leq \frac{4}{5}$ . [49]	$\frac{T}{\frac{1}{D} \sqrt{\frac{1+\gamma'}{2(1-\gamma')P}}} \leq$	$\gamma = \frac{(D_j+q)\sqrt{(1+\gamma'_j)(2(1-\gamma'_j)(P+p))}}{(D_i+q)\sqrt{(1+\gamma'_i)(2(1-\gamma'_i)(P+p))}}$
Westwood	$T = \frac{1}{\sqrt{(D+q)((P+p)(D+q))}} \sqrt{\frac{(1-(P+p))}{P+p}}$ [50]	$\frac{T}{\frac{1}{D} \sqrt{(1-P)}} \leq$	$\gamma = \sqrt{\frac{(D_j+q)((P+p)D_j+q)}{(D_i+q)((P+p)D_i+q)}}$
Vegas/FAST	$T = \alpha''/q$ , where $\alpha''$ is preset parameter. [34], [51]	$\times$	$\gamma = 1$

The end-to-end RTT ( $D_s$ ) and random packet loss ( $P_s$ ) of session  $s$  are

$$P_s = 1 - \prod_{l \in L} (1 - R_{l,s} P_l), \quad D_s = \sum_{l \in L} R_{l,s} D_l$$

We can express the throughput function in equation (31) for D-TCP as

$$T_s = f_s(x) = f_s(x_1, \dots, x_l) = f_s(m_l(p_1, q_1), \dots, m_l(p_1, q_1)) \\ = \max\left(\frac{1}{D_s + q_s}, \frac{\alpha}{\delta B \cdot q_s}\right) \sqrt{\frac{3}{2 \cdot (P_s + p_s)}} \quad (33)$$

According to the assumption (1) and (2) and equation (25), we know that,

- If  $R_{l,s} = 1$ , then  $T_s$  is a strictly decreasing function of  $x_l$
- If  $R_{l,s} = 0$ , then  $T_s$  is a constant
- Letting  $T_s = \epsilon$ ,  $x_{max} = f_s^{-1}(T_s)$  and  $x_l > x_{max}$  implies  $x_s(p_l) < x_s = \epsilon$

Therefore,  $T_s$  is a non-increasing function and the first condition holds. To prove the third condition, lets assume that equilibrium is at  $X = 0$ , then  $p_s = q_s = 0 \forall$  session in  $S$ . From the throughput equation (33) of D-TCP, we have  $x_s \rightarrow \infty$ . This contradicts the first proof that  $RT \leq B$ . Therefore, D-TCP satisfy  $(RT - B) = 0$ . Hence, network with D-TCP can reach equilibrium.

In Table. 3 we have listed the throughput model of the popular TCP congestion control algorithm using [22], [34], [38], [46], [49]–[51].

## 2) D-TCP FRIENDLINESS

A TCP congestion control algorithm is TCP friendly if it operates fair with existing TCP algorithms. To analyze the friendliness of D-TCP we consider a scenario where  $S$  TCP Reno devices share same bottleneck link. Let  $\hat{T}_R$  be the average throughput of  $S$  TCP Reno flows and we replace  $K$  of the  $S$  Reno flows with D-TCP flows. Let  $\hat{T}_D$  be average throughput of rest  $M$  flows given by  $M = S - K$  and the

throughput difference of the TCP Reno flows and D-TCP flows can be given as

$$\hat{T}_\delta = \hat{T}_R - \hat{T}_D \quad (34)$$

If  $\hat{T}_\delta$  is close to 0, then D-TCP is said to be Reno friendly.

**Theorem 2:** D-TCP is TCP Reno friendly i.e.  $\hat{T}_\delta = 0$  if

$$\alpha \leq \frac{q_r}{D_{DTCP} + q_r}$$

where  $q_r$  is queuing delay for TCP Reno and  $D_{DTCP}$  stands for D-TCP delay.

**Proof:** We can write  $\hat{T}_R$  as

$$\hat{T}_R = \frac{1}{D_{DTCP} + q_r} \sqrt{\frac{3}{2(P+p)}} \quad (35)$$

And from equation (33), the throughput of the D-TCP can be written as,

$$\hat{T}_D = \max\left(\frac{1}{D_{DTCP} + q_d}, \frac{\alpha}{\delta B \cdot q_d}\right) \sqrt{\frac{3}{2 \cdot (P + p_d)}} \quad (36)$$

where  $q_d$  and  $p_d$  is the queuing delay and packet loss respectively, when we replace  $K$  TCP Reno connection with our proposed D-TCP congestion algorithm. Solving equations (35) and (36), we obtain that when  $\alpha \leq \frac{q_r}{D_{DTCP} + q_r}$  then  $\hat{T}_R = \hat{T}_D$ . Hence  $\hat{T}_\delta = \hat{T}_R - \hat{T}_D = 0$ . In our experiments we select  $\alpha$  in such a way that it is TCP friendly ( $\alpha \in \{0.1, \dots, 0.5\}$ ).

## 3) D-TCP FAIRNESS

According to [52], the TCP algorithm is fair if throughput remains same for different RTTs, with similar packet loss rate.

**Theorem 3:** If two flows,  $i$  and  $j$ , shares the same bottleneck link, then D-TCP maintains intra fairness if the throughput achieved by each flows is equal i.e.  $TP_i = TP_j$

**Proof:** According to the throughput model of D-TCP equation (30) when  $E[N] > 1$ ,  $TP_{DTCP} = \left(\frac{\alpha}{\delta B}\right) \sqrt{\frac{3}{2 \cdot p}}$ . In our

TABLE 4. Simulation parameters.

Parameter	Value
mmWave TX power [43]	30 dBm
mmWave carrier frequency [43]	28 GHz
mmWave BW [43]	1 GHz
Packet size	1400 Bytes
Channel Model [44] [45]	Samsung 5G field test channel model
RLC AM buffer size [7]	10 MB
Core network latency [7]	1 ms
Remote host latency [7]	10 ms
UE Speed	1.5 m/s

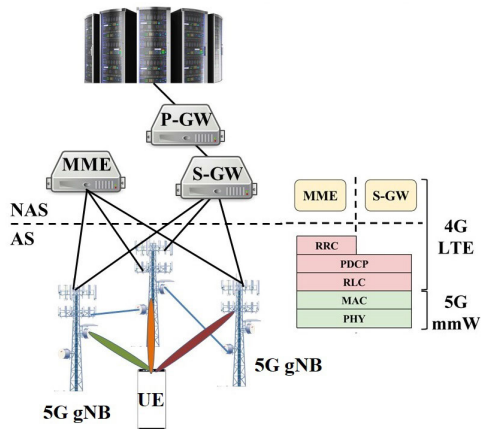


FIGURE 4. 5G NR simulation setup (mmWave-NS3 model).

experiments  $\alpha$  is set as the same value for all the flows. Setting an higher value might impact the friendliness as depicted in equation (36). Therefore, we set 0.2 in our experiment to maintain fairness and friendliness. The ratio of the two flows is written as

$$\gamma_{DTCP} = \frac{TP_i}{TP_j} = \frac{q_i}{q_j} \sqrt{\frac{P + p_j}{P + p_i}} \quad (37)$$

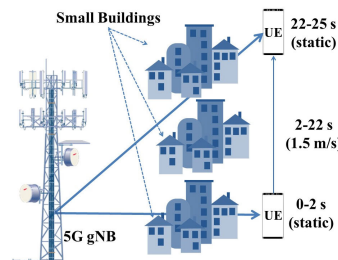
As  $i$  and  $j$  shares the same bottleneck link,  $p_i = p_j$  and  $q_i = q_j$ . Therefore,  $TP_i = TP_j$  and  $\gamma_{DTCP} = \frac{TP_i}{TP_j} = 1$ . Hence, D-TCP maintains intra fairness.

#### IV. PERFORMANCE EVALUATION

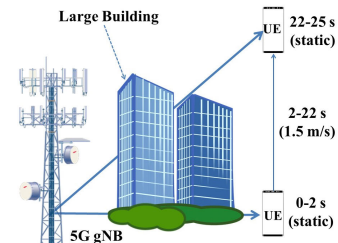
In this section, we evaluate the performance of our proposed model. We analyze the performance of D-TCP algorithm in 5G NR and LTE-A using NS3-mmWave and NS-3 LTE-A simulator, respectively. The simulation results are compared with different TCP algorithms. In last the D-TCP is evaluated in real time experiments.

##### A. 5G NR BASED SIMULATION RESULTS

The performance evaluation of D-TCP are carried out using the mmWave NS-3 module, as explained in [53]. We considered mmWave carrier frequency of 28 GHz, Tx power of 30 dBm and mmWave bandwidth of 1 GHz. Table 4 outlines a list of parameters with values, including the buffer size of 10 MB, the packet size of 1400 bytes, core network latency



(a) NS3 experiment scenario 1



(b) NS3 experiment scenario 2

FIGURE 5. 5G NR simulation scenario.

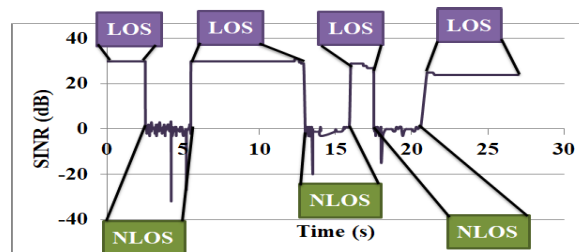


FIGURE 6. SINR with varying time for scenario 1.

of 1 ms and remote host latency of 10 ms. Fig. 4 provides an overview of the simulation setup. The simulations have been performed in the two scenarios where UE moves with a speed of 1.5 m/s and downloading a D-TCP flow with source rate up-to 2Gbps. We considered small buildings scenario 1 shown in Fig. 5 (a) and large buildings in experiment scenario 2 shown in Fig. 5 (b).

Fig. 6 depicts the SINR with varying time in scenario 1. The SINR varies as UE toggles between LOS and NLOS. In the case of LOS, the SINR is up to 40 dB as a direct path exists between gNB and UE. The SINR is less than 0 when UE is in NLOS as UE receives a diffracted signal. Fig. 7 presents the throughput for D-TCP, TCP BBR, TCP BIC, TCP Cubic and TCP NewReno for scenario 1. From Fig. 7, we can infer that, during idle conditions, the congestion control variants perform the same. However, when the UE moves from NLOS to LOS, our proposed D-TCP algorithm aware of the network conditions and immediately restores the full bandwidth. As observed from Fig. 7, TCP BBR and TCP BIC achieve the maximum throughput in 2 s and 4 s respectively and they are fastest in reaching the maximum throughput values among the state-of-the-art TCP algorithms.

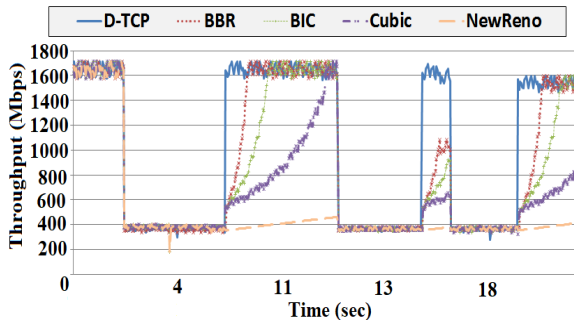


FIGURE 7. Throughput in scenario 1 for different algorithms.

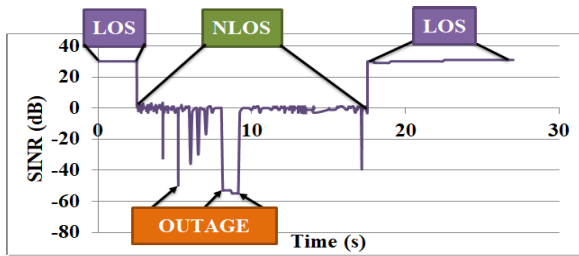


FIGURE 8. SINR with varying time for scenario 2.

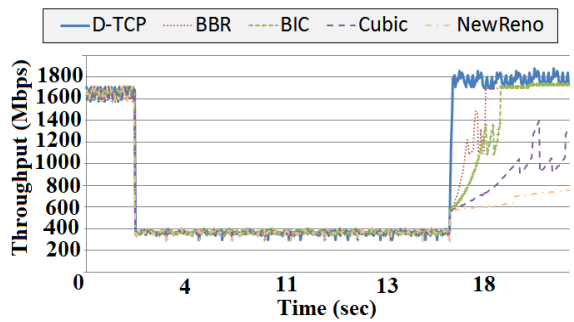


FIGURE 9. Throughput in scenario 2 for different algorithms.

The other variants probe for a longer duration to reach the maximum throughput.

Fig. 8 and Fig. 9 delineate the SINR with varying time and throughput for scenario 2 respectively. It is observed from Fig. 8, when SINR is less than  $-50$  dBm, UE is in outage as large building blocks the direct path. As depicted in Fig. 9, when the NLOS is for a longer duration, the problem with state-of-the-art TCP algorithms are still worse. In the case of D-TCP, thanks to the information on the fluctuation in the SINR, that immediately restores to fully utilize the resources. It is also noticed from Fig. 9 that there is a small outage duration in the UE path. This outage duration does not affect the congestion window growth, as lower layer retransmission mechanisms mask the channel conditions. We also performed high-speed 5G test and observed from Fig. 10, D-TCP, TCP-BBR and BIC copes up better to the high speed network conditions. Cubic and NewReno probe for longer time to reach maximum throughput.

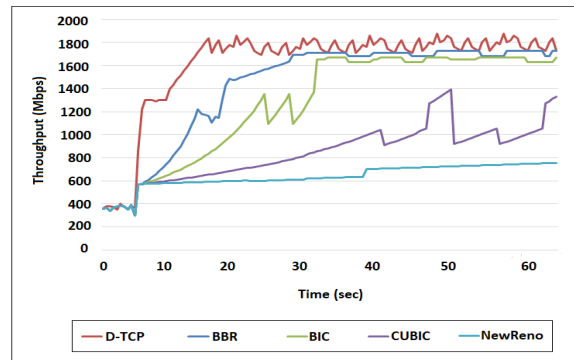


FIGURE 10. Throughput in high-speed scenario for different algorithms.

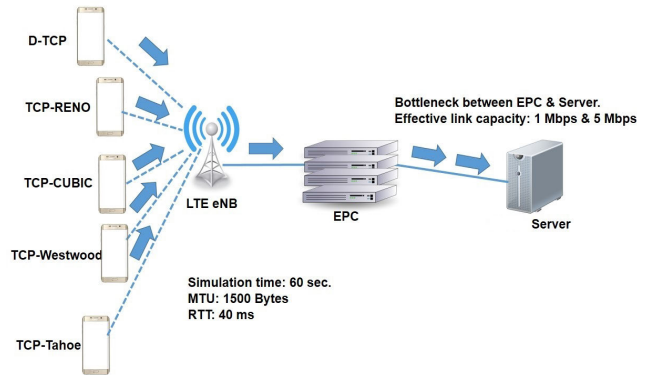


FIGURE 11. Simulation setup for LTE-A based NS3 model.

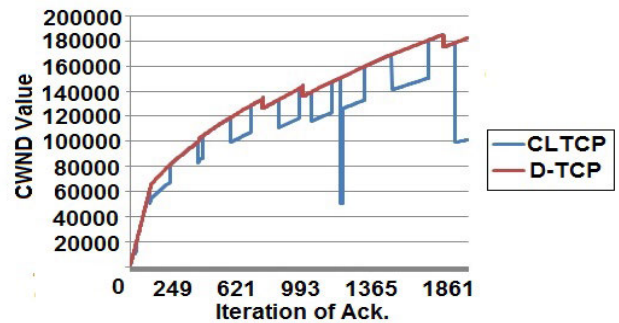


FIGURE 12. CWND growth comparison.

### B. LTE-A BASED SIMULATION RESULTS

To evaluate the efficiency of D-TCP in LTE-A, NS3 based simulations are carried out where the device is capable of running the proposed algorithm. An overview of the experimental setup is shown in Fig. 11. The UE is connected to the server through eNB and EPC. We considered MTU of 1500 bytes,  $RTT = 40$  ms, effective P2P bottleneck link capacity of 1 Mbps & 5 Mbps and total simulation time is 60 sec. We considered the performance parameters: CWND growth, efficiency in lossy, lossless & ideal state network and intra/inter algorithm fairness to analyze the accuracy of D-TCP. The TCP BBR is supported by Linux since kernel version 4.9 and most of the LTE Smartphones does not have

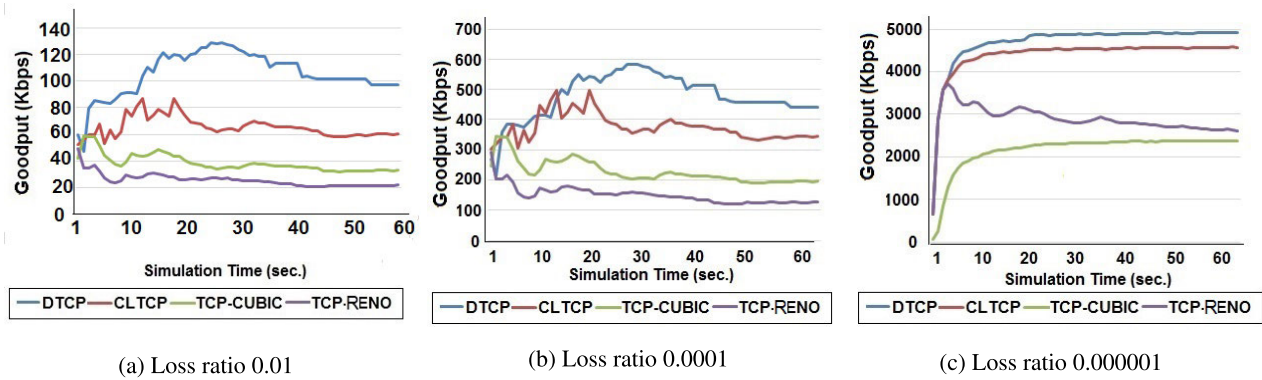


FIGURE 13. D-TCP Goodput comparison in lossy conditions (LTE-A based NS3 model).

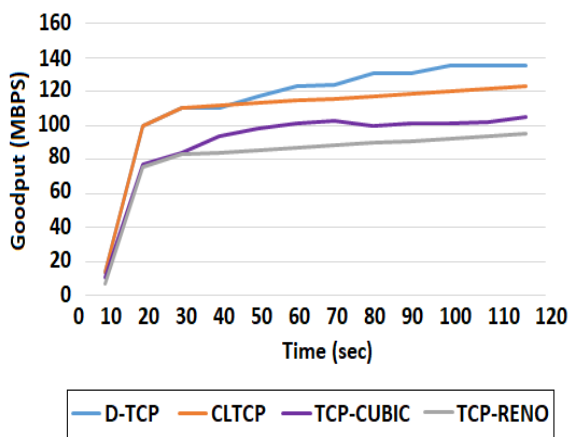


FIGURE 14. D-TCP Goodput comparison in high-speed no bottleneck scenario.

TCP BBR by default. Hence, the performance results of D-TCP are compared with legacy TCP algorithms (TCP-Tahoe, TCP-Reno, TCP-Cubic, TCP-Westwood) and the present (CLTCP [20]) algorithm.

We set the bottleneck data rate to 1 Mbps and RTT delay as 40 ms, to measure the CWND growth in random loss conditions. Fig. 12 presents the CWND values with iterations of ACKs. It is observed from Fig. 12 in case of CLTCP [20] algorithm, CWND increases aggressively as the deep drops in CWND are acknowledged. Moreover, in the case of bursty traffic conditions, the CLTCP algorithm takes more time to restore to its preceding state. As shown in Fig. 12, for our proposed D-TCP, the drops in CWND are very rare and able to exploit the maximum accessible bandwidth. This is because we utilize the discrete time filter to estimate the bandwidth from current and previous bandwidth values. We normalized the estimated bandwidth, as to remove the packet aliasing that takes place while utilizing the discrete time filter.

To analyze the efficiency of D-TCP, we set the bottleneck data rate of the channel to 5 Mbps, RTT delay equal of 40 ms, MTU as 1500 and considered different packet ratio conditions. We compared the efficiency of

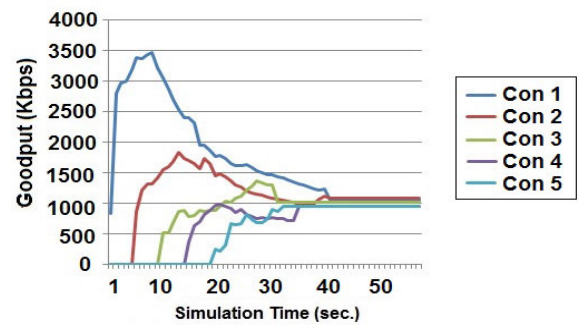


FIGURE 15. Intra-Algorithm fairness comparison.

D-TCP with TCP-Reno, TCP-Cubic and CLTCP. Fig. 13 (a), Fig. 13 (b) and Fig. 13 (c) depicts the Goodput comparison of D-TCP in lossy, lossless and near ideal conditions, respectively. As depicted in Fig. 13 (a) and Fig. 13 (b) for the packet loss ratio of 0.01 (lossy) and 0.0001 (lossless) respectively, D-TCP achieves higher throughput as compared to other TCP variants, because D-TCP is capable of utilizing the maximum available bandwidth. It is noticed from Fig. 13 (c), a similar trend is observed for the packet loss ratio of 0.000001. Moreover, we introduce the wireless losses and observed that D-TCP able to detect wireless losses and avoid the steepest CWND drops. The other TCP variants TCP-Reno, CLTP and TCP-Cubic assume these losses as congestion in the network whereas D-TCP can differentiate between packet drop due to congestion and wireless losses. D-TCP achieves the maximum goodput of 4927 Kbps. TCP-Cubic, TCP-Reno and CLTCP achieve maximum goodput of 2379.68 Kbps, 3706.93 Kbps and 4579.66 Kbps, respectively. The goodput of D-TCP is 32.9% higher than TCP-Reno, 7.5% higher than CLTCP and more than twice of TCP-CUBIC. We also performed test in high-speed LTE-A network without any bottleneck link. Fig. 14 depicts the goodput comparison of D-TCP in high-speed LTE-A network. D-TCP achieves higher throughput than other congestion algorithms even in high-speed wireless network.

In computer network fairness measure is used to ascertain that different application/protocols are sharing fair system



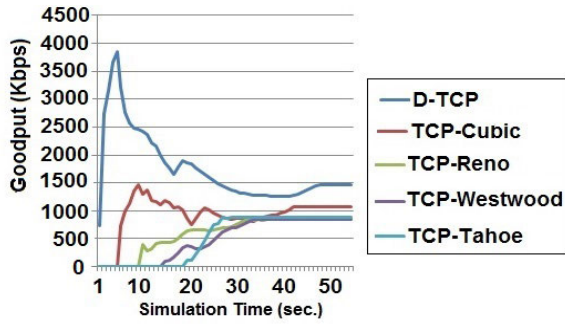


FIGURE 16. Inter-Algorithm fairness comparison.

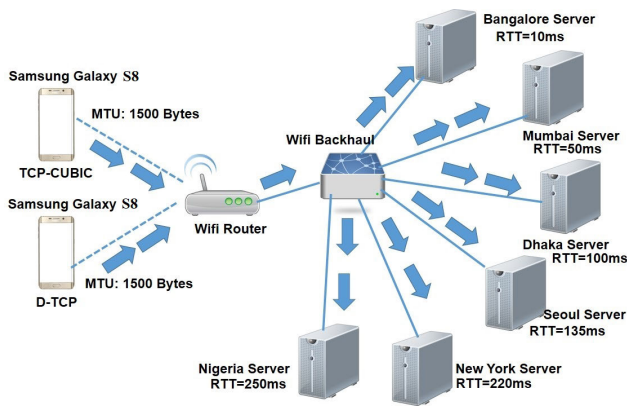


FIGURE 17. Setup for live air experiment.

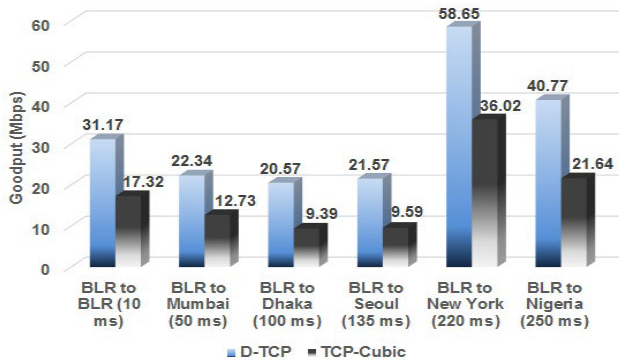


FIGURE 18. Goodput for D-TCP and TCP-Cubic algorithms (live air experiment).

resources. TCP fairness requires that a connection within an algorithm receives no large share than other connection or a new protocol does not acquire a large share of network resources as compared to other TCP protocols. If a protocol acquires uneven resources, this creates a problem such as a congestion collapse. It is important for the congestion control mechanism to interact well with TCP. To analyze the intra-algorithm fairness of D-TCP, we considered five connections of D-TCP, introduced sequentially. Fig. 15, presents the D-TCP Goodput with five connections. It is observed from Fig. 15 that with the insertion of distinct D-TCP connection consecutively, the bandwidth shared evenly among the five available devices. After introducing all the five connec-

tion it converges to the fairness point. The value of  $\gamma_{DTCP}$  (described in section III-F3) is close to 1, after introducing all five connection, which shows that D-TCP is fair. To prove inter-algorithm fairness, we considered five devices configured with D-TCP, TCP-Cubic, TCP-Reno, TCP-Westwood & TCP-Tahoe and a single connection of 5 Mbps, which is shared among all the five variants. Fig. 16 depicts the goodput for different TCP variants. We ascertain from Fig. 16 that D-TCP has very low Bandwidth Plucking Rate (BPR) whereas, other TCP variants have comparatively high BPR. Because of low BPR, D-TCP could utilize the unused bandwidth resulting in higher goodput as compare to other TCP algorithms.

### C. LIVE AIR EXPERIMENT RESULTS

To analyze the performance of D-TCP, we carried out a live air experiments in Samsung R&D India-Bangalore (SRI-B). We considered two Samsung Galaxy S8 devices and the devices are connected to different servers situated in Bangalore (India), Mumbai (India), Dhaka (Bangladesh), Seoul (South Korea), New York (USA) and Nigeria (Africa) through WiFi. We considered WiFi as 5G might expected to operate in Unlicensed spectrum bands in order to achieve the multi-Gigabit/s data rates for a large number of devices [54]. In our live air experiment one cellphone is configured with TCP-Cubic algorithm and the other is configured with proposed D-TCP algorithm. To obtain the SINR value needed by the D-TCP algorithm, our implementation exploits the Radio Interface Layer (RIL) of Android that fetches the signal information such as SINR, RSSI, and so on. Then the RIL exposes the SINR information to the TCP/IP stack through a `proc` variable. We measured the goodput by varying the RTT. Fig. 17 depicts the overall experimental setup.

Fig. 18 depicts the goodput for different RTT (Bangalore: 10 ms, Mumbai: 50 ms, Dhaka: 100 ms, Seoul: 135 ms, New York: 220 ms, Nigeria: 250 ms). It is observed from Fig. 18, the average goodput gain attained when the device connected to a server situated in Seoul is 124.92%. The average goodput gain achieved when the servers are situated in Bangalore, Mumbai, Dhaka, New York and Nigeria are 70.98%, 75.42%, 118.99%, 62.83% and 88.4% respectively. The phenomenal percentage gain attributes to the intelligent selection of dynamic factor  $N$ , which smartly controls the CWND and does not allow it to fall steeply in case of congestion and wireless losses.

### V. CONCLUSION

In this paper, we have proposed a D-TCP congestion control algorithm for mmWave NR. D-TCP estimates the accessible bandwidth and adjusts the congestion window accordingly in an adaptive increase adaptive decrease paradigm. Our proposed congestion control algorithm enables the full utilization of bandwidth in high-BDP network and minimizes the packet loss due to congestion and wireless loss. The performance of D-TCP is analyzed in mmWave NR, LTE-A simulations and live air experiment. The LTE-A simulation results demonstrate that comparing to TCP-Reno, our

proposed D-TCP achieves 32.9% higher goodput. Moreover, in case of live air experiment D-TCP attains 124.92% gain in goodput as compared to TCP-Cubic.

## REFERENCES

- [1] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 1617–1655, 3rd Quart., 2016.
- [2] C. Peng Fu and S. C. Liew, "TCP veno: TCP enhancement for transmission over wireless access networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 2, pp. 216–228, Feb. 2003.
- [3] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the Internet," *Commun. ACM*, vol. 55, no. 1, pp. 57–65, Jan. 2012.
- [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 *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2016, pp. 730–735.
- [5] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan, and S. Panwar, "TCP dynamics over mmWave links," in *Proc. IEEE 18th Int. Workshop Signal Process. Adv. Wireless Commun. (SPAWC)*, Jul. 2017, pp. 1–5.
- [6] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan, and S. Panwar, "The bufferbloat problem over intermittent multi-gbps mmWave links," 2016, *arXiv:1611.02117*. [Online]. Available: <http://arxiv.org/abs/1611.02117>
- [7] T. Azzino, M. Drago, M. Polese, A. Zanella, and M. Zorzi, "X-TCP: A cross layer approach for TCP uplink flows in mmWave networks," in *Proc. 16th Annu. Medit. Ad Hoc Netw. Workshop (Med-Hoc-Net)*, Jun. 2017, pp. 1–6.
- [8] B. Nguyen, A. Banerjee, V. Gopalakrishnan, S. Kasera, S. Lee, A. Shaikh, and J. Van der Merwe, "Towards understanding TCP performance on LTE/EPC mobile networks," in *Proc. 4th Workshop All Things Cellular, Oper., Appl., Challenges*, 2014, pp. 41–46.
- [9] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, 1988.
- [10] M. Allma, V. Paxson, and W. Stevens, *TCP Congestion Control*, document RFC 2581, Apr. 1999.
- [11] S. Floyd and T. Henderson, *The NewReno Modification to TCP's Fast Recovery Algorithm*, document RFC 2582, Apr. 1999.
- [12] L. Xu, K. Harfoush, and I. Rhee, "Binary increase congestion control (BIC) for fast long-distance networks," in *Proc. IEEE INFOCOM*, Hong Kong, Mar. 2004, pp. 2514–2524.
- [13] S. Ha, I. Rhee, and L. Xu, "CUBIC: A new TCP-friendly high-speed TCP variant," *ACM SIGOPS Oper. Syst. Rev.*, vol. 42, no. 5, pp. 64–74, Jul. 2008.
- [14] G. Vardoyan, N. S. V. Rao, and D. Towsley, "Models of TCP in high-BDP environments and their experimental validation," in *Proc. IEEE 24th Int. Conf. Netw. Protocols (ICNP)*, Nov. 2016, pp. 1–10.
- [15] Z. Chen, Y. Liu, Y. Duan, H. Liu, G. Li, Y. Chen, J. Sun, and X. Zhang, "A novel bandwidth estimation algorithm of TCP westwood in typical LTE scenarios," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Shenzhen, China, Nov. 2015, pp. 1–5.
- [16] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Queue*, vol. 14, no. 5, pp. 20–53, Oct. 2016.
- [17] M. Polese, R. Jana, and M. Zorzi, "TCP and MP-TCP in 5G mmWave networks," *IEEE Internet Comput.*, vol. 21, no. 5, pp. 12–19, Sep. 2017.
- [18] P. J. Mateo, C. Fiandrino, and J. Widmer, "Analysis of TCP performance in 5G mm-wave mobile networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–7.
- [19] M. Zhang, M. Polese, M. Mezzavilla, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi, "Will TCP work in mmWave 5G cellular networks?" *IEEE Commun. Mag.*, vol. 57, no. 1, pp. 65–71, Jan. 2019.
- [20] X. Jiang and G. Jin, "CLTCP: An adaptive TCP congestion control algorithm based on congestion level," *IEEE Commun. Lett.*, vol. 19, no. 8, pp. 1307–1310, Aug. 2015.
- [21] M. R. Kanagarathinam, S. Singh, I. Sandeep, A. Roy, and N. Saxena, "D-TCP: Dynamic TCP congestion control algorithm for next generation mobile networks," in *Proc. 15th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2018, pp. 1–6.
- [22] J. Wang, J. Wen, J. Zhang, Z. Xiong, and Y. Han, "TCP-FIT: An improved TCP algorithm for heterogeneous networks," *J. Netw. Comput. Appl.*, vol. 71, pp. 167–180, Aug. 2016.
- [23] J. Wang, J. Wen, Y. Han, J. Zhang, C. Li, and Z. Xiong, "CUBIC-FIT: A high performance and TCP CUBIC friendly congestion control algorithm," *IEEE Commun. Lett.*, vol. 17, no. 8, pp. 1664–1667, Aug. 2013.
- [24] T. A. Le, C. S. Hong, M. A. Razzaque, S. Lee, and H. Jung, "EcMTCP: An energy-aware congestion control algorithm for multipath TCP," *IEEE Commun. Lett.*, vol. 16, no. 2, pp. 275–277, Feb. 2012.
- [25] H.-j. Lee and J.-t. Lim, "Congestion control for streaming service in IEEE 802.11 multihop networks," *IET Commun.*, vol. 4, no. 12, pp. 1415–1422, Aug. 2010.
- [26] V. Jacobson, "Modified TCP congestion control and avoidance algorithms. End2end-interest mailing list," Apr. 1990.
- [27] B. Turkovic, F. A. Kuipers, and S. Uhlig, "Fifty shades of congestion control: A performance and interactions evaluation," 2019, *arXiv:1903.03852*. [Online]. Available: <http://arxiv.org/abs/1903.03852>
- [28] S. Floyd, "High speed TCP for large congestion windows," document RFC 3649, Dec. 2003.
- [29] D. Leith and R. Shorten, "H-TCP: TCP for high-speed and long-distance networks," in *Proc. PFLDnet*, 2004, pp. 1–16.
- [30] R. Al-Saadi, G. Armitage, J. But, and P. Branch, "A survey of delay-based and hybrid TCP congestion control algorithms," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 4, pp. 3609–3638, 4th Quart., 2019.
- [31] L. S. Brakmo and L. L. Peterson, "TCP Vegas: End to end congestion avoidance on a global Internet," *IEEE J. Sel. Areas Commun.*, vol. 13, no. 8, pp. 1468–1480, Oct. 1995.
- [32] G. Hasegawa, K. Kurata, and M. Murata, "Analysis and improvement of fairness between TCP reno and Vegas for deployment of TCP Vegas to the Internet," in *Proc. Int. Conf. Netw. Protocols*, Nov. 2000, pp. 177–186.
- [33] K. N. Srijith, L. Jacob, and A. L. Ananda, "TCP Vegas-A: Improving the performance of TCP Vegas," *Comput. Commun.*, vol. 28, no. 4, pp. 429–440, Mar. 2005.
- [34] D. X. Wei, C. Jin, S. H. Low, and S. Hegde, "FAST TCP: Motivation, architecture, algorithms, performance," *IEEE/ACM Trans. Netw.*, vol. 14, no. 6, pp. 1246–1259, Dec. 2006.
- [35] S. Belhaj and M. Tagina, "VFAST TCP: An improvement of FAST TCP," in *Proc. 10th Int. Conf. Comput. Modeling Simulation (UKSIM)*, 2008, pp. 88–93.
- [36] M. A. Alrshah, M. A. Al-Maqri, and M. Othman, "Elastic-TCP: Flexible congestion control algorithm to adapt for high-BDP networks," *IEEE Syst. J.*, vol. 13, no. 2, pp. 1336–1346, Jun. 2019.
- [37] R. King, R. Baraniuk, and R. Riedi, "TCP-africa: An adaptive and fair rapid increase rule for scalable TCP," in *Proc. IEEE 24th Annu. Joint Conf. IEEE Comput. Commun. Societies.*, Jun. 2005, pp. 1838–1848.
- [38] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," in *Proc. 25th IEEE Int. Conf. Comput. Commun.*, Jun. 2006, pp. 1–12.
- [39] A. Baiocchi, A. P. Castellani, and F. Vacirca, "YeAH-TCP: Yet another highspeed TCP," in *Proc. PFLDnet*, vol. 7, 2007, pp. 37–42.
- [40] S. Liu, T. Başar, and R. Srikant, "TCP-Illinois: A loss-and delay-based congestion control algorithm for high-speed networks," *Perform. Eval.*, vol. 65, no. 6, pp. 417–440, 2008.
- [41] H. Shimonishi, T. Hama, and T. Murase, "TCP-adaptive reno for improving efficiency-friendliness tradeoffs of TCP congestion control algorithm," in *Proc. PFLDnet*, 2006, pp. 87–91.
- [42] G. Marfia, C. Palazzi, G. Pau, M. Gerla, M. Sanadidi, and M. Roccetti, "TCP Libra: Exploring RTT-Fairness for TCP," in *Proc. Int. Conf. Res. Netw. Cham, Switzerland: Springer*, 2007, pp. 1005–1013.
- [43] C. Lee, J. Jung, and J.-M. Chung, "DEFT: Multipath TCP for high speed low latency communications in 5G networks," *IEEE Trans. Mobile Comput.*, early access, Jun. 4, 2020, doi: [10.1109/TMC.2020.3000041](https://doi.org/10.1109/TMC.2020.3000041).
- [44] W. Roh, J.-Y. Seol, J. Park, B. Lee, J. Lee, Y. Kim, J. Cho, K. Cheun, and F. Aryanfar, "Millimeter-wave beamforming as an enabling technology for 5G cellular communications: Theoretical feasibility and prototype results," *IEEE Commun. Mag.*, vol. 52, no. 2, pp. 106–113, Feb. 2014.
- [45] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao, and J. I. Tamir, "Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications," *IEEE Trans. Antennas Propag.*, vol. 61, no. 4, pp. 1850–1859, Apr. 2013.
- [46] J. Padhye, V. Firoiu, D. Towsley, and J. Kurose, "Modeling TCP throughput: A simple model and its empirical validation," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 4, pp. 303–314, 1998.
- [47] A. Tang, J. Wang, S. H. Low, and M. Chiang, "Equilibrium of heterogeneous congestion control: Existence and uniqueness," *IEEE/ACM Trans. Netw.*, vol. 15, no. 4, pp. 824–837, Aug. 2007.

- [48] J. Crowcroft and P. Oechslin, "Differentiated end-to-end Internet services using a weighted proportional fair sharing TCP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 28, no. 3, pp. 53–69, Jul. 1998.
- [49] B. Zhou, C. P. Fu, D. M. Chiu, C. T. Lau, and L. H. Ngoh, "A simple throughput model for TCP Reno," in *Proc. IEEE Int. Conf. Commun.*, vol. 12, Jun. 2006, pp. 5395–5400.
- [50] L. A. Grieco and S. Mascolo, "Mathematical analysis of westwood+TCP congestion control," *IEE Proc.—Control Theory Appl.*, vol. 152, no. 1, pp. 35–42, Jan. 2005.
- [51] C. B. Samios and M. K. Vernon, "Modeling the throughput of TCP Vegas," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 31, no. 1, pp. 71–81, Jun. 2003.
- [52] S. Floyd and M. Allman, *Comments on the Usefulness of Simple Besteffort Traffic*, document RFC 5290, Jul. 2008.
- [53] M. Mezzavilla, M. Zhang, M. Polese, R. Ford, S. Dutta, S. Rangan, and M. Zorzi, "End-to-end simulation of 5G mmWave networks," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2237–2263, 3rd Quart., 2018.
- [54] C. Lagen, L. Giupponi, S. Goyal, N. Patriciello, B. Bojovic, A. Demir, and M. Beluri, "New radio beam-based access to unlicensed spectrum: Design challenges and solutions," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 8–37, 1st Quart., 2020.



**MADHAN RAJ KANAGARATHINAM** (Senior Member, IEEE) received the B.E. degree in computer science and engineering from Anna University, Chennai, India, in 2012. He has seven years of working experience in design and development of TCP/IP protocols, Multipath TCP and UNIX flavored operating systems. He is currently working as a Chief Engineer with Samsung R&D Institute India-Bangalore. Previously, he worked as an Engineer with Aricent Technology (India) Pvt., Ltd. He is the author of ten articles, and more than 20 inventions. His current research interests include communication and network which include Pre-6G/Beyond 5G, next generation mobile network, software-defined network architecture, 40 transport layer protocols, and cross layer optimization technique. He is a member of ACM.



**SUKHDEEP SINGH** (Member, IEEE) received the Ph.D. degree from Sungkyunkwan University, South Korea. He is currently working as a Senior Chief Engineer with Samsung R&D India-Bangalore. His research interest includes 5G/6G wireless communications. His work has been published by the IEEE, Wiley, Taylor & Francis, and IET.



**IRLANKI SANDEEP** received the B.Tech. degree in computer science and engineering from NITW University, Warangal, India, in 2015. He is working as a Lead Engineer with Samsung R&D Institute India-Bangalore. He has five years of working experience in design and development of TCP/IP protocols and multipath-TCP. He is currently working on 5G commercialization and research in next generation protocols. He is the author of six publications and four inventions.



**HANSEOK KIM** received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2003. From 1992 to 1997, he worked at LG Electronics, developing office automation products. Since he joined Samsung Electronics in 2003, he has been developing 4G (WiMAX and LTE) and 5G systems. His research areas include network traffic modeling and analysis, flow control, resource management, and performance analysis in wireless networks.



**MUKESH KUMAR MAHESHWARI** received the Ph.D. degree from Sungkyunkwan University, South Korea, and the master's degree from the University of Leicester, U.K. He is currently working as a Faculty Member with the Department of Electrical Engineering, Bahria University, Pakistan. His research interests include 5G wireless communications and energy-efficient networks. His work has been published by the IEEE, Wiley, Taylor & Francis, and IET.



**JAEHYUN HWANG** received the B.S. degree in computer science from The Catholic University of Korea, South Korea, in 2003, and the M.S. and Ph.D. degrees in computer science from Korea University, Seoul, South Korea, in 2005 and 2010, respectively. He worked at Bell Labs, Alcatel-Lucent, Seoul, as a member of Technical Staff, from 2010 to 2015, and then worked at Samsung Electronics, South Korea, as a Senior Engineer, from 2015 to 2018. He has been with the Department of Computer Science, Cornell University, Ithaca, NY, USA, as a Postdoctoral Associate, since 2018. His current research interests include storage disaggregation, networked storage systems, data center networks, and multipath TCP.



**ABHISHEK ROY** received the B.E. degree from Jadavpur University, India, in 2000, the M.S. degree from The University of Texas at Arlington, in 2002, and the Ph.D. degree from Sungkyunkwan University, in 2010. He is currently working as a Senior Technical Manager at MediaTek. He has strong professional skills in 4G/5G/6G wireless system design, New Radio unlicensed, the IoT, cloud RAN, network modeling, and simulation. He has been published in more than 80 top-notch international journals and magazines (IEEE), coupled with ten patents and standards contributions.



**NAVRATI SAXENA** received the Ph.D. degree from the Department of Information and Telecommunication, University of Trento, Italy. She is currently an Assistant Professor with the Department of Computer Science, San Jose State University, CA, USA. Previously, she worked as an Assistant Professor with Sungkyunkwan University, South Korea, and Amity University, India, and as a Visiting Researcher with The University of Texas at Arlington. Her primary research interests include 4G/5G wireless, the IoT, smart grids, and smart environments. She has coauthored one book and has published more than 80 international journals in the IEEE, Elsevier, and Wiley, and many international conferences.

• • •