Poster: TCP Congestion Control based on Transmission Rate of Wireless LAN Interfaces

Akira Okada
Kyushu Institute of Technology
Fukuoka, Japan
okada.akira105@mail.kyutech.jp

Daiki Nobayashi
Kyushu Institute of Technology
Fukuoka, Japan
nova@ecs.kyutech.ac.jp

Takeshi Ikenaga
Kyushu Institute of Technology
Fukuoka, Japan
ike@ecs.kyutech.ac.jp

***Abstract*—TCP congestion control uses end-to-end delay and packet discard information to control the transmission rate. In the wireless LAN, there is a time lag between TCP congestion control and the transmission rate control of the wireless LAN interfaces, and there is no coordination between the respective controls. Thus, congestion control corresponding to the current state of the wireless network can be achieved by shortening this time lag. Therefore, this paper proposes a method to control the congestion window by detecting changes in the transmission rate of the wireless LAN interfaces. Our proposed method enables to control congestion windows size by using data link layer information, instead of using only transport layer information. This paper verified the effectiveness of the proposed method by network simulator ns-3.40.**

***Index Terms*—Cross layer Design, Congestion Control, TCP/IP, Wireless LAN**

I. Introduction

TCP congestion control algorithms classify based on packet loss, delay, or a combination of both, which can be obtained at the transport layer. Meanwhile, other layers may implement rate control depending on the communication environment. For instance, in the wireless LAN, dynamic transmission rate control mechanisms such as Auto Rate Fallback (ARF) adjust the transmission rate based on frame loss conditions. Since congestion control at the transport layer and rate control at the MAC layer operate independently, excessive control in either layer may degrade communication performance. Therefore, this paper proposes a congestion control method that coordinates with the rate control of the wireless LAN interfaces. The proposed method aims to improve throughput by appropriately controlling the congestion window at the transport layer in response to changes in the rate of the wireless LAN interfaces. This paper implements the proposed method for TCP-CUBIC and evaluates its effectiveness using the network simulator ns-3.40.

II. Related works

There are many studies using TCP in wireless environments [1] [2] [3]. Shaocheng Qu et al. [4] introduced the fuzzy sliding mode congestion control algorithm (FSMC) as a cross-layer congestion control algorithm for wireless sensor networks (WSNs). By applying the signal-to-noise ratio of the wireless channel to TCP and combining fuzzy control and sliding mode control (SMC), they designed a fuzzy sliding mode controller (FSMC). This controller adjusted the length of the buffer queue of congested nodes. The proposed FSMC effectively adapted to changes in queue length and showed excellent performance in terms of rapid convergence, lower average delay, lower packet loss rate, and higher throughput.

Hongtao Liu et al. [5] proposed a new cross-layer congestion control method for WSNs. The method performs bandwidth and delay estimation at the MAC layer of the current link and feeds it back to the transport layer via a cross-layer interaction mechanism. The transport layer at the source node alleviated network congestion by adjusting congestion windows and slow-start thresholds based on received path bandwidth and delay information and explicit congestion notifications (ECNs).

III. Proposed method

This paper proposes the TCP congestion control method based on the transmission rate control of wireless LAN interfaces. The proposed method targets IEEE802.11ac or 11ax wireless LAN interfaces. Upon detecting changes in the modulation and coding scheme (MCS) Index that shows the transmission rate of wireless LAN interfaces, the proposed method adjusts the congestion window size (cwnd) as follows:

$$cwnd = \begin{cases} \max(cwnd_t, cwnd_i) & \text{if } MCS_t > MCS_{t-1}, \\ \min(cwnd_t, cwnd_i) & \text{if } MCS_t < MCS_{t-1}. \end{cases} \tag{1}$$

MCS_t is the current MCS Index value and MCS_{t-1} is the MCS Index value before the change. where $cwnd_t$ is the current congestion window size. $cwnd_i$ is the ideal value of cwnd calculated from MCS Index as follows:

$$cwnd_i = throughput_{eff} * RTT_{e_n}/2, \tag{2}$$

where RTT_{e_n} is the moving average of n RTTs measured by TCP and $throughput_{eff}$ is the effective throughput of the wireless LAN when using the MCS Index after the change [6].

IV. Performance Evaluation

To verify the effectiveness of the proposed method, we evaluated its performance using the network simulator ns-3.40. Fig.1 shows the experimental environment. In this experiment,

979-8-3503-5171-2/24/$31.00 ©2024 IEEE

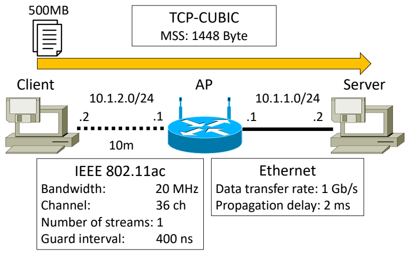
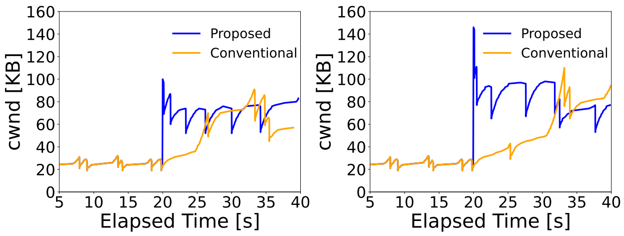


Fig. 1: Experimental environment



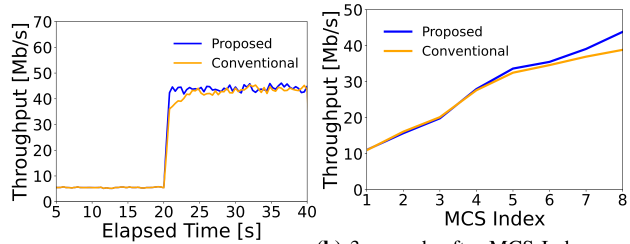
(a) MCS Index $0 \rightarrow 5$ **(b)** MCS Index $0 \rightarrow 8$

Fig. 2: The characteristics of cwnd

the client transmitted 500 MB data to the server via an access point (AP). The send buffer size and receive buffer size were set to 2MB, and maximum segment size (MSS) was set to 1448 Bytes. The client and the AP connect by IEEE 802.11ac wireless LAN, and the AP and the server connect by the wired. The bandwidth of the wireless was set to 20 MHz, the channel was set to 36ch, the number of streams for multi-input multi-output (MIMO) was set to one, a guard interval was set to 400 ns, and MCS Index for control frame transmission was set to zero. The distance between the client and the AP was 10 m. The propagation loss model was a log distance propagation loss model. The size of A-MSDU was 1498 Bytes, and the size of A-MPDU was 32 frames. The bandwidth of the wired was set to 1 Gb/s and a propagation delay of 2 ms. The queuing function adapted to each device was Tail Drop, with a queuing buffer size of 10 packets. The simulation time was 40 seconds. To reproduce the change in the transmission rate of the wireless LAN, the MCS Index for data frame transmission changed from 0 to x 20 seconds after the start of the simulation. In this environment, we investigated the characteristics of the cwnd and the throughput with and without the proposed method.

V. Simulation Results

Fig.2 shows the characteristics of cwnd as the MCS Index changes. Fig.2a shows the case where the MCS Index changed from 0 to 5. In the conventional method, the cwnd gradually increased after 20 seconds. On the other hand, in the proposed method, the cwnd increased rapidly to 100 KB after 20 seconds. After that, according to the algorithm, the cwnd operated in congestion avoidance mode in the range from 50 KB to 90 KB. The results confirm that the proposed method increases the cwnd faster than the comparison method. Fig.2b shows the case where the MCS Index changed from 0 to 8. In the same way as when the MCS Index changed from 0 to 5, the proposed method was able to increase the cwnd more rapidly than the comparison method. In the proposed method, the cwnd increased to 147 KB after 20 seconds. Thus,



(a) MCS Index $0 \rightarrow 8$ **(b)** 3 seconds after MCS Index $0 \rightarrow$ x ($1 \leq$ x ≤ 8)

Fig. 3: Throughput performance

it was confirmed that cwnd increases rapidly as the MCS Index increases with the proposed method.

Fig.3 shows the throughput performance as the MCS Index changes. Fig.3a shows the case where the MCS Index changed from 0 to 8. In the conventional method, the throughput increased to 35 Mb/s after 20 seconds. The throughput then gradually increased to 43 Mb/s in 6 seconds. On the other hand, in the proposed method, the throughput increased to 43 Mb/s after 20 seconds, and did not fluctuate significantly after that. The result was due to the increase in the amount of data transmitted as a result of the rapid increase in cwnd at Fig.2b. These results indicate that when the MCS Index changed from 0 to 8, the proposed method can quickly adapt to the transmission rate of the wireless LAN interfaces to control congestion windows and improve throughput. Fig.3b shows the throughput for 3 seconds after the MCS Index changed from 0 to x ($1 \leq$ x ≤ 8). When the MCS Index changed from 0 to 5 or more, the throughput of the proposed method increased compared to the conventional method. These results indicate that the proposed method can improve the throughput as the MCS Index increases.

VI. Conclusion

In this study, we proposed a method of controlling cwnd in conjunction with rate control of the wireless LAN interfaces, and verified its effectiveness through simulations. The performance evaluation results showed that cwnd and throughput increased quickly when the MCS Index increased, confirming the effectiveness of the proposed method. In the future, we plan to evaluate the effectiveness of the proposed method when the MCS Index decreases.

References

[1] A. Saif, et al., "CBA-TCP: A cross-layer TCP transmission rate adaptation scheme based on the compressed block ACK bitmap of IEEE 802.11n networks," EAI Endorsed Transactions on Mobile Communications and Applications, Vol.6, No.17, Jun. 2022.

[2] A. Manikandan, et al., "An Improved Cross Layer Design For TCP Optimization With Flow Control and Acknowledgment Aggregation," in Proc. of NEIeX2023, Dec. 2023.

[3] F. Zhang, et al., "Throughput-Guaranteed Cross-Layer Design for Multichannel Wireless Networks," in Proc. of ICCT2022, Nov. 2022.

[4] S. Qu, et al., "Cross-layer congestion control of wireless sensor networks based on fuzzy sliding mode control," Neural Computing and Applications, Vol.32, pp.13505–13520, Feb. 2020.

[5] H. Liu, et al., "A Novel Cross-layer Congestion Control Strategy for Wireless Sensor Networks," Journal of Networks, Vol.7, No.11, Nov. 2012.

[6] O. Sharon, et al., "MAC level Throughput comparison: 802.11ac vs. 802.11n," Physical Communication, Vol.12, pp.33–49, Sep. 2014.