LETTER

A Novel Frame Aggregation Scheduler to Solve the Head-of-Line Blocking Problem for Real-Time UDP Traffic in Aggregation-Enabled WLANs

Linjie ZHU[†,††], *Student Member*, Bin WU[†a)], Zhiwei WEI[†], *and* Yu TANG[†], *Nonmembers*

SUMMARY In this letter, a novel frame aggregation scheduler is proposed to solve the head-of-line blocking problem for real-time user datagram protocol (UDP) traffic in error-prone and aggregation-enabled wireless local area networks (WLANs). The key to the proposed scheduler is to break the restriction of in-order delivery over the WLAN. The simulation results show that the proposed scheduler can achieve high UDP goodput and low delay compared to the conventional scheduler.

key words: *frame aggregation, head-of-line blocking, UDP, WLAN*

1. Introduction

Aggregation-enabled wireless local area networks (WLANs), such as IEEE 802.11n/ac/ad/ax [1], [2], are widely adopted for last-mile access to the Internet in electronic devices (e.g., mobile phones and laptops) due to the inexpensive and flexible deployment. However, the performance of real-time user datagram protocol (UDP) traffic, such as online games and live video, on these devices is limited when the channel is noisy. In early WLAN devices, real-time UDP traffic is always encapsulated in a medium access control (MAC) layer's priority queue, as defined in IEEE 802.11e, which does not take advantage of frame aggregation, that is, the aggregate MAC protocol data unit (A-MPDU), defined in 802.11n/ac/ad/ax to improve the MAC efficiency. Few studies have focused on providing quality of service (QoS) guarantees in combination with frame aggregation. Cai et al. [3] aggregated many voice packets into the A-MPDU to achieve better performance. [4] proposed multi-QoS aggregation to prevent lower priority traffic from starvation. [5] proposed an enhanced scheme for heavily loaded best-effort traffic in aggregation-enabled WLANs. However, none of the above papers consider the head-of-line (HOL) blocking problem. HOL blocking is a phenomenon that occurs when a line of MPDUs are blocked by the first lost MPDU, leading to limited performance during A-MPDU transmission under poor channel quality. The authors in [6] proposed a mathematical model to analyze the saturation UDP goodput with the HOL blocking problem but did not identify the root cause or provide improved methods.

In this letter, we first analyze the HOL blocking phenomenon and its impact on performance during A-MPDU transmission. Furthermore, a novel frame (i.e., MPDU) aggregation scheduler is proposed to solve the HOL blocking problem for real-time UDP traffic during A-MPDU transmission in error-prone and aggregation-enabled WLANs. The proposed scheme requires the sender to make only small changes and is therefore protocol-compatible. The simulation results show that the proposed scheduler achieves 44.75% higher UDP goodput and 27.15% lower delay than the conventional scheduler with the HOL blocking problem when the UDP length is 1472 bytes and the frame error rate (FER) ranges from 0.05 to 0.80.

2. The HOL Blocking Problem during A-MPDU Transmissions

Although the A-MPDU aggregation technique boosts the performance of 802.11n/ac/ad/ax, it leads to the HOL blocking problem under noisy channel conditions. If MPDUs within the A-MPDU are lost, holes will be present in the reorder buffer of the recipient and will be prevented until the corresponding lost MPDUs are retransmitted or reach the maximum number of retries. Let X and W represent the sequence number (SN) of the first lost MPDU for the initial A-MPDU transmission and the size of the reorder buffer, respectively. To maintain in-order delivery to the recipient, the subsequent A-MPDU transmissions can aggregate only those MPDUs whose SN⩽X+W-1. In other words, subsequent aggregation is prevented by the first lost MPDU, therefore leading to the HOL blocking problem. An example of the HOL blocking problem during A-MPDU transmission, where W=64, is shown in Fig. 1.

As shown in Fig. 1, the originator first sends an A-MPDU with SN ranging from 00 to 63. During transmission, MPDUs 02 and 63 are lost due to channel noise. Under these conditions, where X=02 and W=64, the reorder buffer of the recipient has two holes (i.e., holes with SN 02 and 63) that block the reorder buffer; thus, only MPDUs 00 and 01 are delivered to the upper layer. The originator identifies the lost MPDUs in a block acknowledgment (BlockAck) reply from the recipient but can aggregate only four MPDUs with SN 02, 63, 64, and 65 in the subsequent A-MPDU transmis-

Manuscript received December 22, 2018.
Manuscript revised February 27, 2019.
Manuscript publicized March 29, 2019.

[†]The authors are with Institute of Microelectronics of the Chinese Academy of Sciences (IMECAS), Beijing, China.
[††]The author is with University of Chinese Academy of Sciences (UCAS), Beijing, China.
a) E-mail: wubin@ime.ac.cn

DOI: 10.1587/transinf.2018EDL8265

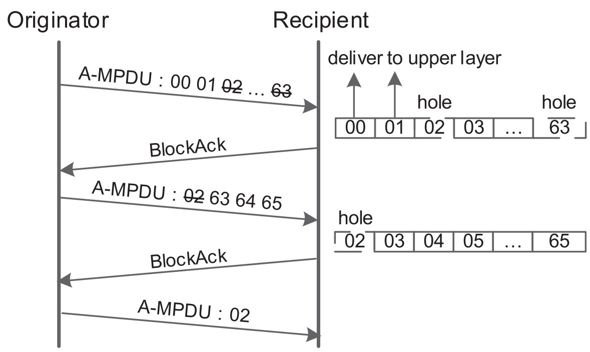

Fig. 1 An example of the HOL blocking problem during A-MPDU transmission, where W=64

sion to maintain in-order delivery. Worse still, if only the first MPDU (i.e., MPDU 02 in Fig. 1) within an A-MPDU is lost, only the lost MPDU can be transmitted in the next round.

The root cause of the HOL blocking problem during A-MPDU transmission originates from the reorder buffer operation defined in aggregation-enabled WLANs. This phenomenon severely reduces the average number of MPDUs aggregated into the A-MPDU and, thus, the performance of WLANs sharply declines. Real-time applications always use UDP traffic due to its simple and fast communication. In error-prone aggregation-enabled WLANs, real-time UDP traffic aggregated into A-MPDU will encounter HOL blocking problems, therefore leading to lower goodput and higher delay.

3. The Proposed Frame Aggregation Scheduler

We propose a novel frame aggregation scheduler whose key idea is to break the restriction of maintaining in-order delivery during A-MPDU transmission to solve the HOL blocking problem for UDP traffic delivery in aggregation-enabled WLANs.

Before we address the details of the proposed method, we first explain why breaking the restriction is reasonable. One of the main application scenarios of WLANs is last-mile access to the Internet. End-to-end UDP traffic on the Internet passes through various processing units, such as switches and routers, via different paths. Traffic transmitted via multiple paths leads to out-of-order arrival; furthermore, many middle processing units do not guarantee in-order delivery because of parallel processing [7], [8]. Consequently, regardless of whether packets are kept in order in WLANs, they are likely to be out-of-order upon reaching remote endpoints or local access points (APs). Moreover, for real-time UDP traffic, delayed arrival due to the HOL blocking problem is far worse than out-of-order arrival. Out-of-order UDP packets are usually processed by the application layer.

We divide real-time applications that use UDP as the transport layer into two categories. The characteristic of the first type of application is that there is no correlation between delivered packets. For example, the parallel remote procedure call (RPC) for distributed computing or transaction-oriented protocols such as the domain name system (DNS), etc., are insensitive to random order delivery. The other type of application is sensitive to jitter between packets, such as multimedia streaming using a leaky bucket buffer. For the first type of application, it is unnecessary to maintain in-order delivery from the MAC layer's perspective because the packets are not correlated. For the second type of application, it is still acceptable to break in-order delivery from the MAC layer's perspective under certain conditions. Let B and R represent the size of the leaky bucket buffer and the constant output rate, respectively. If we break in-order delivery from the MAC layer's perspective, the maximum gaps (i.e., the number of packets between two packets with successive SNs) in the leaky bucket buffer is $W-2$, which, in turn, requires $B \geq 1+W-2+1 = W$ for an initial playout delay to prevent jitter or loss. In addition, the average delay between two packets with successive SNs is required to be less than $\frac{1}{R}$, that is, $(W-1)(\overline{D}_{pp}+\overline{D}_{\text{int}}) \leq \frac{1}{R}$, where $\overline{D}_{pp}$ is the average delay between two continuously transmitted packets over the WLAN and $\overline{D}_{\text{int}}$ is the average delay over the Internet. In the absence of the HOL-blocking problem, $\overline{D}_{pp}$ is relatively small.

An advantage of our proposed scheduler is that it is protocol-compatible with commercial WLAN devices because it is a sender-only scheme; The proposed scheduler, which is simple but effective for solving the HOL blocking problem for UDP traffic, follows three principles:

- Every A-MPDU aggregates up to W MPDUs (i.e., frames encapsulating UDP packets in the MAC's layer) regardless of the value of X.
- If lost MPDUs are found, they are aggregated into A-MPDU if the maximum number of retry attempts has not been reached and then new MPDUs are aggregated from the MAC's priority queue.
- The SNs of the MPDUs within an A-MPDU are monotonically increasing; thus, the SNs of lost MPDUs will be replaced by a new SN in the subsequent transmission.

An example of the proposed frame aggregation scheduler, where W=64, is shown in Fig. 2.

As shown in Fig. 2, an originator sends an A-MPDU with 64 MPDUs, but MPDUs 02 and 63 are lost. According to the proposed scheduler, the SNs of the lost MPDUs (i.e., MPDU 02 and 63) are replaced by 64 and 65, and these MPDUs are aggregated into an A-MPDU; then, the new MPDUs 66 to 127, which are fetched from the MAC's queue, are aggregated into the A-MPDU. Unfortunately, the MPDU 64 is lost again during the second A-MPDU transmission. When the corresponding recipient receives MPDU 66, the hole of MPDU 02 is dropped, and MPDUs 03 to 62 are delivered to the upper layer. Finally, when the recipient receives MPDU 127, the hole of MPDU 63 is dropped. Compared to Fig. 1, the proposed scheduler can avoid the HOL-blocking problem when the first MPDU in the first

A-MPDU transmission is lost. The details of the proposed frame aggregation scheduler are as follows:

Algorithm 1: The proposed frame aggregation scheduler

```
Input: W {the size of the reorder buffer}
Initial count n (n=0) for the MPDUs aggregated in an A-MPDU to send;
For every MPDU in the retransmission queue
   If the MPDU has not reached the maximum number of retry attempts
   Replace the sequence number of the MPDU with an incremental sequence
number and recalculate the frame check sequence (FCS);
   Aggregate the modified MPDU into the A-MPDU;
   n←n+1;
   Endif
Endfor
 While n<W
 Fetch a new MPDU from the MAC's queue;
 Aggregate the MPDU into the A-MPDU;
 n←n+1;
 EndWhile
 Send the A-MPDU;
```

We extend the mathematical model in [6] and [9] to analyze the proposed scheduler. Let P_I, P_S, P_C, and e denote the possibility of idle, successful transmission, and collision transmission and the FER, respectively. The saturated UDP goodput S is calculated as:

$$S = \frac{P_S(1-e)l_{udp}E[L]}{P_I T_I + P_S T_S + P_C T_C}, \tag{1}$$

where T_I, T_S, T_C, l_{udp}, and $E[L]$ are the durations of idle, successful transmission, and collision transmission, the UDP payload length, and the average aggregation size (i.e., the average number of MPDUs aggregated into the A-MPDU). For the conventional scheduler, $E[L]$ fluctuates as e changes due to HOL blocking, whereas for the proposed scheduler, $E[L]$ is always constant.

For a particular physical layer (PHY) rate, the FER e can be expressed as a function of bit error rate (BER). If we assumed bit errors in channel are independent identically distributed, the FER e can be expressed as:

$$e = 1 - (1 - BER)^{l_{udp}+l_{oh}}, \tag{2}$$

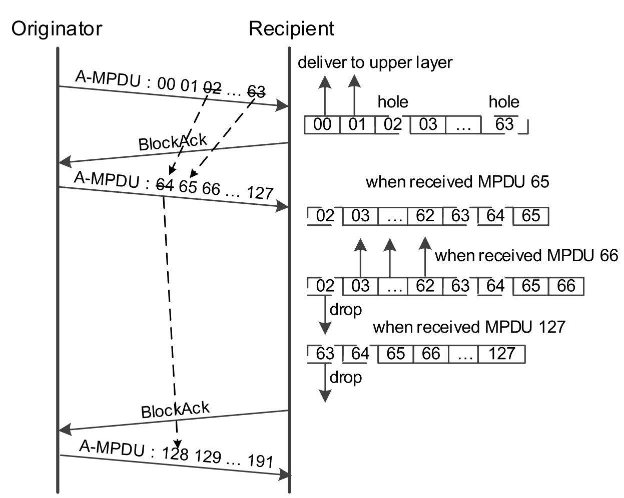

Fig. 2 An example of the proposed frame aggregation scheduler, where W=64

where l_{oh} is the extra overhead in additional to the UDP payload, such as PHY's header, MAC layer's header and so on. In 802.11 WLAN systems, PHY is responsible for the estimation of the BER, and, thus, the MAC layer can know the BER from PHY, so that the FER e can be calculated according to (2).

4. Performance Evaluation

To study the performance of the proposed frame aggregation scheduler, we implement the scheme in network simulator 3 (NS-3) [10]. In the experiments, the sender sends a saturated UDP stream with l_{udp} = 1472 bytes. All the experiments run 5 times for 100 s each time. The average of the 5 results is taken as the experimental result. The network topology for the simulation is shown in Fig. 3.

All the experiments are executed in a network where a client device with an 866 Mbps rate is accessing an 802.11ac AP. As shown in Fig. 3, there are two routers (denoted as R1 and R2) between the AP and the remote server. The link rate between the AP and R1 and between R2 and the server is fixed at 1000 Mbps, and both delays are 1 ms. The link between R1 and R2, with a rate at 1000 Mbps, is used to simulate multiple-hop routings and packet reordering over the Internet; therefore, its delay is configurable.

In the experiment, we adjusted the distance between the client and the AP to generate different BER conditions, resulting in FER ranging from 0.05 to 0.80.

We first investigate the performance of the proposed scheme on the edge of the network (i.e., WLAN). In the simulation, the client device sends saturated UDP traffic to the AP. Figure 4 shows the UDP goodput and the average delay versus FER over the WLAN.

As shown in Fig. 4, the UDP goodput of the proposed scheme is always superior to that of the conventional scheme given that the FER is constant. The conventional scheduler's goodput drops rapidly as the FER increases because $E[L]$ in (1) decreases as FER increases due to the HOL blocking problem. Because the proposed scheduler is not affected by the HOL blocking problem, $E[L]$ in (1) is always constant and the maximum allowed by the protocol (i.e., 64 in 802.11ac); thus, the goodput is linearly related to the FER. Compared to the conventional scheduler, the av-

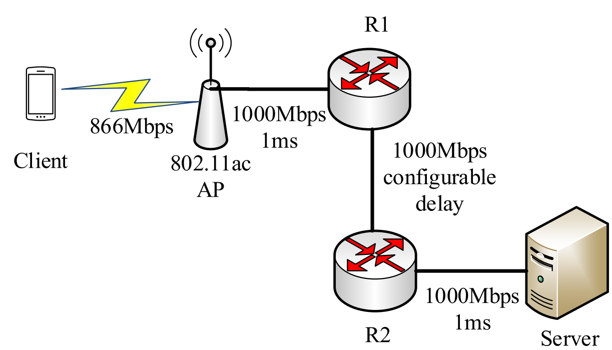

Fig. 3 The network topology for the simulation

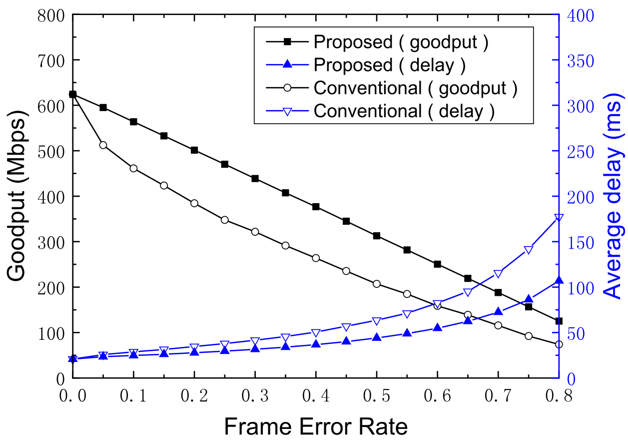

Fig. 4 UDP goodput and average delay versus FER over the WLAN

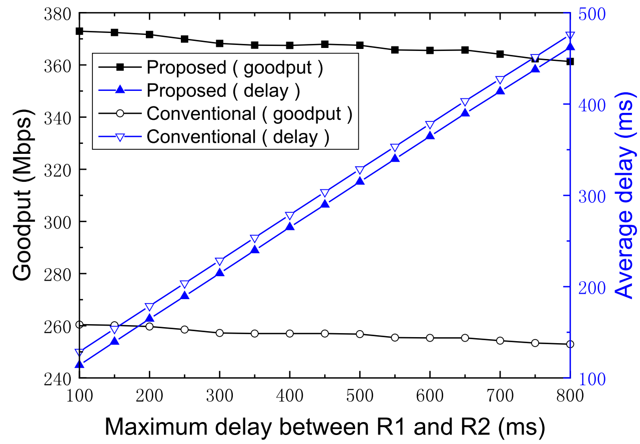

Fig. 5 UDP goodput and the average delay versus $D_{\max}$ from the client to the server when the FER is 0.40

erage speedup for the proposed scheduler is 44.75% when the FER ranges from 0.05 to 0.80. Moreover, the proposed scheme not only achieves higher goodput but also lower latency. For the conventional scheme, the impact on delay due to the HOL blocking problem increases in severity as the FER increases, therefore leading to higher average delay. By contrast, the proposed scheme is free from the HOL blocking problem and thus achieves lower delay. In our simulation, where the worst case is FER=0.80, the delays of the conventional scheme and the proposed scheme are 177 ms and 107 ms, respectively. In this case, the proposed scheduler reduces the average delay by up to 39.5%. Overall, the proposed scheduler reduces the average delay by 27.15%.

In another simulation scenario, the client sends UDP traffic to the remote server to simulate traffic over the Internet. In this simulation, the configurable delay between R1 and R2 changes every 20 ms. The delay is uniformly distributed with a range of 50 ms to $D_{\max}$, where $D_{\max}$ is the maximum delay between the link and $D_{\max} \in$ [100 ms, 800 ms]. A larger $D_{\max}$ leads to larger path changes, therefore increasing the degree of out-of-order delivery. Figure 5 shows the UDP goodput and the average delay versus $D_{\max}$ from the client to the server when the FER is 0.40.

As shown in Fig. 5, the UDP goodput of the proposed scheduler is always superior to that of the conventional scheduler given $D_{\max}$ is constant. The average speedup of the proposed scheduler is 43.1%, which is almost the same as that over WLAN. In addition, the average delays of both the proposed method and the conventional scheme linearly increase as $D_{\max}$ increases. The difference in the average delay between the conventional and proposed schemes is 13.9 ms, which is consistent with the performance over WLAN when the FER is 0.40. This result indicates that the delay is also not affected by the degree of out-out-order delivery. Overall, the proposed scheduler, which solves the HOL blocking problem, is advantageous compared to the conventional scheduler when out-of-order delivery occurs in this simulation.

5. Conclusion

In this letter, we propose a novel frame aggregation scheduler to solve the HOL blocking problem for real-time UDP traffic in aggregation-enabled WLANs when the channel is noisy. The proposed scheduler can be deployed in commercial WLAN devices to achieve high UDP goodput and low average delay for final access to the Internet.

References

[1] IEEE computer society, "Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications," IEEE Standard 802.11-2016, 2016.
[2] IEEE computer society, "Part 11: wireless LAN medium access control (MAC) and physical layer (PHY) specifications amendment enhancements for high efficiency WLAN," IEEE P802.11ax/D3.0, 2018.
[3] L.X. Cai, X. Ling., X.S. Shen, J.W. Mark, and L. Cai, "Supporting voice and video applications over IEEE 802.11n WLANs," Wirel. Netw., vol.15, no.4, pp.443–454, Sept. 2009.
[4] M.G. Sarret, J.S. Ashta, P. Mogensen, D. Catania, and A.F. Cattoni, "A multi-QoS aggregation mechanism for improved fairness in WLAN," Proc. 78th IEEE Vehicular Technol. Conf., Las Vegas, USA, pp.1–5, Sept. 2013.
[5] S. Seytnazarov and Y.-T. Kim, "QoS-aware adaptive A-MPDU aggregation scheduler for voice traffic in aggregation-enabled high throughput WLANs," IEEE Trans. Mob. Comput., vol.16, no.10, pp.2862–2875, Feb. 2017.
[6] S. Seytnazarov, J.-G. Choi, and Y.-T. Kim, "Enhanced mathematical modeling of aggregation-enabled WLANs with compressed Block-ACK," IEEE Trans. Mob. Comput., Early Access, July 2018.
[7] J.C.R. Bennett, C. Partridge, and N. Shectman, "Packet reordering is not pathological network behavior," IEEE-ACM Trans. Netw., vol.7, no.6, pp.789–798, Dec. 1999.
[8] V. Paxson, "End-to-End Internet Packet Dynamics," IEEE-ACM Trans. Netw., vol.7, no.3, pp.277–292, June 1999.
[9] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordination function," IEEE J. Sel. Areas Commun., vol.18, no.3, pp.535–547, March 2000.
[10] G.F. Riley and T.R. Henderson. "The ns-3 network simulator," Modeling and Tools for Network Simulation, pp.15–34, Springer-Verlag, Berlin, 2010.