Received August 19, 2019, accepted September 19, 2019, date of publication October 4, 2019, date of current version October 17, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2945582

DL-TCP: Deep Learning-Based Transmission Control Protocol for Disaster 5G mmWave Networks

WOONGSOO NA[1], BYUNGJUN BAE[1], SUKHEE CHO[1], AND NAYEON KIM[2]

[1]Media Research Division, Electronics and Telecommunications Research Institute, Daejeon 34129, South Korea
[2]Department of Communications & Media, University of Science and Technology, Daejeon 34113, South Korea

Corresponding author: Woongsoo Na (wsna@etri.re.kr)

This work was supported in part by the Institute for Information and Communications Technology Promotion (IITP) grant funded by the Korea Government (MSIT) through the terrestrial UHD-based disaster broadcasting service for reducing disaster damage under Grant 2018-0-01364.

ABSTRACT The 5G mobile communication system is attracting attention as one of the most suitable communication models for broadcasting and managing disaster situations, owing to its large capacity and low latency. High-quality videos taken by a drone, which is an embedded IoT device for shooting in a disaster environment, play an important role in managing the disaster. However, the 5G mmWave frequency band is susceptible to obstacles and has beam misalignment problems, severing the connection and greatly affecting the degradation of TCP performance. This problem becomes even more serious in high-mobility drones and disaster sites with many obstacles. To solve this problem, we propose a deep-learning-based TCP (DL-TCP) for a disaster 5G mmWave network. DL-TCP learns the node's mobility information and signal strength, and adjusts the TCP congestion window by predicting when the network is disconnected and reconnected. As a result of the experiment, DL-TCP provides better network stability and higher network throughput than the existing TCP NewReno, TCP Cubic, and TCP BBR.

INDEX TERMS Deep-learning, mmWave, TCP, 5G, supervised-learning.

I. INTRODUCTION

In recent years, a series of natural disasters, such as the tsunami in Indonesia, the magnitude 9.0 earthquake in Japan, and the hurricane in USA, have been threatening the life of mankind. Natural disasters have been occurring increasingly frequently in the world along with abnormal weather phenomena, and taking human lives with many financial losses. To minimize damage in case of a disaster, it is very important to quickly recognize it and accurately and promptly communicate useful information such as emergency situations and action guidelines to all people in the disaster area [1].

However, when a disaster occurs, there is a limitation to delivering emergency information because the existing infrastructure (e.g., LTE base station and WiFi Hotspot) is broken, and thus, the connectivity is cut off. To solve this problem, employing unmanned aerial vehicles (UAVs) with IoT devices for shooting that roam and shoot the scene of the emergency situation with a small 5G base station can be one of the solutions. The UAVs can shoot high-definition videos for emergency situations and deliver them quickly to The broadcast station through the 5G core network [1]. In addition, this video can be utilized for rescue operations or broadcasts to everybody in a disaster area. For a captured video image to be efficiently used for lifesaving activities and disaster management, 1) high-quality videos should be transmitted without loss (high throughput/reliability) and 2) with low latency. As 5G networks can provide high-throughput (enhanced mobile broadband), fast, and accurate services (ultra reliable and low latency), delivering emergency information is a good use case for 5G networks (Fig. 1).

The 5G networks use a wide frequency band, millimeter wave (mmWave) band, to provide high throughput.[1] Although mmWave has a wide bandwidth, beamforming technology is absolutely necessary owing to its high path loss and strong directivity. Fortunately, the 3GPP 5G stan-

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

[1]In fact, in 5G, numerology is applied to cope with several operating frequencies (e.g., 3.5 and 28 GHz), but in this paper, we discuss only mmWave with a high bandwidth, assuming that a high-quality video image is transmitted.

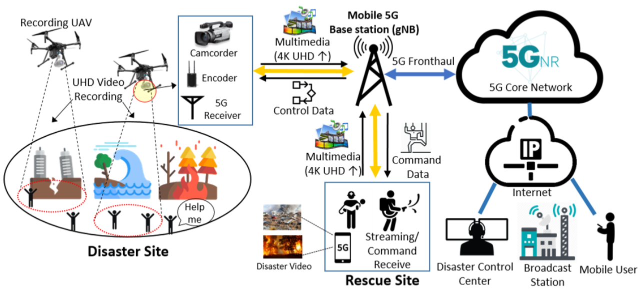


FIGURE 1. System architecture of disaster 5G networks.

dard groups have discussed and proposed beamforming technologies for 5G networks [2]. Despite the advancement of beamforming technology, the following problems remain in the mmWave band.

- **Blockage problem**: The phenomena that the signal cannot pass through the obstacle owing to the directivity and the receiving SNR value is severed.
- **Beam misalignment**: The receiver has a poor SNR value owing to nonmatching of transmitting/receiving beam pairs.

In particular, the use of the transmission control protocol (TCP) is essential owing to the characteristics of disaster communication, where end-to-end communication reliability is very important. However, in 5G networks, especially in mmWave band, TCP is very vulnerable owing to beam misalignment and blockage issues. TCP in the mmWave band has been extensively studied recently. In [3], [4], the authors simulated the performance of several existing TCP schemes, such as Reno, Cubic, and BBR, to verify how TCP will work in the mmWave band. Furthermore, in [3] and [4], the authors determined that the conventional loss-based TCP, such as Ha *et al.* [5], is not suitable for the mmWave capacity owing to its wide bandwidth, while congestion-based TCP, such as BBR, is more suitable. In addition, the authors discussed how beam tracking, handover, and mobility affect TCP performance in 5G mmWave networks. However, no TCP for the 5G mmWave network has been proposed yet.

This paper proposes a new TCP for transmitting high-quality videos (UHD 4K level) at low latency in disaster 5G mmWave networks. The proposed TCP performs learning based on mobility, location, and reception signal-to-noise ratio (SNR) value of the terminal, and the learning agent (UE) predicts the time duration for which the transmitting signal is disconnected (blockage duration). If the blockage duration is less than the TCP retransmission timeout (RTO) value, the congestion window (*cwnd*) value is fixed and buffering is performed for the corresponding time to utilize the mmWave capacity. By contrast, the congestion control (CC) algorithm is performed by determining that the actual congestion has occurred.

The contributions of this study are summarized as follows:

- We analyzed and identified how the mmWave technical challenges (blockage problem and beam misalignment) affect TCP in the disaster 5G mmWave network.
- We designed a machine learning (ML) model that predicts blockage duration based on mobility, location, and received SINR value of the TCP sender in the disaster 5G mmWave network. The proposed ML model can predict blockage duration with a probability of about 90% or more, and is independent of the mobility model of UE (TCP sender).
- Based on blockage duration, we proposed a new loss-based TCP model suitable for the mmWave environment. The proposed TCP distinguishes between a temporary disconnection of a link and actual congestion, and carries out a CC algorithm according to the situation.
- We determined that our proposed TCP outperforms other existing TCPs in not only the disaster site but also any topology (sport stadium, smart city, indoor, etc.) with the 5G mmWave environment.

The remainder of this paper is organized as follows. In Section II, we describe related work on the mmWave 5G network and existing TCPs. Furthermore, we describe the problem of conventional TCPs in disaster 5G mmWave networks. In Section IV, we describe the proposed machine learning architecture for ML-based TCP in 5G mmWave networks. We compare the performance of the proposed TCP with the existing TCPs in various network typologies in V. Finally, in Section VI, the conclusions and suggestions for future research directions are detailed.

II. RELATED WORK

TCP research has been around for a long time and has become one of the core protocols of the Internet. Thus, various TCP versions have been proposed accordingly. In this section, we present a brief overview of previous studies on 1) conventional TCPs, 2) TCPs in 5G mmWave environment, and 3) TCPs with machine learning.

A. CONVENTIONAL TCP

The conventional congestion control technique for TCP assumes that all segmentation loss reasons are based on congestion. Therefore, the most popular TCP-Reno andNew-Reno, at first, exponentially increase the congestion window in the slow start phase and go through the congestion avoidance phase when segmentation loss occurs, to adjust the transmission rate [6], [7]. However, in wireless media with a strong randomness factor, segmentation loss is caused not only by network congestion but also by factors such as interference, path loss, and mobility [8].

Recently, various new versions of TCP studies have been conducted [5], [9]–[11]. Of these, TCP Cubic [5] is the most popular TCP technique currently used in Linux Kernels 2.6.19 and above. In TCP Cubic, the size of the congestion window grows very quickly right before the congestion event occurs, and then, convexly increases the size of the congestion window to prove the network state. Although TCP Cubic is designed for high-bandwidth-delay product (BDP) networks, it is not suitable for wireless networks, which have random channel characteristics.

One of the most popular TCPs, bottleneck bandwidth and round-trip propagation time (BBR) [10], is used to maintain an optimal congestion window based on the current network bandwidth and round-trip time (RTT). However, BBR is not still suitable for wireless media where packets are still randomly dropped. To cope with the characteristics of wireless media, other wireless TCPs have been proposed for many years [12]–[15]. However, in these studies, the wireless access is considered the bottleneck area and is not suitable for mmWave networks with a large bandwidth.

B. TCP IN 5G MMWAVE NETWORK

With the release of 5G mobile communication standardization, the question of whether existing TCP works well in the mmWave band naturally emerges. In [3], [4], [16], the authors simulated the operation of the conventional TCP (Tahoe, Reno, Cubic, BBR, etc.) in 5G mmWave networks. They experimented end-to-end performance between a mobile terminal and a cloud server in an environment where signals are attenuated owing to obstacles such as buildings and trees. In their experiments, the authors determined that 1) the conventional TCPs take a long time to activate the wide bandwidth of mmWave, 2) the end-to-end latency increases owing to the high link error rate, and 3) a frequent RTO occurs.

In addition, in [16], the authors performed comprehensive experiments on how mobility management techniques (e.g., handover) affect TCP performance in 5G mmWave networks. They concluded that in a channel intermittent environment (e.g., mmWave band), the fast adaptation of the servicing base station is a key technology that can improve the TCP performance. These studies [3], [4], [16]–[18] raised a problem with TCP performance in the mmWave band, but did not present a new TCP suitable for 5G mmWave networks.

Additionally, in [17], the authors proposed advanced 5G-TCP suitable for a wide bandwidth of mmWave by adjusting the parameters of high-speed TCP (HSTCP). However, they focused on how quickly a 5G mmWave network with a wide bandwidth can be maximally activated without considering frequent link errors.

C. TCP WITH MACHINE LEARNING

Recently, machine learning techniques have effectively advanced the state-of-the-art for many research domains. In particular, machine learning can be applied to TCP applications such as congestion control [19], [20], network state prediction [21], [22], traffic classification [23], traffic monitoring, and security [24]. Congestion control, which is the main function of ML-based TCP, is classified into loss prediction through supervised learning and intelligent congestion window control through reinforcement learning.

In [19], the authors proposed a Q-learning framework with TCP design (QTCP), which enables senders to gradually learn the optimal congestion control policy in an online manner. QTCP performs congestion window control (increase/decrease/hold) based on the time intervals of transmitted packets and received ACK packets and average RTT. As Q-learning based TCP, throughput and RTT are modeled as utility so that each action can optimize the utility. Similarly, in [20], the authors proposed supervised learning- based TCP (LP-TCP) and reinforcement learning-based TCP. LP-TCP predicts the probability of loss of the current packet by learning the received ACK packet, current received *cwnd* size, RTT, and exponentially weighted moving average.

QTCP and LP-TCP have the advantage of being applicable to various network environments because it conducts congestion control through self- learning rather than rule-based learning. However, these studies predict or enhance network congestion control through packet-based information (congestion window size, RTT, time interval of ACKs, etc.). Therefore, congestion control algorithms will not work properly for blockage problems caused by mobility and beam misalignment problems (factors that are not caused by packets). This problem will become even more serious in the 5G mmWave disaster network, where mobility is frequent and several obstacles are placed. To cope with this problem, we propose a machine learning framework based on mobility, location, and received SNR information of the moving objects.

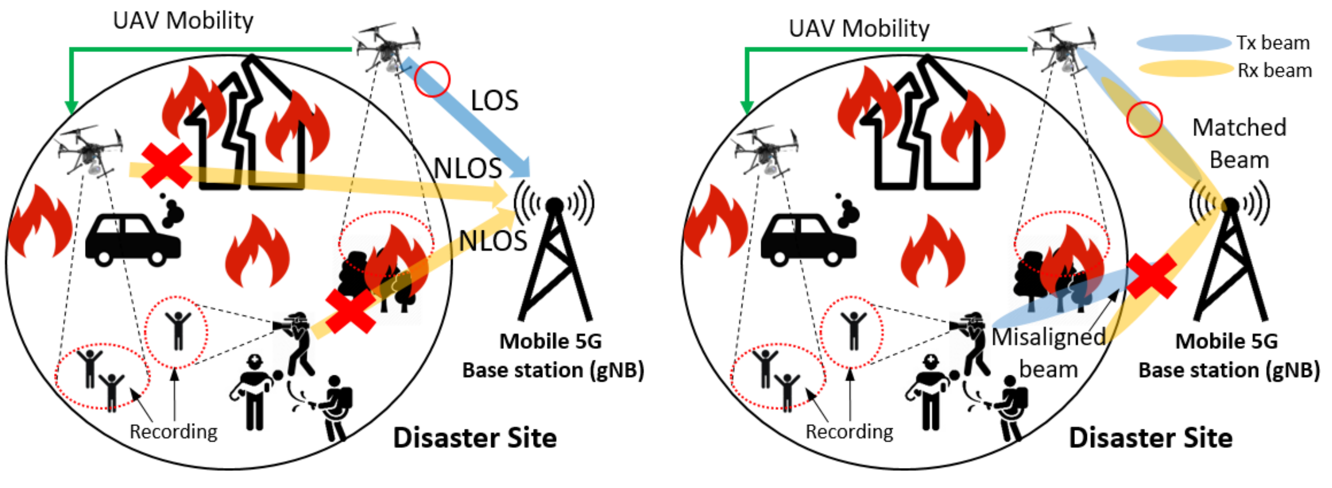
III. TCP IN DISASTER 5G mmWave NETWORKS

In this section, we describe our proposed 5G mmWave network model and related issues for the disaster site. As shown in Fig. 1, we assume an uplink scenario model where UAVs, for shooting extensively at disaster sites, and firefighters with an electronic news-gathering camera, for shooting details of the disaster sites, transmit the video to a broadcast station or control centers. They can transmit high-quality videos in raw data form without compression[2] to broadcast stations and disaster control centers via the 5G network. However, to transmit uncompressed video images without delay, a data rate on the order of Gbps is required, which enables a 5G mmWave network with a wide bandwidth.

In 5G mmWave networks, the beamforming technology is essential for compensating for a high path loss. The formed beams between the 5G base station and the transmission terminal (transmitting UE) provide a high antenna gain; however, it is vulnerable to obstacles due to the beam directivity. In addition, when the directions of the formed beams between the transmitting and receiving antennas are not matched, the receiving gain is lowered, and therefore, the SNR value decreases. In particular, in the disaster 5G mmWave network model, the transmitting UEs have variable mobility, which can lead to communication problems.

Fig. 2 (a) shows the blockage problem in the mmWave environment. UAVs that capture disaster scenes can transmit video images to a 5G basestation (gNB) within line-of-sight (LOS) locations. However, if non-line-of-sight (NLOS) is formed due to obstacles such as trees/collapsed buildings, it is impossible to transmit signals in the mmWave band.

[2] In general, video encoder equipment is limited to portable video cameras for reasons of heavy weight and battery inefficiency.



(a) Blockage problem in disaster 5G mmWave Networks (b) beam misalignment in disaster 5G mmWave Networks

FIGURE 2. Communication problems in mmWave band.

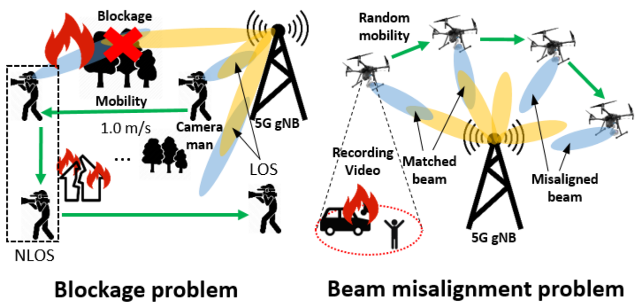


FIGURE 3. Blockage problem and beam misalignment problems.

Camera mans moving fast to cover disaster scenes and UAVs with fast mobility are more likely to be exposed to these problems. Similarly, Fig. 2 (b) shows the problem of beam misalignment in the mmWave environment. As shown in the figure, the UAV and gNB beams are correctly matched and the SNR of the signal is high, but the beams of the camera man and the gNB are misaligned, and thus, the signal cannot be received.

A. BLOCKAGE PROBLEM IN TCP

To analyze how a conventional TCP behaves when a blockage problem occurs in mmWave networks, we have configured a simple scenario, as shown in Fig. 3. In the left side of Fig. 3, initially, the camera man and gNB are located at LOS distance from each other, and thus, it can transmit the video data without any problem. The camera man moves to the NLOS area with 1.0 m/s mobility, and continues to transmit video data. Fig. 4 shows the SINR value received by the camera man. For simulation parameters, we set the receiving threshold to -5dB and data rate to 1 Gbps and use the NewReno TCP [7]. As shown in the graph of Fig. 4, when the camera man stays in the NLOS region, the SNR temporarily falls below the receiving threshold value.

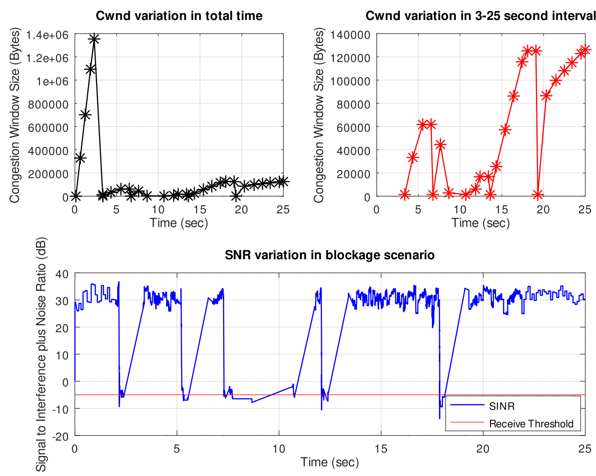


FIGURE 4. SNR and cwnd variation in blockage scenario.

Fig. 4 also shows the congestion window variation in the TCP layer in the above scenario. As shown in the figure, when the SNR value falls below the receiving threshold value, RTO occurs, which causes the *cwnd* value to be initialized frequently. This leads to decreased total network throughput.

One way to fundamentally solve the blockage problem is to install a base station to widen the LOS area, or install a mirror that captures the signal being transmitted and relays it to the NLOS area. However, in disaster areas where existing infrastructure has collapsed or a metropolitan-area network topology where many obstacles are concentrated, the above solution has limitations. Therefore, the blockage problem cannot be solved in a situation where the infrastructure is limited. However, if the blockage is intermittent, we can address part of the performance degradation by simply maintaining the transfer rate at the TCP end rather than initializing it. With this motivation, we applied the ML framework to distinguish RTOs between the blockage problem and network congestion

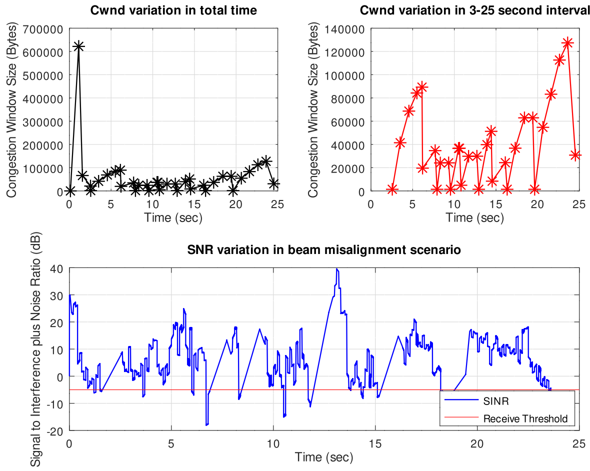


FIGURE 5. SNR and cwnd variation in beam misalignment scenario.

caused by network buffers, which we will describe further in the next section.

B. BEAM MISALIGNMENT IN TCP

As in the previous section, we have designed a simple scenario to investigate the behavior of TCP in beam-misalignment issues in the mmWave environment. In our experiment, as shown on the right side of Fig. 3, we set random mobility within the specified topology for relatively fast UAVs. The UAV is free to fly within the topology at a rate of 25 to 30 m/s and transmits video data to the gNB. As shown in Fig. 5, SNR fluctuation is very high with high mobility. In the mmWave environment, even if the phases of the transmitting and receiving antenna are slightly mismatched, the receiving power drops significantly [25]. Compared to the experimental results of low-speed movement (Fig. 4, it can be seen that the fluctuation in SINR in Fig. 5 is more intense. Initially, when the terminal and gNB match the beam, the communication is performed through the corresponding beam for a predetermined period. However, if the direction is changed due to the mobility of the terminal during this period, the beam pairs are mismatched and SNR decreases.

The beam sweep technique, which finds the optimal beam again after the beam is mismatched, can be performed after about 0.1-0.3 s (100-300 ms) [2].[3] However, if the UAV continues to move, the SINR is still continuously dropped. In other words, even if the terminal and gNB find the best beam pair for communication, the signal quality can significantly deteriorate. To solve these problems, there is a beam management technology, such as beam tracking, but there is an accuracy limitation in the environment where the mobility is rapidly changing. We only assume the beam sweep technique for recovering the beam misalignment problem, and the beam management technique is beyond the scope of this paper.

[3]According to 3GPP, the beam sweeping technique takes about 100-300 ms because it measures signal strength for all beam pairs of the transceiver.

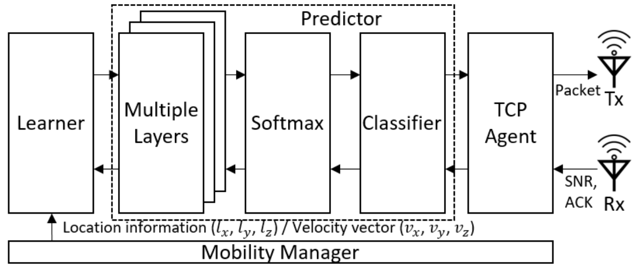


FIGURE 6. Architecture of the proposed Deep-learning based TCP scheme.

Here, we aim to control the behavior of TCP. As shown in Fig. 5, the beam misalignment problem causes *cwnd* initialization by RTO, which occurs intermittently. A frequent initialization of *cwnd* leads to the problem of not taking advantage of the wide bandwidth of mmWave. In the TCP layer, the solution is the same as that for the blockage problem. Even if beam misalignment occurs, the sender can recognize it and keep *cwnd* uninitialized. Then, the sender will find the optimal beam again in the beam sweep period and not waste the bandwidth.

IV. MACHINE LEARNING ARCHITECTURE

In this section, we describe the proposed deep-learning-based TCP (DL-TCP) in mmWave networks, which have blockage and beam misalignment problems. Fig. 6 shows the architecture of the proposed DL-TCP scheme. The TCP sender comprises a learner engine for learning, a predictor, a mobility manager for managing velocity and location information, and a TCP agent for determining TCP behavior. Each component plays the following roles:

- **A learner** learns the duration of network failure based on the information received from the mobility manager and Rx antennas.
- **A predictor** is a module that predicts whether a TCP RTO is a temporary or a long-term disconnect, based on the learned information.
- **Mobility manager** is a module that provides the current location information and velocity vector of the TCP sender.
- **TCP agent** controls the *cwnd* value based on the information predicted by the predictor.

As shown in section III, when a mobile TCP sender performs communications in the mmWave band, blockage and beam misalignment cause intermittent communications to be disconnected and initialize *cwnd*. In case of the blockage problem, if the TCP sender's mobility is relatively fast, the time to pass through the obstacle will be relatively short, and in the opposite case, the disconnect time will be long. At this time, the TCP agent can perform the following actions when RTO occurs owing to signal disconnection. If the network failure duration is short, the size of *cwnd* is maintained to prevent the data rate from dropping, and in the opposite case, the *cwnd* is initialized and enters the congestion avoidance phase.

t	l_x	l_y	l_z	v_x	v_y	v_z	γ_t
0.40	43.0	2.54	10	7.63	6.34	0	18.0
0.41	43.1	2.60	10	7.63	6.34	1	17.9
0.42	43.2	2.67	10	7.63	6.34	0	17.9
...	...	...	...	...	...	...	...
8.91	44.3	3.06	10	-1.8	-9.6	0	-7.1
8.92	44.3	2.97	10	-1.8	-9.6	0	-4.9
8.93	44.3	2.88	10	-1.8	-9.6	0	-7.6
8.94	44.2	2.79	10	-1.8	-9.6	0	-7.3
...	...	...	...	...	...	...	...

[Example of training data set]

H_l	H_s	H_n
0	0	1
0	0	1
0	0	1
...	...	...
0	1	0
0	0	1
1	0	0
1	0	0
...	...	...

[Example of result set]

FIGURE 7. An example of training data set and result set ($\theta_r = -5$ dB).

Even in the case of beam misalignment, the TCP agent can maintain the size of *cwnd*, but as the beam sweeping technique is performed after 100-300 ms, the network failure cannot exceed 100-300 ms. Therefore, even if the packet loss occurs, the network state can be restored within 100-300 ms. In other words, keeping the size of *cwnd* for that period is a good solution. However, the conventional TCP initializes the size of *cwnd* when it detects the packet loss event.

By combining the above analysis, we establish the following hypothesis:

1) $\mathbf{H}_l$: If the network failure duration is long, this may be due to network congestion or signal interruption from the obstacles. Thus, it is better to initialize *cwnd* size.
2) $\mathbf{H}_s$: If the network failure duration is short, it is a temporary signal interruption that can be recovered soon. Thus, it is better to maintain the size of *cwnd*.
3) $\mathbf{H}_n$: It is better to increase the size of *cwnd* if the LOS between the TCP sender and gNB is formed and communication is possible.

Based on $\mathbf{H}_l$, $\mathbf{H}_s$, and $\mathbf{H}_n$, our proposed DL-TCP determines whether to maintain the *cwnd* size by predicting the network failure duration when a packet loss event occurs.

A. TRAINING NETWORK FAILURE DURATION

To operate the proposed DL-TCP, training is required for a specific topology. In the training phase, the mobile TCP sender moves randomly and records the SNR value received from the gNB. We use the NS3-based mmWave network simulator to collect the training data.

As shown in Fig. 7, the structure of the training data is as follows:

1) **Time** (t): The time taken by the TCP sender to update the SNR value from the gNB.
2) **Location information** (l_x^t, l_y^t, l_z^t): The current TCP sender's x, y, z coordinates at t.
3) **Velocity vector** (v_x^t, v_y^t, v_z^t): The current TCP sender's mobility vector at t.
4) **SNR** (γ_t): The SNR value received by the TCP sender from the gNB at t.

Let N denote the amount of collected information in the simulation. Then, $N \times 1$ vectors of the time and SNR are defined by

$$T = [t_0, t_1, t_2, \ldots, t_{N-1}]^T, \quad S = [\gamma_0, \gamma_1, \gamma_2, \ldots, \gamma_{N-1}]^T, \tag{1}$$

respectively, and an $N \times 3$ matrix of the location information and velocity vector is defined by

$$L = \left[\bar{l}_0, \bar{l}_1, \bar{l}_2, \ldots, \bar{l}_{N-1}\right]^T, \quad V = [\bar{v}_0, \bar{v}_1, \bar{v}_2, \ldots, \bar{v}_{N-1}]^T, \tag{2}$$

where $\bar{l}_i = \left[l_x^i, l_y^i, l_z^i\right]$ and $\bar{v}_i = \left[v_x^i, v_y^i, v_z^i\right]$, respectively.

Based on the collected training datasets, the TCP sender can generate a result set as shown in Fig. 7. The result set ($\hat{y}$) is generated as a one-hot vector based on the network failure duration between the TCP sender and the gNB.

Assuming θ_r to denote the receiving threshold, the disconnected state at t ($\mathbf{D}_t$) is given by

$$\mathbf{D}_t = \begin{cases} 0, & \text{if } \theta_r \le \gamma_t \\ 1, & \text{otherwise.} \end{cases} \tag{3}$$

where $\mathbf{D}_t = 1$ indicates that the link state between the TCP sender and the gNB is disconnected. After calculating $\mathbf{D}_t, \forall t$, $\hat{y}$ can be generated as Algorithm 1. In Algorithm 1, η denotes the long network failure duration threshold. If the network failure duration is longer than η, it is considered a long network failure.

Based on the generated result set and training dataset, the learner trains parameters ($\bar{\mathbf{w}}_i$[4]) of multiple fully connected (FC) layers. The output of each layer is passed to the adjacent layer via the following rectified linear unit activation function:

$$f(x) = \begin{cases} 0, & \text{if } x < 0 \\ x, & x \ge 0. \end{cases} \tag{4}$$

During the training, the parameters of the FC layers are trained to inform the TCP agent about whether the network is disconnected temporarily or for a long period. For the cost function of the training, we consider the cross-entropy function, which is given as

$$C = -\sum_i y_i \log \tilde{y}_i + (1 - y_i) \log(1 - \tilde{y}_i) \tag{5}$$

where C and i denote the entropy loss and iteration epoch, respectively.[5]

B. PREDICTION NETWORK FAILURE DURATION AND DL-TCP

In this section, we describe a technique for predicting network failure duration through trained FC-layer parameters, and accordingly, the behavior of DL-TCP.

[4] $\bar{\mathbf{w}}_i$ denotes the vector of weight parameter of hidden layer i. $\bar{\mathbf{w}}_i = (w_i^0, w_i^1, \ldots, w_i^{j-1})$ where j denotes the number of states.

[5] Cross-entropy cost function is the most popular model for classification problems with many datasets due to low error rate and high learning efficiency [29], [30].

Algorithm 1 Network Failure Duration-Based Result Set Generation Algorithm

```
Result set (ŷ) generation function (T, S):
    Input  : Training data set T and S
    Output: ŷ                          /* Result set */
begin
    η ← T_RTO;  /* T_RTO: TCP timeout value; */
    foreach i        /* i: index (0 ≤ i < N) */
    do
        t ← t_i;
        if D_t == 0 then
            H_l^i ← 0; H_s^i ← 0; H_n^i ← 1;
            ŷ_i ← H_l^i, H_s^i, H_n^i;
        else
            foreach j    /* j: index (i ≤ j ≤ k); k = argmax_k(t_k) s.t. t_k ≤ t + η */
            do
                if D_t == 0 then
                    H_l^i ← 0; H_s^i ← 1; H_n^i ← 0;
                    ŷ_i ← H_l^i, H_s^i, H_n^i;
                    break;
                else
                    continue;
                end
            end
            H_l^i ← 1; H_s^i ← 0; H_n^i ← 0;
            ŷ_i ← H_l^i, H_s^i, H_n^i;
        end
    end
    return ŷ;
end
```

As mentioned in section IV-A, if a TCP sender puts t, l_x, l_y, l_z, v_x, v_y, v_z, and γ_t as input to the predictor, the current network status is classified as one of the three states ($\mathbf{H_l}$, $\mathbf{H_s}$, or $\mathbf{H_n}$). Fig. 8 shows our designed multi-class deep-neural-network architecture. Our DNN structure comprises K depths and we use the Xavier initializer [26]. In addition, Dropout is applied to prevent over-fitting [27] and the Adam optimizer [28] is used to minimize the loss function.

Our proposed DL-TCP operation is simple. Basically, DL-TCP works same as the most popular version of TCP Cubic [5]. In most TCP versions, including TCP Cubic, when a packet loss event occurs, the size of the current *cwnd* is initialized and the TCP agent enters the congestion control phase. However, in the proposed DL-TCP, when a packet loss event occurs, the predictor classifies the duration of the packet loss event and the TCP agent receives this information. As shown in Algorithm 2, the TCP agent performs the following actions:

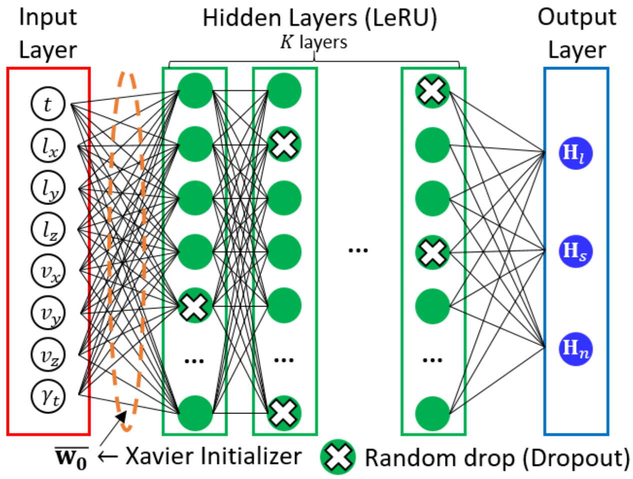


FIGURE 8. Multi-class deep-neural networks architecture.

Algorithm 2 Behavior of DL-TCP When Packet Loss Occurs

```
1  DL-TCP congestion control function:
2  Packet loss:
3  begin
4      t ← T_cur;           /* T_cur: current time */
5      y ← predict(t, l̄_t, v̄_t, γ_t);  /* y = [H_l, H_s, H_n] */
7      if H_l == 1 /* long time failure */ then
8          Delays RTO by T_RTO;
9          Keeps size of cwnd;
10     else if H_s == 1 /* short time failure */
       then
11         Retransmits packets that have been transmitted
           for T_cur − RTT/2;
12         Keeps size of cwnd;
13     else if H_n == 1 /* network congestion */
       then
14         Decreases size of cwnd;
15         Enters the steady state;
16 end
```

- If the TCP agent receives $\mathbf{H_l} = 1$, it delays the RTO by T_{RTO} while keeping the size of *cwnd* uninitialized (*lines 7-9*).
- If the TCP agent receives $\mathbf{H_s} = 1$, it immediately retransmits packets that have been transmitted for $T_{cur} - RTT/2$, without initializing the size of *cwnd* (*lines 10-12*)[6]
- If the TCP agent receives $\mathbf{H_n} = 1$, it determines network congestion. Thus, it decreases the size of *cwnd* and goes to the steady state to adjust the data rate (*lines 13-15*).

[6] $T_{cur} - RTT/2$ means the time of the most recently transmitted packet.

TABLE 1. Prediction accuracy of DL-TCP (random-walk model).

	Layer = 1		Layer = 2		Layer = 4		Layer = 8	
	SC1	SC2	SC1	SC2	SC1	SC2	SC1	SC2
Prediction Accuracy (Only DNN model)	89.52	88.46	93.24	93.41	96.17	95.76	**98.23**	**97.83**
Prediction Accuracy (/w dropout)	-	-	93.62	93.14	95.72	95.11	97.87	97.35
Prediction Accuracy (/w Xavier)	**91.05**	**90.87**	**95.25**	94.71	**96.74**	96.31	96.75	96.51
Prediction Accuracy (/w dropout, Xavier)	-	-	94.36	**94.95**	95.47	**96.74**	96.42	96.94

TABLE 2. Prediction accuracy of DL-TCP (Scan mobility model).

	Layer = 1		Layer = 2		Layer = 4		Layer = 8	
	SC1	SC2	SC1	SC2	SC1	SC2	SC1	SC2
Prediction Accuracy (Only DNN model)	90.23	89.88	94.41	94.11	96.81	96.32	**98.44**	98.12
Prediction Accuracy (/w dropout)	-	-	94.23	94.12	96.45	96.23	98.51	**98.27**
Prediction Accuracy (/w Xavier)	**92.45**	**92.12**	**96.56**	**95.88**	**97.35**	97.12	97.21	97.12
Prediction Accuracy (/w dropout, Xavier)	-	-	95.65	95.73	96.21	**97.17**	97.51	97.84

V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the proposed DL-TCP based on the mmWave NS-3 simulator [31]. First, we measure the prediction accuracy of the proposed DL-TCP. Basically, we assume a topology with 200 m^2 size, small obstacles of 3.5 m in height and 0.5 m in width, and large obstacles of 20 m in height and 10 m in width. We set two scenario models for comparison, as follows:

- **Scenario 1 (SC1):** We place fewer obstacles in a given topology. In this scenario, 10 small obstacles and 2 large obstacles are randomly deployed.
- **Scenario 2 (SC2):** We place obstacles densely within a given topology. In this scenario, 50 small obstacles and 10 large obstacles are randomly deployed.

The UAV is associated with the gNB in the topology and the gNB performs automatic beam sweeping every 100-300 ms. For the mobility model, we considered a random-walk model in which each UAV moves randomly and a scan mobility model that performs a mission to search the entire topology [32]. In addition, we assumed that the UAV speed follows a uniform distribution between 54 and 90 km/h. To collect training data, we collected SNR data for about 1 day.[7]

Table 1 shows the accuracy of the results predicted from the input values $(t, \bar{l}_t, \bar{v}_t, \gamma_t)$, by using the TCP Agent learned by the collected training data in a random-walk mobility scenario. We experimented with increasing the number of FCs while fixing the learning rate to 0.01 and the number of epochs to 2000. As shown in Table 1, the maximum prediction accuracy reached 98.23% and 97.83% for SC1 and SC2, respectively. Note that when a packet loss event occurs, it can be distinguished by about 97.8-98.23%, whether due to network congestion or link error. Moreover, if the TCP agent recognizes the link error, it can also detect whether the link error phenomenon has a long duration or short duration. Although the prediction accuracy of SC2 with a lot of obstacles is about 0.5% less than that of SC1, we can confirm that the proposed model has a high probability of 97% or more. Table 2 shows the prediction accuracy in the scan mobility scenario, where the UAV path is deterministic, and thus, has a relatively high prediction accuracy. As shown in Table 2, there is little difference in accuracy between

[7]In a real environment, UAV does not have any time to collect training data, but it learns while performing communication. In this experiment, we set the collecting period to check the achievable prediction accuracy when the training data are sufficient.

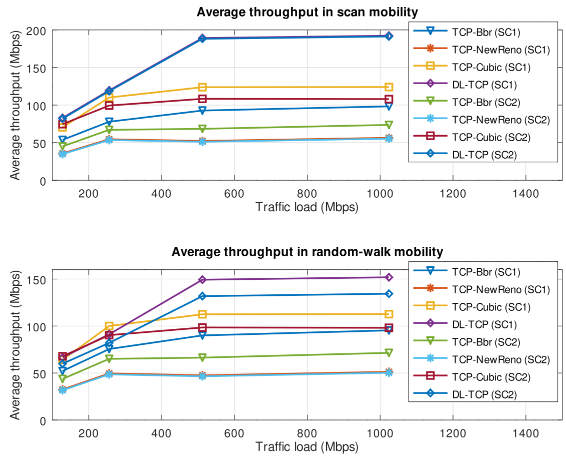


FIGURE 9. Average throughput versus traffic loads.

SC1 and SC2. In other words, it shows similar prediction accuracy without being greatly influenced by the presence or absence of obstacles. This is because our proposed model is independent of obstacles because the TCP agent learns based on mobility information and SNR in each region. In addition, because it has learned enough deterministic mobility, this result is analyzed as over-fitting. However, as the UAV at the disaster site has mission-oriented mobility, the path is programmed in advance, and thus, over-fitting is not a big problem.

To evaluate the actual performance of TCP based on the proposed DL architecture, we measured the following performance metrics:

- **Network throughput (Mbps):** total data traffic in bits transferred successfully from TCP sender to end terminal divided by time.
- **Round-trip-time (ms):** total round-trip time from TCP sender to end-terminal.
- **Variation in cwnd size (bytes):** variation in *cwnd* size of TCP sender.

For the simulation parameters, we set the data rate to 1 Gbps and the operating frequency to 28 GHz (mmWave), which is the same as the prediction accuracy measurement scenario. In addition, the performance of DL-TCP is compared to those of NewReno [7], BBR [10], and Cubic [5].

The above part of Fig. 9 shows the average throughput versus the traffic loads with the scan mobility model. As shown in the above figure, the average throughput increases with traffic, for all schemes. The proposed DL-TCP has about

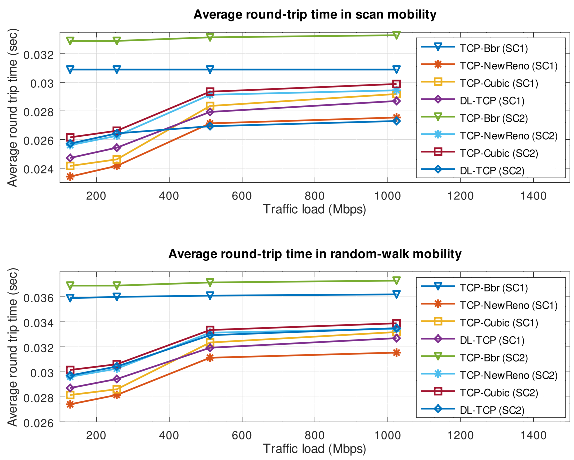


FIGURE 10. Average round-trip time versus traffic loads.

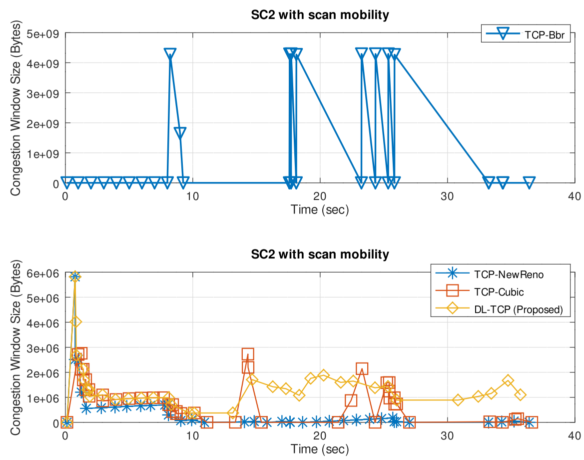


FIGURE 11. Variation of congestion window size with scan mobility.

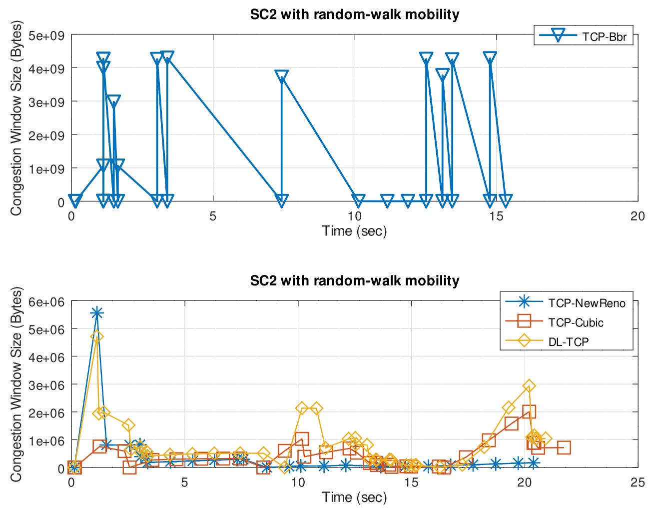


FIGURE 12. Variation of congestion window size with random-walk mobility.

36% performance improvement compared to other TCP techniques. In the conventional TCP, when a packet loss event occurs, it recognizes it as a network congestion and initializes a window or moves to a phase of gradually decreasing it. Meanwhile, DL-TCP distinguishes congestion from temporary disconnection and holds the window size when a temporary disconnection occurs to utilize the available bandwidth. In TCP-BBR and TCP-Cubic, the performance in SC1 with fewer obstacles is higher than that in SC2, and DL-TCP and TCP-NewReno show almost the same results regardless of the number of obstacles. This means that DL-TCP shows consistent performance regardless of the obstacles, and TCP-NewReno does not utilize the wide bandwidth of mmWave. The bottom part of Fig. 9 shows the average throughput with a random-walk mobility model. Unlike the scan mobility model, the random-walk mobility model shows a lower overall throughput for all schemes because the blockage and beam misalignment problems occur more frequently. In addition, the throughput of the proposed technique differs between SC1 and SC2. In the random-walk model, the mobility of the node is significantly fluctuating, where the learning of the TCP agent is not sufficiently performed. Therefore, it is analyzed that the prediction accuracy is lowered in SC2 with many obstacles. Nevertheless, as the environment is very tough, other TCP techniques also show poor performance and the proposed DL-TCP exhibits the best performance.

Fig. 10 shows the average round-trip time versus the traffic loads. As shown in the figure, TCP-NewReno, TCP-Cubic, and DL-TCP show similar trends with a difference of about 2-3 ms. The proposed DL-TCP shows RTT similar to that of the existing TCP despite high throughput. TCP BBR has a relatively high RTT because it tends to use full bandwidth over other TCP technologies. The results for random-walk mobility are about 4-5 ms lower than those for the scan mobility model. This is because similar to the result of throughput, the connection between nodes is frequently disconnected and RTT is increased accordingly. In the SC2 of the scan mobility model, RTT does not change much with the throughput because the TCP agent has been sufficiently learned, while the random-walk mobility model shows the same increase as other schemes for the proposed DL-TCP scheme.

Fig. 11 and Fig. 12 show the variation in *cwnd* size for all TCP schemes in SC2 with the scan mobility model and random-walk mobility, respectively. As shown in the figure, it shows very rapid *cwnd* fluctuations for all TCP schemes. This is because the link is unstable due to beam misalignment from the mobility and obstacles within the topology. In particular, in the case of the random-walk mobility model, the graph tends to become more intense. TCP-NewReno shows a tendency to maintain a low *cwnd* size without filling the wide bandwidth, due to frequent link errors. However, our proposed DL-TCP lowers *cwnd* only when the link is congested or when RTO occurs, and maintains the *cwnd* size when it determines that the link disconnection is temporary. Therefore, DL-TCP has higher average throughput, as shown in Fig. 9, compared to other TCP techniques that lower the *cwnd* size every time a packet loss event occurs.

VI. CONCLUSION

The 5G mmWave network is one of the most suitable models for rapid disaster response. However, due to the characteristics of the mmWave band, deterioration in signal strength from the obstacles and beam misalignment can be a factor in attenuating TCP performance. This paper proposes a DL-TCP suitable for the disaster 5G mmWave band. Our proposed deep-learning architecture learns the link disconnection time based on the node mobility information and signal strength, and predicts the link disconnection time when a packet loss event occurs. DL-TCP was designed to operate without wastage of mmWave bandwidth by performing proper *cwnd* size control according to the predicted time, which was confirmed through NS-3-based simulation. As the proposed model is a loss-based TCP model in which a packet loss event occurs, it is not a radical solution for avoiding link error. However, the deep-learning architecture proposed in this paper can be applied to the beam reflection technology to avoid obstacles in the mmWave band and beam management technology to provide a seamless communication environment. In the future, we will apply the deep-learning architecture to research on minimizing errors in the mmWave band.

REFERENCES

[1] S. A. R. Naqvi, S. A. Hassan, H. Pervaiz, and Q. Ni, "Drone-aided communication as a key enabler for 5G and resilient public safety networks," *IEEE Commun. Mag.*, vol. 56, no. 1, pp. 36–42, Jan. 2018.

[2] M. Giordani, M. Polese, A. Roy, D. Castor, and M. Zorzi, "A tutorial on beam management for 3GPP NR at mmWave frequencies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 173–196, 1st Quart., 2018.

[3] 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.

[4] 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.

[5] 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.

[6] V. Jacobson, "Congestion avoidance and control," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 18, no. 4, pp. 314–329, Aug. 1988.

[7] J. Hoe, "Improving the start-up behavior of a congestion control scheme for TCP," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 26, no. 4, pp. 270–280, Aug. 1996.

[8] Y. Tian, K. Xu, and N. Ansari, "TCP in wireless environments: Problems and solutions," *IEEE Commun. Mag.*, vol. 43, no. 3, pp. S27–S32, Mar. 2005.

[9] 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. 1465–1480, Oct. 1995.

[10] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," *Commun. ACM*, vol. 60, no. 2, pp. 58–66, Feb. 2017.

[11] K. Tan, J. Song, Q. Zhang, and M. Sridharan, "A compound TCP approach for high-speed and long distance networks," in *Proc. IEEE INFOCOM*, Apr. 2006, pp. 1–12.

[12] C. P. 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.

[13] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for wireless IP communications," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 4, pp. 747–756, May 2004.

[14] S. Bhandarkar, N. E. Sadry, A. L. N. Reddy, and N. H. Vaidya, "TCP-DCR: A novel protocol for tolerating wireless channel errors," *IEEE Trans. Mobile Comput.*, vol. 4, no. 5, pp. 517–529, Sep. 2005.

[15] S. Biaz and N. H. Vaidya, "'De-randomizing' congestion losses to improve TCP performance over wired-wireless networks," *IEEE/ACM Trans. Netw.*, vol. 13, no. 3, pp. 596–608, Jun. 2005.

[16] 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.

[17] I. Petrov and T. Janevski, "Advanced 5G-TCP: Transport protocol for 5G mobile networks," in *Proc. 14th IEEE Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jan. 2017, pp. 103–107.

[18] M. Polese, M. Mezzavilla, S. Rangan, and M. Zorzi, "Mobility management for TCP in mmWave networks," in *Proc. ACM Workshop Millimeter-Wave Netw. Sens. Syst.*, Oct. 2017, pp. 11–16.

[19] W. Li, F. Zhou, K. R. Chowdhury, and W. Meleis, "QTCP: Adaptive congestion control with reinforcement learning," *IEEE Trans. Netw. Sci. Eng.*, vol. 6, no. 3, pp. 445–458, Jul./Sep. 2019.

[20] Y. Kong, H. Zang, and X. Ma, "Improving TCP congestion control with machine intelligence," in *Proc. Workshop Netw. Meets AI ML*, Aug. 2018, pp. 60–66.

[21] B. A. A. Nunes, K. Veenstra, W. Ballenthin, S. Lukin, and K. Obraczka, "A machine learning framework for TCP round-trip time estimation," *EURASIP J. Wireless Commun. Netw.*, vol. 2014, no. 1, p. 47, Dec. 2014.

[22] D. H. Hagos, P. E. Engelstad, A. Yazidi, and Ø. Kure, "A machine learning approach to TCP state monitoring from passive measurements," in *Proc. Wireless Days (WD)*, Apr. 2018, pp. 164–171.

[23] T. T. T. Nguyen and G. Armitage, "A survey of techniques for Internet traffic classification using machine learning," *IEEE Commun. Surveys Tuts.*, vol. 10, no. 4, pp. 56–76, 4th Quart., 2008.

[24] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 305–316.

[25] S. Hur, H. Yu, J. Park, W. Roh, C. U. Bas, R. Wang, and A. F. Molisch, "Feasibility of mobility for millimeter-wave systems based on channel measurements," *IEEE Commun. Mag.*, vol. 56, no. 7, pp. 56–63, Jul. 2018.

[26] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. Artif. Intell. Statist.*, May 2010, pp. 249–256.

[27] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jun. 2014.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, Jan. 2014, pp. 1412–6980.

[29] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," Feb. 2016, *arXiv:1609.04836*. [Online]. Available: https://arxiv.org/abs/1609.04836

[30] Y. Tang, "Deep learning using linear support vector machines," Feb. 2013, *arXiv:1306.0239*. [Online]. Available: https://arxiv.org/abs/1306.0239

[31] 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.

[32] O. Bouachir, A. Abrassart, F. Garcia, and N. Larrieu, "A mobility model for UAV ad hoc network," in *Proc. Int. Conf. Inmanned Aircraft Syst. (ICUAS)*, May 2014, pp. 383–388.

WOONGSOO NA received the B.S., M.S., and Ph.D. degrees in computer science and engineering from Chung-Ang University, Seoul, South Korea, in 2010, 2012, and 2017, respectively. He is currently a Senior Research Engineer with the Media Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. Prior to joining the ETRI, he was an Adjunct Professor with the School of Information Technology, Sungshin University, Seoul, from 2017 to 2018. His research interests include TCP, mmWave systems, wireless mobile networks, 5G broadcasting systems, intelligent media, and beyond 5G.

BYUNGJUN BAE received the B.S., M.S., and Ph.D. degrees in electronics engineering from Kyungpook National University, South Korea, in 1995, 1997, and 2006, respectively. From 1997 to 2000, he was a Researcher with LG Electronics Inc., where he was involved in digital signal processing in digital television. Since 2000, he has been with the Media Research Division, Electronics and Telecommunications Research Institute (ETRI), Daejeon, South Korea. He has also been a Professor with the Department of Communication and Media Technology, University of Science and Technology (UST), Daejeon. His current research interests include next-generation broadcasting protocols and systems, emergency information signal processing, and intelligent media processing.

SUKHEE CHO received the B.S. and M.S. degrees in computer science from Pukyong National University, South Korea, in 1993 and 1995, respectively, and the Ph.D. degree in electrical and computer engineering from Yokohama National University, Japan, in 1999. She joined the Media Research Division, Electronics and Telecommunications Research Institute (ETRI), South Korea, in 1999, as a Senior Member Engineering Staff, where she is currently a Principal Researcher. Her current research interests include video processing, video coding, media application format, and metadata for media delivery services.

NAYEON KIM received the M.S. degree in communication and media technology from the University of Science and Technology, Daejeon, South Korea, in 2018, where she is currently pursuing the Ph.D. degree in communication and media technology. Her current research interests include media transport and media intelligence.