Simulation study of TCP proxy in multi-connectivity enabled 5G mmWave network

Goodsol Lee, Siyoung Choi, and Saewoong Bahk
Department of Electrical and Computer Engineering and INMC, Seoul National University, Seoul, South Korea
Email: {gslee2, sychoi}@netlab.snu.ac.kr, sbahk@snu.ac.kr

***Abstract*—With the rapid growth in mobile data traffic, 5G communication has been emerging. 5G communication utilizes mmWave links owing to its large bandwidth, but the high-frequency spectrum is not as reliable as the sub-6GHz bands. This is critical to TCP traffic because the network can not distinguish between link errors from network congestion. Multi-connectivity is a promising technology that provides link robustness at the network level. Meanwhile, TCP proxy, an intermediate node that accelerates TCP, can improve TCP throughput in high speed communication with mmWave link. Even though TCP performance in mmWave networks improves by adopting multi-connectivity and TCP Proxy, there is no study that considers using these two together. We integrate the two schemes and simulate the operation of TCP proxy on the multi-connectivity enabled mmWave network using ns-3 simulation. From simulation results, we find that performance of TCP proxy depends on the location of proxy placement, backhaul delay, and buffer size.**

***Index Terms*—TCP, TCP proxy, mmWave network, multi-connectivity, ns-3 simulation**

I. Introduction

The exponential increase of mobile data traffic has led evolution to fifth-generation (5G) communication. One of the main features of 5G is utilization of high-frequency spectrum, which is known as millimeter-wave (mmWave) spectrum. Comparing with traditional sub-6GHz bands, three distinctive characteristics of mmWave are large bandwidths, the sensitivity to blockages, and high path loss. Hence a mmwave link allows fast communication only in limited situations. To overcome the shortcoming, multi-connectivity (MC) is proposed for mmWave network to provide robust radio link [1]. A MC enabled network is composed of an master node (MN) and a secondary node (SN). The MN is responsible for mobility management with sub-6GHz link and the SN provides high throughput data transmission with mmWave link [2].

However, additional backhaul delay is added since data path of SN passes through backhaul between an SN and an MN. Because of the high cost of backhaul deployment, practical implementation prefers non-ideal backhaul, which incurs additional delay in an MC enabled network. TCP traffic is heavily influenced by this additional backhaul delay. TCP server increases its sending rate with acknowledgement (ACK) packets of sent packets. To receive an ACK packet, TCP server waits for one round-trip-time (RTT). If RTT is long, mmWave link should wait for quite long time to utilize network's full bandwidth. To solve this problem, we propose to deploy TCP proxy on BSs in an MC enabled network and show its performance enhancement with simulation results.

Saewoong Bahk is the corresponding author.

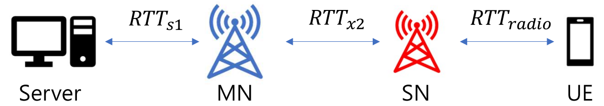


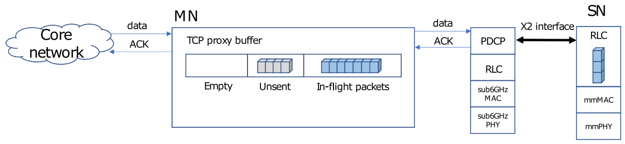
Fig. 1. End-to-end data path model of multi-connectivity enabled network. MN is a master node and SN is a secondary node. Data traffic passes through S1 interface link, X2 interface link, and radio link.

In this paper, we especially focus on TCP proxy, which operates as a middlebox for TCP data packets [3]. TCP proxy can be deployed on the middle of a network and enhance performance of overall TCP operation by splitting end-to-end connection into a server to proxy session and a proxy to user session. We consider deploying a TCP proxy on the mmWave network, since performance improvements due to the TCP proxy with mmWave link is significantly large by dividing unreliable mmWave link error and core network congestion [4]. However, there are no works that considered TCP proxy with MC. If TCP proxy is deployed on an MC enabled mmWave network, additional delay of backhaul between BSs should be considered in placement of TCP proxy. We deploy TCP proxy on an MN or an SN, and show that performance enhancement of TCP proxy varies with placement. To see the performance difference of different TCP proxy placement, we implement TCP proxy on ns3-mmWave module [5], and evaluate it in the MC enabled network with various backhaul delay and buffer sizes of TCP proxy.

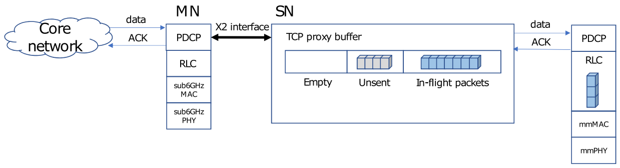
The paper is organized as follows. Section II explains the system model about TCP proxy operation and deployment on the MC enabled network. Section III shows simulation results and evaluation of the TCP proxy in the MC enabled network. Section IV summarizes and concludes our work.

II. System Model

In the following, we consider TCP proxy on MC enabled network. First, we describe our MC model that UE connects to both MN and SN simultaneously. Then TCP proxy deployed on BS is introduced with its several advantages. Finally, we



(a) TCP proxy on the MN.



(b) TCP proxy on the SN.

Fig. 2. TCP proxy with different placement in the multi-connectivity enabled network. Buffer of TCP proxy consists of empty space, unsent packets, and in-flight packets.

introduce how different TCP proxy deployment place can affect user's QoE.

A. Multi-connectivity

In our model, we assume one UE connects to one sub-6GHz BS, which operates as the MN and one mmWave BS, which operates as the SN. Simultaneous multiple connections provide seamless data transmission by switching data path to other connections quickly if one connection is blocked. For MC, X2 interface connection is required between an MN and an SN as shown in Fig. 1. X2 interface is logical connection between BSs, and can be installed with a set of ideal backhaul or non-ideal backhaul. Unlike ideal backhaul, non-ideal backhaul incurs additional delay in an end-to-end path, but has lower cost than ideal backhaul. Besides, dense small cell deployments in 5G network burden operators due to not only BS deployments but also X2 interface deployments between BS, when they adopt MC enabled network. All X2 interface cannot be an ideal backhaul in economic aspects, and we consider X2 interface as a non-ideal backhaul.

B. TCP proxy

In this subsection, we describe characteristics and expected performance enhancement of considered TCP proxy. By deploying TCP proxy, growth rate of CWND increases, wired network congestion and wireless channel error can be distinguished, and BS buffer overflow is prevented. TCP proxy is deployed on an MN or an SN as shown in Fig. 2a and Fig. 2b. The difference between the two cases is that PDCP of the MN is connected to RLC of the SN in Fig. 2a, while PDCP of the MN is connected to the TCP proxy of the SN and it sends data to the SN's PDCP in Fig. 2b. This is because PDCP layer compress TCP/IP headers. Since TCP proxy requires intact TCP/IP headers, the MN forwards packets without going through PDCP in Fig. 2b. In both cases, overall TCP session is divided into a server to proxy session and a proxy to UE session by TCP proxy.

Since two sessions are independent, CWND of each session grows independently. From this feature, TCP proxy deployed network increases overall CWND faster than that of non-proxy network, since each session's round trip time (RTT) becomes shorter than overall RTT. MmWave network provides large bandwidth, and fast CWND growth is required to utilize mmWave fully. TCP proxy can help faster CWND growth.

By splitting end-to-end session, TCP proxy can divide congestion event as a network congestion and a link error. With TCP proxy, network does not need to drop its congestion window with frequent mmWave link error. Also if CWND is dropped, each sessions in TCP proxy enable faster recovery of CWND for short RTT.

TCP proxy holds packets from a server until it receives ACK packets from UE. Buffer of TCP proxy consists of in-flight packets, unsent packets and empty space as shown in Fig. 2a and Fig. 2b. In-flight packets are sent but not yet received ACK packets. After receiving ACK packets, in-flight packets are dequeued from the buffer. Unsent packets will be sent when congestion window of TCP proxy is available. TCP proxy holds unsent packets when server to proxy session is slower than proxy to UE session. Empty space is an available buffer. The important thing is that TCP proxy prevents buffer overflow by notifying its empty buffer size to a server with ACK packets as a receive window size (RWND). Since TCP determines its sending rate as the minimum of CWND and RWND, TCP proxy can regulate TCP's sending rate of server to prevent buffer overflow from BS.

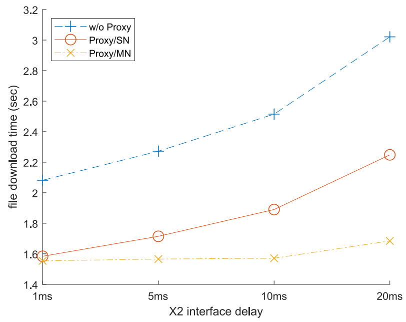
Parameter	Value
mmWave carrier frequency	28 GHz
mmWave BS TX power	30 dBm
User TX power	25 dBm
mmWave bandwidth	200 MHz
TCP proxy buffer size	[1, 10] MB
RLC buffer size	[1, 10] MB
X2 link delay	[1, 5, 10, 20] ms
Server to BS delay	[1, 30] ms

TABLE I: Simulation parameters

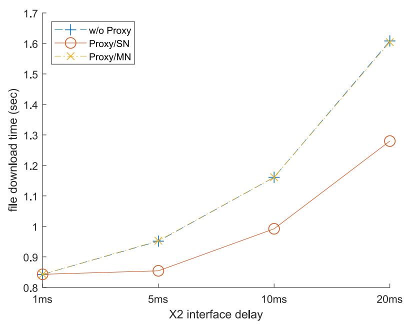
C. User's QoE with different TCP proxy placement

We consider an MN and an SN as candidates for the TCP proxy placement. The sessions split by TCP proxy will have different RTT depending on where TCP proxy is placed. Since the CWND growth rate depends on RTT and the amount of data that the user actually receives depends on the CWND, the QoE of the user can be changed according to the placement of the TCP proxy. Also, we consider the buffer size of TCP proxy, which is related to RWND. The small TCP proxy buffer size ensures that even if the server has a large CWND, it can report a small RWND and prohibit a high data rate from server.

In Fig. 1, an overall RTT of TCP traffic is $RTT_{s1} + RTT_{x2} + RTT_{radio}$. With TCP proxy, the overall RTT is divided by an RTT of server to proxy and an RTT of proxy to UE. With Fig. 2, we consider two TCP proxy placement model. If TCP proxy is deployed on the MN like Fig. 2a,

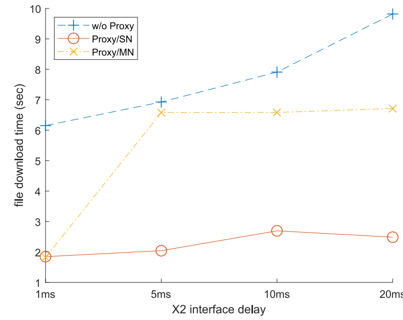


(a) File downloading from the remote server with sufficient buffer.

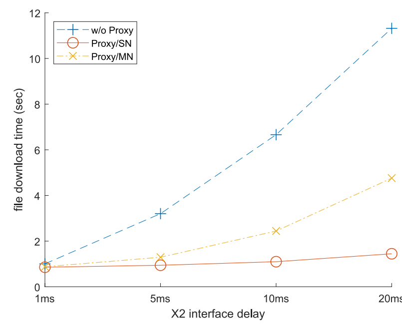


(b) File downloading from the edge server with sufficient buffer.

Fig. 3. File download time in the sufficient buffer situation. The size of TCP proxy buffer and RLC buffer is 10 MB.



(a) File downloading from the remote server with limited buffer.



(b) File downloading from the edge server with limited buffer.

Fig. 4. File download time in the limited buffer situation. The size of TCP proxy buffer and RLC buffer is 1 MB.

each session's RTT is followed by

$$RTT_{sp} = RTT_{s1}, RTT_{pu} = RTT_{x2} + RTT_{radio}, \quad (1)$$

where RTT_{sp} is the RTT of server to proxy and RTT_{pu} is the RTT of proxy to UE. Else if it is deployed on the SN as shown in Fig. 2b, RTT can be expressed as

$$RTT_{sp} = RTT_{s1} + RTT_{x2}, RTT_{pu} = RTT_{radio}. \quad (2)$$

The difference between the two cases is in which session the X2 interface delay is included. If the session with X2 interface delay becomes bottleneck link, X2 interface delay affects overall CWND growth rate.

Meanwhile, RWND reported to the server can be approximated as

$$RWND_{sp} \approx BUFFER_p - R \times RTT_{pu}, \quad (3)$$

where $RWND_{sp}$ is a receive window reported from TCP proxy to server, $BUFFER_p$ is a total buffer size of TCP proxy, and R is a data rate from TCP proxy to user. We consider user's RWND is enough to serve data rate from the server. Note that $R \times RTT_{pu}$ is a bandwidth delay product of both downlink and uplink, and it is an in-flight packets. For simplicity, we ignore unsent packets of Fig. 2. If TCP proxy is on the MN, long X2 interface delay reduces RWND, since RTT_{pu} includes RTT_{x2}. Otherwise, RWND of TCP proxy on the SN is irrelevant to X2 interface delay.

Since CWND grows faster with short RTT, session with short RTT between split sessions will have larger CWND. However, overall CWND depends on session with small CWND. In other words, regardless of short RTT session, session with long RTT determines overall throughput of network with TCP proxy. RWND as well as CWND should be considered, since it can regulate data rate from the server. Hence the data rate experienced by the user is determined by:

$$SWND = min\{CWND_{sp}, CWND_{pu}, RWND_{sp}\}, \quad (4)$$

where $SWND$ is a send window size of server, $CWND_{sp}$ and $CWND_{pu}$ are CWND from the server to proxy and the proxy to user respectively. In the next section, we present various simulation results to figure out the impact of TCP proxy placement in the MC enabled network.

III. Simulation

We performed simulation to figure out the impact of TCP proxy placement in the MC enabled network. We deployed TCP proxy on an MN or an SN. The RTT of each session depends on the placement of TCP proxy. We implemented all our simulations with ns-3. In our simulation, one UE is connected to the MN and the SN, and receives data from the SN only. the BS and the UE are in a fixed, line-of-sight (LOS) position. A server transmits file to the UE at 1000 Mbps to utilize high throughput of mmWave link. To prevent buffer overflow, we set a buffer size of TCP proxy equal to an RLC buffer size. We evaluate file download completion time, which is directly associated with user's QoE. Three cases of different proxy placement are simulated, without TCP proxy (w/o Proxy), deploying TCP proxy on the SN (Proxy/SN), and deploying TCP proxy on the MN (Proxy/MN). We simulated with various X2 interface delay, server delay, and buffer sizes of TCP proxy. We set edge server delay as 1 ms and remote server delay as 30 ms. The TCP proxy's buffer size is simulated for a sufficient 10 MB buffer and a limited 1 MB buffer. The detailed simulation parameters are shown in Table I.

A. *Impact of TCP proxy*

Both simulation with sufficient buffer and limited buffer show the worst file download time when there is no TCP proxy. Performance degradation factors are different in both cases. In Fig. 3, the main performance degradation factor of the sufficient buffer case is slow growth rate of CWND. In case of TCP proxy, end-to-end TCP session is split and CWND of each session grows with shorten RTT of each session. The bottleneck link determines overall throughput, and the RTT of the bottleneck link of TCP proxy is always short than that of without TCP proxy. From this, network with TCP proxy can get faster file download time than network without TCP proxy. However, as shown in Fig. 3b, if there is no difference between the RTT of the bottleneck link in TCP proxy enabled network and the RTT of network without TCP proxy, there is no significant gain. In the limited buffer case, RLC buffer overflow degrades throughput and file download time is several times slower than Proxy/SN as shown in 4. TCP proxy reports its available buffer and server regulates data rate with the reported RWND. Without TCP proxy, small RLC buffer is easily overflowed by massive data rate and server perceives it as a network congestion and drops its CWND. After CWND is dropped, it is hard to recover an original CWND but the TCP proxy can overcome the problem by placing it in an appropriate location in a mmWave network.

B. *Sufficient buffer case*

Fig. 3 shows file download time of a remote server case and an edge server case with 10 MB TCP proxy and RLC buffer. In Fig. 3a, we plot download completion time with different X2 interface delay in the remote server case. Proxy/MN has the shorter download time than other cases. When the X2 interface delay is 20 ms, Proxy/MN is about 32 percent faster than Proxy/SN. The bottleneck link of Proxy/MN and Proxy/SN is an RTT_{sp} in (1) and (2). Since only the RTT of the bottleneck link in Proxy/SN increases with X2 interface delay, Proxy/SN shows worse CWND growth rate and download completion time than Proxy/MN. Fig. 3b shows download completion time with different X2 interface delay in the edge server case and has a quite different results from the remote server case. Proxy/SN shows better performance than Proxy/MN since the bottleneck link of Proxy/MN is longer than Proxy/SN. The bottleneck link of Proxy/MN is RTT_{pu}, while that of Proxy/SN is RTT_{sp}. The results show that the file download time is faster with shorter RTT of the bottleneck link with sufficient buffer.

C. *Limited buffer case*

Fig. 4 depicts file download time of a remote server case and an edge server case with 1 MB TCP proxy and RLC buffer, which is limited buffer size compared to the previous case. With limited buffer, $RWND_{sp}$ affects rate R. We present the remote server case in Fig. 4a. Proxy/MN and Proxy/SN shows comparable results when X2 interface delay is 1 ms. However file download time of Proxy/SN is 3 times faster than Proxy/MN when X2 interface delay is greater than 5 ms. While server rate of Proxy/MN is adjusted by $RWND_{sp}$ in (3), that of Proxy/SN is determined by $CWND_{sp}$ in (4). This is because $RWND_{sp}$ becomes smaller with long RTT_{pu}. While RTT_{pu} of Proxy/MN increases with X2 interface delay, that of Proxy/SN is not affected at all. The edge server case is plotted in Fig. 4b. Proxy/SN has the fastest file download time, just like on the remote server case. But for Proxy/MN, unlike the remote server, file download time increases with the increase of X2 interface delay. While the RTT_{sp} is as short as 1 ms, the RTT_{pu} increases several times longer than 1 ms according to the increase of RTT_{x2}. If RTT_{x2} becomes longer, the TCP proxy buffer gets full faster, and if the buffer is full, the server will send the data with small $RWND_{sp}$. Hence the file download time of Proxy/MN becomes longer with X2 interface delay.

IV. Conclusion

This paper presents the details of TCP proxy in multi-connectivity enabled 5G mmWave network. File download time of TCP proxy enabled network is investigated with various server to BS delay, X2 interface delay, and buffer size of TCP proxy. We find the three important findings to understand TCP proxy in multi-connecivity enabled mmWave network. Fisrt, TCP proxy enhances file download time with fast CWND growth and preventing RLC buffer overflow. Second, if the buffer size of TCP proxy is sufficient, the RTT of bottleneck link between split sessions is an index for determining performance. Finally, with limited buffer size of TCP proxy, RWND reported to server determines the performance difference between MN placement case and SN placement case. As a future work, we plan to propose a new scheme to optimize overall TCP proxy operation in a network to support mmWave communication.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science, ICT and Future Planning (No. 2017R1E1A1A01074358)

References

[1] M. Giordani, M. Mezzavilla, S. Rangan, and M. Zorzi, "Multi-connectivity in 5G mmWave cellular networks," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2016 Mediterranean*. IEEE, 2016, pp. 1–7.

[2] S. Kang, S. Choi, G. Lee, and S. Bahk, "A Dual-Connection based handover scheme for Ultra-Dense Millimeter-Wave cellular networks," in *2019 IEEE Global Communications Conference: Mobile and Wireless Networks (Globecom2019 MWN)*, Waikoloa, USA, Dec. 2019.

[3] J. Border, M. Kojo, J. Griner, G. Montenegro, and Z. Shelby, "Performance enhancing proxies intended to mitigate link-related degradations," Tech. Rep., 2001.

[4] M. Kim, S.-W. Ko, H. Kim, S. Kim, and S.-L. Kim, "Exploiting caching for millimeter-wave tcp networks: Gain analysis and practical design," *IEEE Access*, vol. 6, pp. 69 769–69 781, 2018.

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