A transport protocol for millimeter-wave links

Hiroaki Sato and Masaki Bandai
Dept. of Information and Communication Sciences, Sophia University,
4–7 Yombancho, Chiyoda, Tokyo 102–0081, Japan
a) toriha-saiko@eagle.sophia.ac.jp

Abstract: This letter proposes a transport protocol for millimeter-wave (mm-wave) communications to shorten the time to resume communication after a mm-wave link is reconnected. The proposed method distinguishes whether a retransmission timeout occurs due to a mm-wave disconnection or network congestion. If a TCP sender detects a mm-wave disconnection, the sender keeps its retransmission timer to prevent the retransmission timer from increasing exponentially. We confirm the effectiveness of the proposed method using computer simulations.

Keywords: TCP, millimeter-wave link

Classification: Network

References

[1] T. Nitsche, C. Cordeiro, A. B. Flores, E. W. Knightly, E. Perahia, and J. C. Widmer, “IEEE 802.11ad: Directional 60GHz communication for multi-Gigabit-per-second Wi-Fi,” IEEE Commun. Mag., pp. 132–141, Dec. 2014. DOI:10.1109/MCOM.2014.6979964
[2] Y. Lim, Y. Chen, E. M. Nahum, D. Towsley, and K. Lee, “Cross-layer path management in multi-path transport protocol for mobile devices,” Proc. IEEE INFOCOM 2014, pp. 1815–1823, April 2014. DOI:10.1109/INFOCOM.2014.6848120
[3] S. K. Saha, A. Garg, and D. Koutsonikolas, “A first look at TCP performance in indoor IEEE 802.11 ad WLANS,” Proc. IEEE INFOCOM 2015, pp. 63–64, April 2015. DOI:10.1109/INFCOMW.2015.7179344
[4] Y. Oguma, R. Arai, T. Nishio, K. Yamamoto, and M. Morikura, “Proactive base station selection based on human blockage prediction using RGB-D cameras for mmWave communications,” Proc. IEEE Globecom 2015, Dec. 2015. DOI:10.1109/GLOCOM.2015.7417432
[5] H. Sato and M. Bandai, “A TCP enhancement for millimeter wave links,” Proc. ICMU 2016, Oct. 2016.
[6] “MultiPath TCP (MPTCP) implementation in ns-3,” http://github.com/mkheirkhah/mptcp.
[7] “ns3-802.11ad,” http://github.com/hanyassasa87/ns3-802.11ad.

1 Introduction

In recent years, millimeter-wave (mm-wave) technology has received much attention as future high-speed wireless access. However, mm-wave technology has two
unique physical characteristics: strong straightness and diffraction, and data communication that is disturbed easily by obstacles [1]. In general TCP, when detecting a retransmission timeout, the sender doubles the timeout value of its retransmission timer exponentially to mitigate network congestion. When a wireless-link disconnection occurs in mm-wave communication, multiple TCP segments are lost and consecutive retransmission timeouts occur. In this case, even if the mm-wave link is reconnected, the sender must wait during the long retransmission timeouts. Thus, the time to resume communication becomes long. Multipath TCP with MAC-Layer awareness (MPTCP-MA) is presented for reducing the time needed to resume communication after Wi-Fi is reconnected [2]. Although MPTCP works well with Wi-Fi, the radio characteristics of the mm-wave and Wi-Fi are different [3]. Unlike Wi-Fi, TCP throughput is not expected from the wireless signal strength of mm-wave communications. Therefore, MPTCP-MA cannot be adopted in mm-wave communications. The method in [4] uses a camera to predict a mm-wave link disconnection caused by human movement.

In this letter, we propose a transport protocol for mm-wave communications to shorten the time to resume communication after a mm-wave link is reconnected. The proposed method prevents the retransmission timer from increasing exponentially. This letter is the extended version of the poster proceeding [5]. Extended from [5], this letter evaluates the performance of the proposed method considering a realistic mm-wave environment. This letter uses the model in which the propagation delay of a mm-wave link changes due to MAC-layer retransmissions.

2 Fundamental observation

2.1 Problem when TCP works with mm-wave communications

We investigate how TCP works with mm-wave communications. The left side of Fig. 1 shows an example of the time variation of the congestion window (cwnd) of a TCP sender in mm-wave communications. In this figure, at $t_0$, the mm-wave link is disconnected. In the shaded duration, the mm-wave link continues to be disconnected. At $t_1$, the first retransmission timeout occurs, and then the sender doubles its retransmission timer (to $2T$). At $t_2$, the retransmission timeout occurs again, and then the sender doubles its retransmission timer again (to $4T$). As a result, the time to resume the communication after the mm-wave link is reconnected is $\Delta T$.

2.2 Variation of RTT in TCP with mm-wave communications

We observe how the round-trip time (RTT) changes when TCP works with mm-wave communications. The right side of Fig. 1 shows an example of the time variation of RTT for a TCP sender in mm-wave communications. In this figure, at $T_1$, the TCP sender receives Dup-ACKs, and the RTT is large due to light network congestion. At $T_2$, the retransmission timeout occurs due to heavy network congestion, and the last recorded RTT is large. At $T_3$, the mm-wave link is disconnected, and the last recorded RTT is not a large value because the wireless link disconnections are obviously not related to network congestion. Based on these observations, the proposed method distinguishes the cause of a retransmission timeout.
The proposed method consists of two operations:

1. Recording RTTs.
2. Setting the retransmission timeout value.

In Recording RTTs, when receiving an ACK, a TCP sender records the RTT value as $rtt_{last}$. When a sender receives a new ACK, $rtt_{last}$ is overwritten. Therefore, $rtt_{last}$ keeps the last recorded RTT. When a TCP sender receives Dup-ACKs, the sender checks the RTT. Only if the RTT is the minimum value among RTTs of all received Dup-ACKs, the sender records (overwrites) the RTT as $rtt_{base}$. $rtt_{base}$ keeps the minimum RTT of all received Dup-ACKs.

In Setting the retransmission timeout value, when a retransmission timeout occurs, the data sender uses the recorded RTTs to distinguish whether the timeout occurs due to a mm-wave link disconnection or network congestion. When a retransmission timeout occurs, the data sender compares $rtt_{last}$ and $\alpha \cdot rtt_{base}$, where $\alpha$ is a constant margin ($0 \leq \alpha \leq 1$). If $rtt_{last} < \alpha \cdot rtt_{base}$, the data sender determines that the cause of the retransmission timeout is disconnection of the mm-wave link. In this case, the data sender holds the value of the retransmission timer in order to reduce the time to resume communication after the mm-wave link is reconnected. On the other hand, if $rtt_{last} \geq \alpha \cdot rtt_{base}$, the data sender determines that the cause of the retransmission timeout is network congestion. In this case, the sender doubles the timeout value of its retransmission timer exponentially to mitigate network congestion. Re-setting the retransmission time is a normal operation of TCP.

The left side of Fig. 2 illustrates the variation of RTT in the proposed method. We assume $\alpha = 0.8$. At time $T_0'$, we assume that a TCP sender receives an ACK whose RTT is 30 ms. Therefore, $rtt_{last}$ is 30 ms at time $T_0'$. At time $T_1'$, the data sender receives Dup-ACKs, and renews $rtt_{base}$. We assume that the RTT of the received Dup-ACKs is 100 ms. At time $T_2'$, $rtt_{last}$ is 100 ms. At time $T_3'$, we assume that the retransmission timeout occurs due to network congestion. In this case, $rtt_{last} = 100$ ms and $\alpha \cdot rtt_{base} = 80$ ms. The data sender determines that the retransmission timeout is caused by network congestion because $rtt_{last} > \alpha \cdot rtt_{base}$. At time $T_4'$, we assume that the retransmission timeout occurs due to a mm-wave link disconnection. In this case, we assume that $rtt_{last} = 40$ ms and $\alpha \cdot rtt_{base} = 80$ ms. Because $rtt_{last} < \alpha \cdot rtt_{base}$, the data sender determines that the cause of the retransmission timeout is a mm-wave link disconnection.
The right side of Fig. 2 illustrates the variation of the congestion window of the proposed method. In this figure, at $t_0$, the mm-wave link is disconnected. In the shaded duration, the mm-wave link continues to be disconnected. At time $t_1$, we assume that a retransmission timeout occurs due to the mm-wave link disconnection, and the data sender holds its retransmission timer. At time $t_2$, $t_3$, and $t_4$, a retransmission timeout again occurs due to the mm-wave link disconnection, and the data sender holds its retransmission timer again. As a result, the time to resume the communication after the mm-wave link is reconnected is $\Delta T$. In comparison with the left side of Fig. 1, the proposed method reduces $\Delta T$. 

4 Performance evaluation

We implemented the proposed method on the network simulator ns-3 (version 3.19) [6] and used TCP Reno as the conventional method for comparison. Before the evaluation, we conducted a preliminary experiment to model the mm-wave link. Using ns3-802.11ad [7], we simulated a situation where three mobile nodes are around an AP and every node transmits data packets to the AP with TCP. Based on the simulation results, we observed that the propagation delay of a mm-wave link changed due to MAC-layer retransmissions. We modeled the characteristics probabilistically. In the model, the link delay changes every 40 ms and we denote probability as $p$. The link delay $L$ was selected randomly between 300 us and 650 us with probability $p$, and was selected randomly between 700 us and 1,550 us with probability $1 - p$.

The top of Fig. 3 shows the network topology in the simulations. All links are wired. Link 1 is a mm-wave link, which is simulated based on the preliminary result. The simulation time is 30 seconds. We simulate a mm-wave link disconnection for one second at time 25 seconds from the start of the simulation by setting a frame error rate of hundred percent in Link1. The maximum segment size is 1,400 bytes. The queue length of router 1 and 2 is 100 packets. We set the value of $\alpha$ to 0.99.

Fig. 3 shows the variation of the congestion window after 20 seconds from the start of the simulation. As shown on the left side of Fig. 3, the communication in TCP Reno resumes at about 26.4 seconds. Then, the communication in TCP Reno exponentially increases its retransmission timer to 1.6 seconds. On the other hand, on the right side of Fig. 3, the communication in the proposed method resumes at about 26 seconds. Then, the proposed method holds its retransmission timer during
the mm-wave disconnection. These results demonstrate that the proposed method resumes communication 0.4 seconds faster after the mm-wave link is reconnected. In addition, we evaluated the performance of the proposed method for various disconnection durations. The table of Fig. 3 shows the time to resume communication after the mm-wave link is reconnected in various disconnection durations, $d$.

From this table, we observe that the time to resume communication after the mm-wave link is reconnected in the proposed method is shorter than that of TCP Reno in most cases. However, as can also be seen from this table, the time to resume communication after the mm-wave link is reconnected in TCP Reno and the proposed method is the same when $d = 3$. This implies that the proposed method is occasionally not effective. The effectiveness depends on the timing of the mm-wave disconnection/resume communication and the retransmission timer. These timings sometimes become almost the same such as the case when $d = 3$. In such a case, the performances of the proposed method and TCP Reno become almost the same. Note that the proposed method can obtain at least the same performance as TCP Reno.

### 5 Conclusion

In this letter, we proposed a transport protocol for a mm-wave link to reduce the time to resume communication after the mm-wave is reconnected. We confirmed that the proposed method can reduce the time to resume communication after the mm-wave is reconnected. As future work, we will implement the proposed method on a real system and conduct experiments.

### Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 15K00136.