

An Efficient MAC Layer Packet Fragmentation Scheme with Priority Queuing for Real-Time Video Streaming

Byoungheon Shin, Jalil Abdullayev, and Dongman Lee

School of Computing

KAIST

Daejeon, Korea

{bhshin, abdullayev, dlee}@kaist.ac.kr

Abstract— This paper proposes a novel priority-aware packet fragmentation extension to high throughput WLANs such as IEEE 802.11n for streaming of H264/AVC encoded videos. Unlike existing fragmentation schemes, the proposed scheme fragments IP packets based on the priority of video packets and the characteristics of MPEG-2 TS, where the original IP packet is fragmented into smaller IP packets containing fewer TS packets and prioritizes individual TS packets, allocated to an appropriate priority queue. The proposed scheme is evaluated on a testbed with various network congestion levels and channel conditions. The results show that the proposed scheme achieves higher quality of the streaming video in terms of PSNR than existing schemes as the network congestion level and the bit error rate increase.

Index Terms— Real-time video streaming, H264/AVC, MPEG-2 TS, fragmentation, prioritization.

I. INTRODUCTION

Wide deployment of mobile devices such as smartphones and tablet PCs over the past decade has led to the explosion of mobile data traffic, and video traffic takes a significant portion of overall Internet traffic. Real-time video streaming applications such as IPTV, remote desktop, and video chatting are widely used in mobile devices. Most real-time video streaming applications stream a video encoded in H264/AVC codec which is encapsulated by MPEG-2 Transport Stream (TS) and delivered by Real Time Protocol (RTP) as a transport protocol. Each MPEG-2 TS video stream packet contains seven TS packets each carrying H264/AVC data along with other synchronization data.

Nowadays, a high throughput wireless LAN (HT WLAN) such as IEEE 802.11n/ac is a common networking platform for video streaming application users. However, interference and user mobility can sharply deteriorate the performance of a wireless link. This, in turn, causes real-time video stream packets to be dropped or delivered too late to be decoded by a receiver, which can lead to glitches or freezes in the received video.

In order to solve this, several improvements have been made to the IEEE 802.11n/ac MAC layer, which can be classified into two approaches: fragmentation and packet prioritization. Fragmentation schemes [1, 2] improve throughput by dividing a long packet into smaller ones and retransmitting only the failed

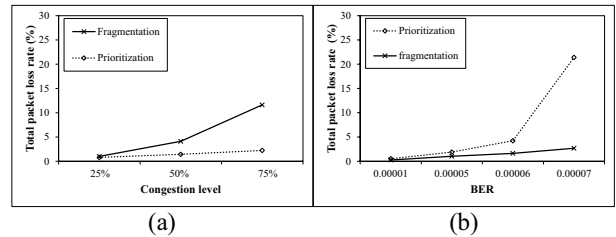


Fig. 1. The packet loss rate of fragmentation and prioritization schemes when (a) congestion level increases and (b) bit error rate (BER) increases.

fragments and thereby saving network resources. They fragment a packet without considering the characteristics of an MPEG-2 TS packet and its priority, and each fragment is considered equally important. As a result, low priority TS packets get mixed with high priority ones within one fragment. Moreover, the existing fragmentation schemes assume that every fragment should be received at the receiver side in order to assemble the original packet and passed to the upper layers. If one of fragments is not successfully received, then the whole packet associated with the fragment is dropped at the MAC layer. This causes unnecessary delay and higher loss rate for high priority TS packets. On the other hand, packet prioritization schemes [3, 4, 5] improve the robustness of video streaming by assigning packets containing important data such as I-frame data and frame headers to the high priority queue in the wireless multimedia (WMM) while packets containing B and P frame data to the low priority one. This approach has two issues. First, even if only a small portion of a packet is allocated for a higher priority frame, the rest is also put into a high priority queue. Second, since there is no fragmentation, a packet is prone to be dropped as its size gets big.

The characteristics of fragmentation and prioritization schemes are shown in Fig. 1 as the result of a simulation using QualNet. Under IEEE 802.11n environment, each node sends IP packets with the same size of MPEG-2 video streams. As the congestion level increases, a prioritization scheme outperforms the fragmentation scheme because the total number of delivered packets is smaller than the fragmentation scheme. On the other hand, when the channel condition gets worse by the increased

bit error rate, the prioritization scheme becomes more error-prone because the probability that a single packet contains erroneous bits is higher due to its bigger packet size than the fragmentation scheme. Thus, fragmentation scheme can protect more packets from bit errors while the prioritization scheme can deliver important packets faster. One can, thus, imagine that combining the merits of these two schemes would further improve the quality of real-time video streaming, especially under bad channel conditions. To the best of our knowledge, little effort has been made to combine these two schemes in an efficient way.

In this paper, we propose a novel packet fragmentation scheme for the IEEE 802.11 MAC where an IP packet containing seven MPEG-2 TS packets is fragmented in a way that the resulting fragments carry only consecutive TS packets of the same priority. Then fragments carrying high priority TS packets are mapped to the Access Category of Voice (AC_VO) as the highest priority while those carrying low priority TS packets are mapped to the Access Category of Video (AC_VI), which facilitates the fine-grained prioritization of a data. During a bad channel condition, the resulting fragments can be fragmented further to match the optimal fragment size. We devise a formula that maps the channel condition (packet error rate) to an optimal fragment size. Each fragment contains its own IP/UDP and RTP headers where the sequence number in the RTP header is modified to reflect the correct order of fragments. Thus each fragment can be independently received and passed to the upper layers. This eliminates the unnecessary delay of already received fragments and does not require any modification at the receiver side MAC layer unlike the existing fragmentation scheme. We implement the proposed scheme on the OpenWrt [6] and evaluate it on a test bed consisting of programmable APs. We compare the proposed scheme to the legacy HT WLAN, and the existing prioritization and fragmentation schemes. Experiment results show that the proposed scheme can achieve lower packet loss rate and higher PSNR than the legacy and existing schemes as the channel condition gets bad and congestion level increases.

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we discuss how to solve the issues, and in Section 4, we describe the proposed scheme. Section 5 shows our evaluations, and Section 6 concludes the paper.

II. RELATED WORK

This section explains existing packet fragmentation and prioritization schemes. The following subsections summarize fragmentation schemes and prioritization schemes.

A. Packet Fragmentation Schemes

Packet fragmentation is a part of HT WLANs. However, in HT WLANs, fragmentation is disabled when frame aggregation is performed. The reason is that an Aggregate MAC Protocol Data Unit (A-MPDU) may contain several fragments and some of these fragments might get corrupted during transmission. Thus, fragments may arrive out of order, and there is no mechanism to restore their order. Such problem does not arise

when aggregation is disabled. However, fragmentation without aggregation causes high PHY and MAC layer overhead and can reduce the overall throughput significantly [7]. Therefore, fragmentation is effective only together with aggregation.

An extension to HT WLAN has been proposed by the authors in [1] where they introduce new field to IEEE 802.11n MAC layer headers in order to enable fragmentation with aggregation (AFR). In their scheme, each MPDU has a sequence number which shows the sequence of the fragment within a frame, and a start position which shows from which byte the particular fragment starts. The receiver of the fragment has to buffer an individual fragment until all of the associated fragments are successfully received. Once every fragment is received, the receiver assembles them to form an initial frame. If one of the fragments fails, the whole frame is dropped. He et al. [8] developed a dynamic fragment size selection algorithm combined with rate adaptation to achieve optimal fragment size and data rate in order to maximize the throughput. However, none of the above mentioned schemes considers the characteristics of the H264/AVC over MPEG-2 TS video streaming.

B. Packet Prioritization Schemes

Because the inherent resilience mechanism against packet errors in the H264/AVC protocol is not enough for robust video streaming in a high packet loss rate over WLANs, many cross-layer packet prioritization approaches have been proposed.

There are several existing approaches [3, 4, 5, 9] where the network abstraction layer (NAL) header information of H.264/AVC, such as `Nal_Ref_Idc` (NRI), is mapped to the IEEE 802.11e access categories (ACs) to provide highly error-resilient transmission of video. Ksentini et al. [3] proposed a frame prioritization scheme where H264 slices are mapped to different IEEE 802.11e Enhanced Distributed Channel Access (EDCA) queues based on each slice priority. Each AC has different `CWmin`, `CWmax`, Arbitration Inter-Frame Spacing (AIFS), Distributed Coordination Function Inter-frame Space (DIFS) and retransmission count (RTC) values. They show that such setting can achieve high resiliency for important video slices during streaming video. Go et al. [4] extended it to MPEG-2 TS and prioritized a video stream on a packet level. Their main contribution is to give higher priority to slice headers thereby achieving higher packet decoding rate. Yao et al. [5] propose a dynamic mapping algorithm where each frame is mapped to a queue not only based on slice priority but also queue occupancy. Authors in [9] propose a service differentiation for streaming a video encoded using the extended profile over IEEE 802.11 WLANs. The extended profile offers a partitioning feature that divides video data into three sets with different levels of importance. However, the extended profile is not used widely in real life.

Another group of existing schemes is the frame prioritization schemes that manage the transmission rates and the delays of video streaming [10-13]. In [10], the authors propose a mechanism for robust video streaming under short-term channel bandwidth fluctuations. The cross-layer mechanism consists of slice classification at application layer, dynamic packet selective transmission (DPST) at MAC layer, and channel condition

prediction at physical layer. Wang et al [11] develop a QoS-based IEEE 802.11e MAC layer protocol taking into account the delay and importance of a video packet. The packets that are received too late are considered as low priority and dropped in the network. The available bandwidth is also considered to drop lower priority packets. Zhang et al [12] present a cross-layer design of transmitting scalable video streams from a base station to multiple clients over a shared fading wireless network by jointly considering the application layer information and the wireless channel conditions to perform priority-based packet scheduling. In [13], an EDCA-based prioritization scheme is extended to guarantee the throughput of the packets allocated to a high priority queue by optimizing contention window size under saturated network conditions.

In summary, little efforts have been made to do fine-grained fragmentation which can cooperate with existing prioritization schemes for real-time video streaming.

III. DESIGN CONSIDERATIONS

A. Separation of TS Packet Priorities

The priorities of the TS packets within an IP packet can be different. Some of the TS packets may contain important header files or I-frame data which is essential to the video decoding process. Thus they should be treated differently. In existing prioritization schemes an IP packet is considered to be of a high priority if there exist just one TS packet with high priority. This results in inefficiency since low priority TS packets get mixed with high priority TS packets and end up consuming scarce network resources. The fragmentation schemes on the other hand slice the fragments without considering the TS packet priorities. Thus a portion of the high priority packets get mixed with the low priority packets and thus it becomes hard for the prioritization scheme to parse them. Therefore, the fragment size should be chosen so that each fragment contains TS packets of the same priority. This would enable a fine grained prioritization of the fragments later.

B. Putting Necessary Headers to Fragmented Packets

The existing fragmentation scheme requires modification to the MAC layer at the receiver side because the current IEEE 802.11n/ac standards do not have any mechanism for preserving the order of the packets at the receiver side. This creates a deployment issue because all the mobile devices must be updated with the new MAC layer to be compatible with the new fragmentation scheme.

In contrast, if all the modification is done only at the sender side MAC, any access point vendor can optimize only the AP and have superior performance. Furthermore, in existing fragmentation schemes, all the fragments should be received at the MAC layer of the receiver so that the original packet is constructed and passed to the upper layer. If one of the fragments is failed then the whole packet is dropped. This is inefficient for obvious reasons. Thus, to fix it, we need another mechanism to restore the correct order of fragments and to make it appear as a normal packet at the receiver side. To do so, we can utilize the sequence number field in the RTP header. In addition, MAC, IP,

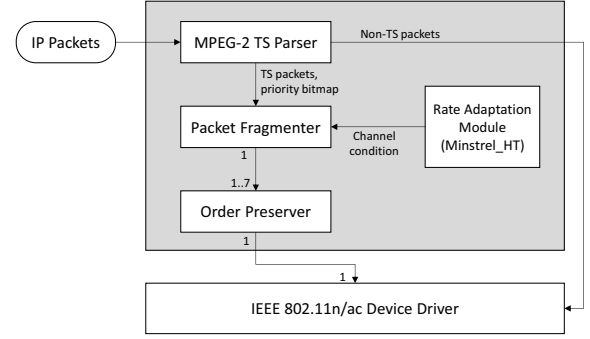


Fig. 2 The architectural diagram of the proposed scheme.

and UDP headers should be kept for each fragment so that it appears as a normal independent packet.

C. Tradeoff between Overhead and Throughput

Because fragmentation divides one packet into two or more packets, it increases the portion of header in the total number of bits transmitted for the same amount of data. When the wireless channel gets erroneous (i.e. a higher bit error rate), fragmented packets are more robust to packet loss than one packet because only fragments with errors are retransmitted instead of sending the whole data. However, when the channel condition is good enough to deliver all packets without bit errors, the increased amount of header leads to the degradation of goodput. In this case, it is better to transmit a packet without fragmentation in order to save the amount of duplicate headers. Thus, there is a tradeoff between the header overhead and the robustness of data, and the fragmentation scheme should determine the appropriate fragment size based on channel condition, i.e., bit error rate. Network monitoring is necessary to get continuous channel condition information in real-time. Although we can manually calculate the channel condition using state variables in the MAC layer, it would generate additional processing overhead in the MAC protocol. Since existing rate adaptation schemes in the MAC layer such as Minstrel_HT select an optimal rate based on channel condition, it would rather use the contextual information of rate adaptation algorithms to prevent duplicate actions on monitoring channel conditions.

IV. PROPOSED SCHEME

This section explains the proposed packet fragmentation scheme in detail. The overall architecture of the proposed scheme is shown in Fig. 2. The MPEG-2 TS Parser firstly classifies all outgoing IP packets into MPEG-2 TS packets and non-TS packets and parses each TS packet with priority bitmap information. The Packet Fragmenter fragments the TS packet into several smaller TS packets based on the priority bitmap information provided by the MPEG-2 TS Parser. On deciding the optimal fragment size, it exploits channel condition information from the existing rate adaptation module. The Order Preserver is in charge of preserving the order of fragmented packets by applying appropriate IP/UDP/RTP headers.

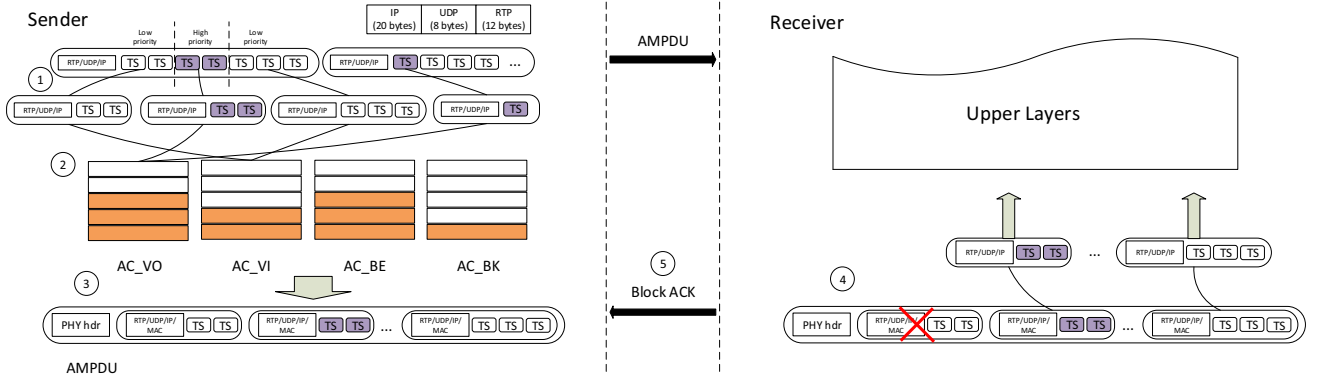


Fig. 3 A dynamic packet fragmentation and prioritization scheme process overview which consists of five mains steps. (1) A packet is fragmented, (2) Each fragment is assigned to a WMM queue depending on its priority, (3) Frame aggregation is performed on the fragments, (4) The fragments are sent to the receiver and failed fragments are dropped, and (5) A Block ACK is sent back to the sender.

A. Overview

Figure 3 shows the overall process (five steps) of the proposed scheme. Note that according to the IP datagram standard [14], an IP packet can contain up to seven TS packets. Each TS packet is 188 bytes long, and thus the data length of the IP packet containing seven TS packets is 1316 bytes. First, each packet is parsed to get the priorities of the TS packets within an IP packet. Based on TS packet priorities, an IP packet is fragmented into smaller IP packets each containing fewer TS packets. After that sender calculates an optimal fragment size and fragments that are larger than the optimal fragment size are further fragmented. Second, each fragment is assigned to one of WMM queues based on packet priority. Third, fragments in the same WMM queue are dequeued and aggregated into an A-MPDU and transmitted to the receiver. Fourth, a receiver checks each packet in an A-MPDU and drops corrupted packets. Successfully received packets are passed to the upper layers, while the failed ones are dropped. The receiver sends a Block ACK to the transmitter, which is a bitmap showing which packets are successfully received and which ones are failed. Finally, using the success/failure rate information the sender recalculates the optimal fragment size. The following subsections explain each step in detail.

B. MPEG-2 TS Packet Structure

This section explains the structure of a MPEG-2 TS packet as shown in Fig. 4 and how to parse and assign priorities to TS packets. Each TS packet consists of the 4-byte header and the 184-byte payload. The TS header starts with a sync byte which is used to detect a TS packet. The payload unit start indicator (PUSI) shows whether a TS packet contains the start of a new frame. The packet identifier (PID) field shows the ID of a TS stream. If PID is zero, then a TS packet contains the program map table (PMT), which is a list of PIDs. A TS packet may optionally contain an adaptation field. The payload of a TS packet contains a packet elementary stream (PES) that contains a single H264/AVC frame. The PES header is parsed to retrieve a stream ID and a packet length. The stream ID is used to detect

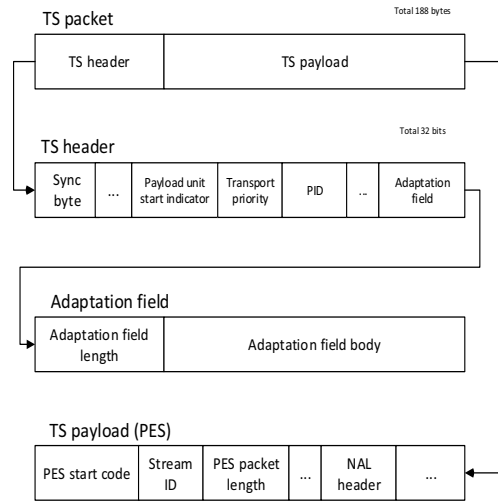


Fig. 4 MPEG-2 TS Packet Structure.

```

next_ts_packet_priority = null
func parse_ts_packet(TSPacket TS):
begin
    if TS.PID == 0 then TS.priority = High
    else if TS.streamID is audio then
        TS.priority = High
    else
        begin
            if TS.PUSI == 1 then
                begin
                    else if TS.streamID is video then
                        if TS.NRI is high priority then
                            next_ts_packet_priority = High
                        else next_ts_packet_priority = Low
                    TS.priority = High
                end
            else TS.priority = next_ts_packet_priority
        end
    end
end

```

Fig. 5 TS packet parsing and priority allocation algorithm.

whether the PES payload is an audio or a video stream. Table 1 shows the range of values and the corresponding payload type. Each H264/AVC frame inside PES payload contains the NAL header containing the NRI field which is used to get the priority of the frame. Table 2 shows the NRI field to frame type mapping.

C. MPEG-2 TS Packet Parsing and Priority Assignment

Figure 5 shows TS packet parsing and priority allocation algorithm. First we check PID; if it is zero, then we assign high priority to the TS packet. If the PID is not zero, then we check the stream ID field of PES header. If it corresponds to an audio stream then we assign a high priority to the current TS packet and skip to the next one. If stream ID corresponds to a video stream then we check the PUSI field. If it is set to one, it means a start of a new frame and the TS packet contains a video frame header and we assign high priority to the current TS packet. Then we check the NRI field in the NAL header to determine a type of a video frame. We store the type of the frame in a global variable in order to assign priorities to the subsequent TS packets carrying frame data and skip to the next TS packet. If the PUSI is set to zero, then it means that the current TS packet contains frame data. In this case we check the global variable to determine a type of a frame. Then we assign the priority of the current TS packet according to Table 3.

D. Dynamic Fragmentation and Prioritization

This section describes when and how fragmentation is carried out. We first explain how we devise a formula that can find an optimal fragment size for the current network conditions. Then we describe an algorithm that uses this formula to carry out packet fragmentation.

In order to find out an optimal fragment size, we use packet error rate, $P_e = \frac{N_{failed}}{N_{total}}$, which is a ratio of the number of unsuccessfully transmitted packets to the total transmitted packets within an A-MPDU. The relationship between BER, and P_e can be formulated as follows:

$$P_e = 1 - (1 - BER)^N \quad (1),$$

where N is a length of the single packet within an A-MPDU. From above formula, we can find an estimated BER:

$$BER_{est} = 1 - (1 - P_e)^{\frac{1}{N}} \quad (2).$$

The amount of useful data (without network layer headers) transmitted in a single transmission is

$$N_{data} = (N_{opt} - N_{overhead}) * (1 - P_e) \quad (3).$$

By substituting P_e from (1) in (2), we get the following formula for N_{data} :

$$N_{data} = (N_{opt} - N_{overhead}) * (1 - BER_{est})^N,$$

where $N_{overhead}$ is the number of bits consumed by network headers. By taking derivative and applying simple arithmetic we can show that the value of N_{opt} that maximize N_{data} can be calculated by the following formula:

$$N_{opt} = \frac{1 + N_{overhead} * \ln(1 - BER_{est})}{\ln(1 - BER_{est})} \quad (4).$$

By using the definition of BER_{est} from (2) in (4), we can obtain a final formula:

TABLE I. STREAM ID TO STREAM TYPE MAPPING

Stream ID	Stream type
1011 1101 0xBD	Private stream 1 (non MPEG audio, subpictures)
1011 1110 0xBE	Padding stream
1011 1111 0xBF	Private stream 2 (navigation data)
110x xxxx 0xC0 - 0xDF	MPEG-1 or MPEG-2 audio stream number x xxxx
1110 xxxx 0xE0 - 0xEF	MPEG-1 or MPEG-2 video stream number xxxx

Table II. NRI FIELD TO NAL DATA MAPPING

NRI	NAL Data
3	I-frame, SPS, PPS
2	P-frame, B-frame
0	B-frame

Table III. PRIORITIES BASED ON TS PACKET TYPE
TABLE III. PRIORITIES BASED ON TS PACKET TYPE

Priority	TS Packet Type
High	SPS, PPS, PAT, PMT, I-frame header, I-frame data, P-frame header, B-frame header, audio
Low	P-frame data, B-frame data

$$N_{opt} = \frac{N + N_{overhead} * \ln(1 - P_e)}{\ln(1 - P_e)} \quad (5).$$

We fragment a packet based on the priorities of its TS packets. Consecutive TS packets that have the same priority are grouped together. Then, we check the optimal fragment size using (5) and if any of the fragments are larger than optimal fragment size then it is further fragmented using a recursive formula defined below:

$$Fr(P_n, N_{opt}) = \begin{cases} P_n, & \text{if } (n * 188) + overhead < N \text{ or } n == 1 \\ Fr(P_{\lfloor \frac{n}{2} \rfloor}, N_{opt}), Fr(P_{\lceil \frac{n}{2} \rceil}, N_{opt}), & \text{if } (n * 188) + overhead > N \end{cases} \quad (6),$$

where P_n is an IP packet containing n TS packets. Thus, Fr is a recursive function that fragments a packet until each fragment is smaller than the optimal fragment size. After fragmentation, the fragments are mapped to one of the WMM queues. Fragments that carry high priority data are mapped to AC_VO and fragments that carry low priority data are mapped to AC_VI. Thus, we can ensure that high priority data get more chance of being transmitted within a reasonable delay limit.

E. Preserving TS Packet Orders

When a packet is fragmented, each fragment may arrive out of order at the receiver. However, a video decoder can reorder the packets based on RTP sequence field [15]. Therefore, an RTP header is attached to each fragment to enable a correct reordering. All the fields within the RTP header of each fragment are identical to those of the original packet except the RTP sequence field. The RTP sequence field of each fragment is calculated as below:

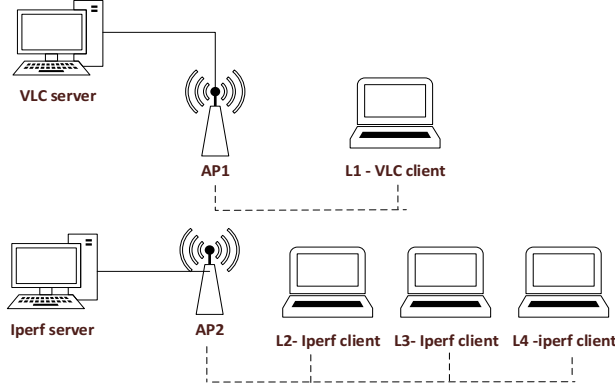


Fig. 6. Experimental settings with two access points and four clients.

$$Seq_{new} = ((Seq_{current} - Seq_{initial}) * 7) \bmod(2^{16}) + TS_{offset} \quad (7),$$

where $Seq_{initial}$ is the sequence number of the first RTP packet (RTP packets start from a random number for security reasons), $Seq_{current}$ is the sequence number of the original packet and TS_{offset} is an offset of a TS packet from which the current fragment starts. The value of TS_{offset} should be less than seven because the original IP packet contains only seven TS packets.

V. EVALUATION

In order to verify the effectiveness of our proposed scheme, we compare our scheme against the existing prioritization scheme [4], the existing fragmentation scheme [1], and the legacy IEEE802.11n MAC without prioritization and fragmentation. As evaluation metrics, we measure high and low priority TS packet loss rate, and video PSNR. We first explain experiment settings in the first subsection and summarize the experiment results under various conditions in the following subsections.

A. Experiment Settings

We implement the proposed scheme as well as the existing fragmentation and prioritization schemes on an OpenWrt, an open-source operating system for wireless routers with the version 15.05 Chaos Calmer [6]. To run the proposed scheme with OpenWrt, we use an access point Buffalo WZR-600HP which uses Qualcomm Atheros ath9k as its IEEE 802.11n MAC chipset.

We build an experimental testbed which consists of two access points (AP1, AP2) and four laptops (L1, L2, L3, and L4) as shown in Fig. 6. Both AP1 and AP2 are tuned to the channel 36 (5.180 GHz), and the channel width is 20 MHz. The reason why we use a wireless channel on 5 GHz is that there are too many access points running on channels within 2.4 GHz nearby our testbed which may show unexpected interference during experiments. The bitrate is set constant for both APs with Modulation and Coding Scheme (MCS) index 3 (26 Mbits/sec). The Transmission power is set to 0 dB (1mW) for both APs. The laptop L1 is connected to the AP1 and used for receiving a video stream from a remote server, and laptops L2, L3, and L4 are

TABLE IV. EXPERIMENT PARAMETERS

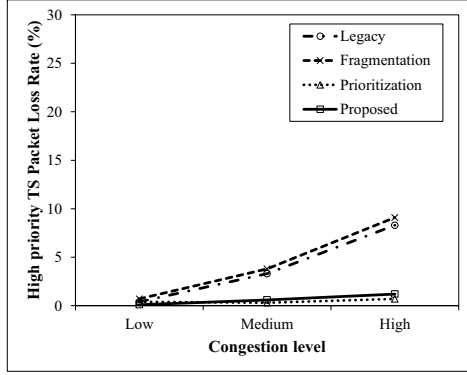
Operating System	OpenWrt 15.05
Device model	Buffalo WZR-600DHP
Chipset	Qualcomm Atheros ath9k
Wireless MAC Protocol	IEEE 802.11n
Wireless channel	36 (5.180 GHz)
Data rate	26 Mbits/s
Transmission power	0 dB
Congestion data rate	30 Mbits/s (per node)
Congestion levels	Low (1 node), medium (2 nodes), high (3 nodes)
Channel conditions	Good ($BER\ 7.7 \times 10^{-5}$), bad (6.5×10^{-4})

connected to the AP2 for generating network congestion. We use VLC [16] for streaming video contents and *iperf* [17] for generating CBR congestion traffic. The data rate of each CBR traffic is set to 30 Mbits/sec. The reason why the data rate of congestion traffic in the application layer is higher than the data rate set up at the network interface is to create a highly congested situation so that packets are rapidly enqueued. Each packet is marked with a number, and its transmission and reception times are recorded in order to estimate the packet delay (by subtracting reception time from transmission time). We use *ntptime* to synchronize times between the video server (remote server) and the client laptop (L1). Packets whose delay exceeds the delay threshold of 300ms are discarded.

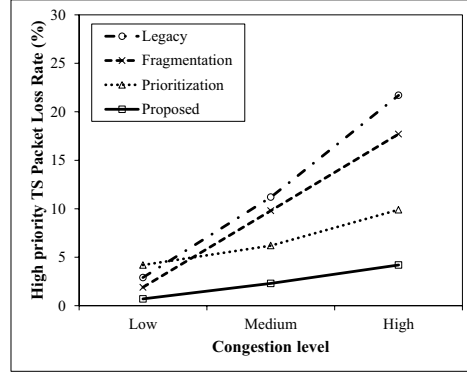
We perform experiments in different congestion levels and channel conditions. Congestion level is controlled by varying the number of contending devices, namely L2, L3, and L4, while channel condition is controlled by increasing the distance between L1 and AP1. We classify congestion levels into low, medium and high, which correspond to one, two, and three contending devices respectively. The channel conditions are classified into good and bad, which correspond to average BER values of 7.7×10^{-5} and 6.5×10^{-4} between the sender (AP1) and the receiver (L1) respectively. The BER values were obtained from the average packet error rate of 20% and 80% in the PHY layer when the average packet size is 360 bytes. We use a sample video file with the title of "Big Buck Bunny" [18], and the resolution of a video is 864x480. The streaming duration is 20 seconds for each experiment. Important experiment parameters are summarized in Table 4.

B. TS Packet Loss Measurements

Figures 7(a) and 7(b) show the loss rate of high priority TS packets for various congestion levels and channel conditions, and Figs. 8(a) and 8(b) show the loss rate of low priority TS packets for various congestion levels and channel conditions. The total packet loss rates including both low and high priority TS packets are summarized in Figs. 9(a) and 9(b). As the congestion level increases, the prioritization scheme can preserve high priority packets better than legacy and the fragmentation schemes. This is because the prioritization scheme allocate video packets to higher priority queues

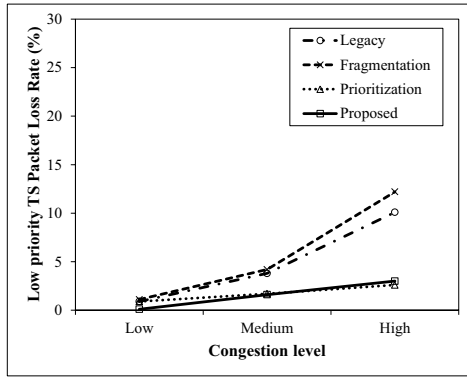


(a)

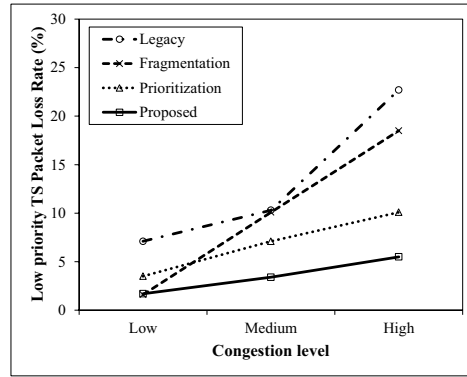


(b)

Fig. 7. Loss rates of high priority TS packets at different congestion levels: (a) a good channel condition, (b) a bad channel condition.

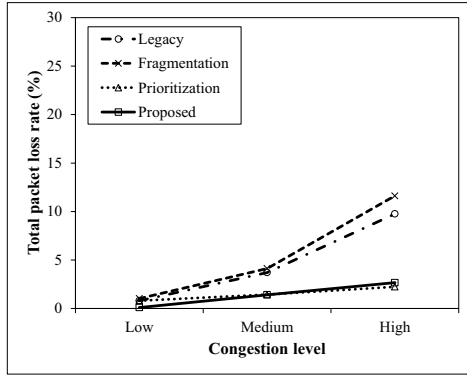


(a)

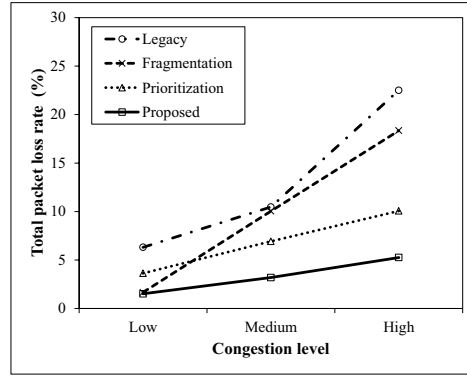


(b)

Fig. 8. Loss rates of low priority TS packets at different congestion levels: (a) a good channel condition and (b) a bad channel condition.



(a)



(b)

Fig. 9. Total TS packet loss rate at different congestion levels: (a) a good channel condition and (b) a bad channel condition.

(AC_VO and AC_VI) than congestion traffic packets which are allocated to AC_BE, and higher priority queues take a higher chance of accessing the wireless channel with lower contention window sizes.

If the channel condition is good with a low BER, the proposed scheme shows almost similar packet loss rate with the prioritization scheme because packets are rarely fragmented. On the other hand, when the channel condition gets worse under a low congestion, the prioritization scheme shows higher loss rate

of TS packets than the fragmentation scheme. This is because a smaller packet size means higher probability of successful packet transmission. Since the proposed scheme also fragments packets under bad channel conditions, it reduces up to 4% of the TS packet loss rate than the existing prioritization scheme. Thus, the prioritization scheme in general outperforms the fragmentation scheme under a congested network while the fragmentation scheme shows better performance under a bad channel condition in an uncongested case. Consequently, the

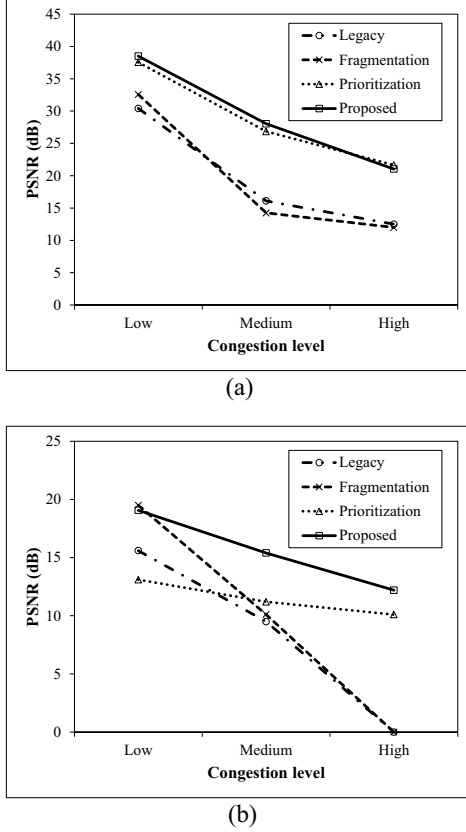


Fig. 10. The PSNR of a sample video at different congestion levels: (a) a good channel condition, (b) a bad channel condition.

proposed scheme outperforms all existing schemes because it can fragment packets and allocate different priorities to fragments based on the channel condition.

In general, the results of high priority TS packets and low priority TS packets are not much different because both packets are at least considered as video packets which are allocated to the AC_VI queue. We can see that low priority TS packets get less chances to access the channel when congestion level is higher.

If multiple video streams coexist within a network, the proposed scheme will allocate all MPEG-2 TS packets to higher priority queues than other types of traffic. One can imagine that there will be situations that the highest priority queue (AC_VO) becomes full due to congestion. This can happen if there are too many video streams and their total data rate exceeds the data rate of the wireless channel. It will become worse if the wireless channel is interfered by environmental issues. Note that the proposed scheme can work properly when the wireless capacity is bigger than the total amount of coexisting video streams.

C. PSNR Measurements

We measure the peak signal-to-noise ratio (PSNR) of received video streams for all schemes. PSNR is a common quality metric of lossy compression videos and images [19]. Figures 10(a) and 10(b) show the change of PSNR values in dB for various congestion levels when the channel condition is good

and bad, respectively. Note that the PSNR value of zero means that the video is unplayable. In a good channel condition, the proposed scheme and the prioritization scheme achieve better results than the other schemes. This is because they not only have lower packet loss rate in high congestions but also prioritize higher priority packets which are important for final video quality.

When the channel condition becomes bad, the overall PSNR values of all schemes gets reduced due to more packet drops. However, the existing prioritization scheme achieves lower PSNR due to the large number of packet drops. We can also see that the existing fragmentation scheme and the legacy scheme show very high packet drop rate under the bad channel condition, and thus the video becomes unplayable when both the bit error rate and the network congestion increases. Thus, the proposed scheme which conducts adaptive fragmentation with priority based queuing can achieve higher PSNR in various network congestion levels and bit error rates than the existing approaches. Moreover, note that the relation between PSNR and packet loss rate is not linear. As the packet loss rate increases, successive drop in PSNR gets lower.

VI. CONCLUSION

In this paper, we propose an efficient fragmentation scheme leveraging prioritization for high throughput WLANs to improve the performance of live video streaming. The main advantage of the proposed scheme is that it can achieve higher performance in case of high network congestion and bad channel conditions. Another advantage of the proposed scheme is that it does not require any modification at the receiver side like the existing fragmentation scheme. It makes the deployment of the proposed scheme much easier. Furthermore, although we focus on MPEG-2 TS protocol due to its popularity, the idea can be extended to any other video stream carried over an RTP. As a future work, we plan to investigate the impact of rate adaptation on the proposed scheme.

ACKNOWLEDGEMENT

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.B0717-16-0034, Versatile Network System Architecture for Multi-dimensional Diversity).

REFERENCES

- [1] Li et al., "Aggregation with Fragment Retransmission for Very High-Speed WLANs," *IEEE/ACM Transactions on Networking*, vol. 17, no. 2, pp. 591–604, Apr. 2009.
- [2] Sidelnikov, Alexey, Jeongyun Yu, and Sunghyun Choi. "Fragmentation/aggregation scheme for throughput enhancement of IEEE 802.11 n WLAN," *IEEE Network*, 2002.
- [3] Ksentini, Adlen, Mohamed Naimi, and Abdelhak Guéroui. "Toward an improvement of H. 264 video transmission over IEEE 802.11e through a cross-layer architecture," *Communications Magazine, IEEE*, vol. 44, no. 1, pp. 107–114, 2006.
- [4] Go, Kyungmin, et al. "Cross-layer packet prioritization for error-resilient transmission of IPTV system over wireless network," *Proceedings of the 5th ACM Multimedia Systems Conference*. ACM, 2014.

- [5] Yao, Xin-Wei, et al. "Ipb-frame adaptive mapping mechanism for video transmission over IEEE 802.11e WLANs," ACM SIGCOMM Computer Communication Review, vol. 44, no. 2, pp. 5-12, 2014.
- [6] OpenWrt. <https://openwrt.org/>
- [7] Abdullayev Jalil, Byoungheon Shin, and Dongman Lee. "A Dynamic Packet Fragmentation Extension to High Throughput WLANs for Real-Time H264/AVC Video Streaming." The 10th International Conference on Future Internet, ACM, Jun. 2015.
- [8] He, Xin, Frank Y. Li, and Jiaru Lin. "Link adaptation with combined optimal frame size and rate selection in error-prone 802.11 n networks." Wireless Communication Systems, ISWCS'08, IEEE International Symposium on, 2008.
- [9] Yaser, P.F., Panos, N., and Hussein, A. Efficient transmission of H.264 video over multirate IEEE 802.11e WLANs. EURASIP Journal on Wireless Communications and Networking, vol. 2008, no. 11, Jan. 2008.
- [10] Connie, A.T., Panos, N., Yaser, P.F., and Leung, V.C.M. Sctp-based transmission of data-partitioned h.264 video. In Proc. of the 4th ACM workshop on Wireless multi-media networking and performance modeling (WMuNep), pp. 32-36, 2008.
- [11] Srinivasan, S.K., Vahabzadeh-Hagh, J., and Reisslein, M. The effects of priority levels and buffering on the statistical multiplexing of single-layer H.264/AVC and SVC encoded video streams. IEEE Transactions on Broadcasting, vol. 56, no. 3, pp. 281-287, Sep. 2010.
- [12] Haghani, E., Parekh, S., Calin, D., Kim, E., and Ansari, N. A quality-driven cross-layer solution for MPEG video streaming over WiMAX networks. IEEE Transactions on Multimedia, vol.11, no.6, pp. 1140-1147, Oct. 2009.
- [13] Qinglin Zhao, Taka Sakurai, Jiguo Yu, and Limin Sun. "Maximizing the stable throughput of high-priority traffic for wireless cyber-physical systems." EURASIP Journal on Wireless Communications and Networking, vol. 2016, no.53, pp. 1-12, Feb. 2016.
- [14] A Standard for the Transmission of IP Datagrams over Ethernet Networks. <http://tools.ietf.org/html/rfc894>
- [15] RTP: A Transport Protocol for Real-Time Applications. <https://tools.ietf.org/html/rfc3550>
- [16] VideoLAN VLC. <http://www.videolan.org/vlc/index.html>
- [17] iPerf. <https://iperf.fr/>
- [18] Big Buck Bunny. <https://peach.blender.org/>
- [19] PSNR (Peak signal-to-noise ratio). <http://www.ni.com/white-paper/13306/en/>