A Hierarchical Packet Pre-Dropping Approach for Improved MPEG-4 Video Transmission over IEEE 802.11e Networks

Chih-Heng Ke
Department of Computer Science and Information Engineering, National Quemoy University
Kinmen, Taiwan
smallko@gmail.com

Chong-Yi Yang, Jiann-Liang Chen
Department of Electrical Engineering, National Taiwan University of Science and Technology
Taipei, Taiwan
chongyi.yang@gmail.com, lchen@mail.ntust.edu.tw

Abstract—**This paper develops a hierarchical packet pre-dropping (HPPD) approach that exploits different methods to process I (intra-coded)/P (predictive-coded) /B (bidirectionally predictive-coded) video frame packets to improve the video transmission quality over IEEE 802.11e Wireless Local Area Networks (WLANs). HPPD refers to the dropping of arriving packets if the preceding video frame packets in the same Group of Pictures (GOP) have been dropped. The simulation results demonstrate that, in terms of PSNR, the proposed scheme outperforms exiting schemes such as EDCA (Enhanced Distributed Channel Access) and adaptive mapping and novel mapping mechanism, suggesting that the resource of access channel in WLANs is not wasted in the transmission of useless video data.**

Keywords- MPEG-4; 802.11e; video transmission quality; packet pre-dropping

I. INTRODUCTION

The wireless local area networks (WLANs) has rapidly developed in the recent years because of mobility support, low cost and simplicity of deployment. The multimedia applications in WLANs are becoming increasingly popular, especially for video transmission. IEEE 802.11e has been introduced to support the QoS of multimedia transmission in WLANs, which defines a contention-based channel access method called enhanced distributed channel access (EDCA). The EDCA adopts four access categories to support QoS in the MAC layer. Four access categories were defined as AC_VO, AC_VI, AC_BE, and AC_BK (voice, video, best effort, and background traffics, respectively).

However, no differentiates between different types of video frames in the EDCA, and all packets of video frames of different types are treated equally and mapped into AC_VI. The mechanism would lead to the overflow of the AC_VI quickly. Hence, the loss of some packets of important video frames may reduce the quality of reconstructed data, and even cause video recovery to fail [1]. Hence, several efficient mechanisms use cross-layer designs base on application and MAC layer features [2][3]. Some researchers have proposed mechanisms for supporting differential service according to the significance of video frames.

However, the quality of the video transmission may still be limited due to the transmission of information that becomes useless to the receiver. In our previous work, the mapping mechanisms only consider the importance of video frames, the traffic loads of each queue, and the actions taken on previous packets of the same video frame.

Accordingly, this work developed a hierarchical packet pre-dropping approach to allocate the video frame packets to the access category queues, based on the action that was taken on packet of the dependent video frame that was transmitted previously.

This paper is an extended version of our previous work [4][5] and is organized as follows. Section 2 discusses the related works. Section 3 describes the proposed mechanism. Section 4 presents the performance evaluation results. Section 5 concludes the study.

II. RELATED WORK

H.264/AVC is the most suitable standard for multimedia applications on the Internet, which defines three types of video frame, which are *I* (intra-coded) frames, *P* (predictive-coded) frames, and *B* (bidirectionally predictive-coded) frames. These three types of frame are used to encode various portions of a video signal at different levels of quality. The encoding and decoding of an *I* frame are independent of other frames. The *P* frame is uni-directionally predictively coded, and its encoding and decoding require information that precedes *I* frame or *P* frame. Since the coding of the *B* frame is predictive and bi-directional, encoding and decoding the *B* frame requires information that precedes and follows *I* or *P* frame. Therefore, the most important frame is the *I* frame that follows the P and B frames.

A periodic Group of Pictures (GOP) structure is recommended to prevent or at least to minimize error propagation. A GOP pattern is characterized as G(N, M), where N denotes the distance between two adjacent *I* frames, and M is the distance from *I* frame to *P* frame. Figure 1 shows as an example, G(9, 3), which means that the GOP pattern consists of one *I* frame, two *P* frames and six *B* frames. The arrows indicate the dependency of the frame decoding. The figure shows that the *I* frame is independent and that *decoding* of B frame and P frame must refers to the preceding or succeeding *I* or *P* frames.

Since the decoding of *B* frame had to reference the *I* or *P* frame, all of which must precede the B frame in transmission

QSHINE 2015, August 19-20, Taipei, Taiwan

DOI 10.4108/eai.19-8-2015.2260892

order. Figure 1 shown the different of the display order and transmission order in GOP(9, 3). In Figure 1, the display order of video sequence is I_1, B_2, B_3, P_4, B_5, B_6, P_7, B_8, B_9, I_{10}, and the transmission order of video sequence is I_1, P_4, B_2, B_3, P_7, B_5, B_6, I_{10}, B_8, B_9.

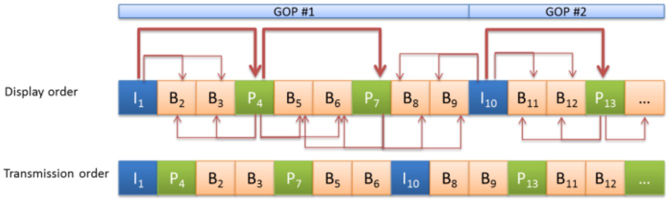

Figure 1. An example of the display order and transmission order in GOP(9, 3)

Therefore, essential video frames have the highest priority and the most opportunities for transmission, which improves video quality.

To support the differential data service in WLANs, the EDCA mechanism for allocating all video frames to AC_VI was developed. The mechanism does not guarantee the transmission of the most important video frame with the highest priority, and it does not consider the traffic loads of the ACs. Accordingly, the most important video frames, such as the *I* frames, may be dropped under heavy network loading. Numerous recent works have improved video streaming over an IEEE 802.11e network [2][3][6][7]. Nevertheless, the static mapping mechanism was developed for streaming MPEG-4 and H.264/SVC, which allocates the video frame packets to a predefined AC based on the significance of the video frame (*I/P/B* frame) [3], without considering the traffic load of the ACs. For instance, if the traffic load of AC_VI is light and those of AC_BE and AC_BK are heavy, then the static mapping mechanism will result in the dropping of various P/B frames packets owing to the overflow of AC_BE and AC_BK. *Wan et al.* presented an adaptive video scheduling scheme that is based on cross-layer design and unequal protection [6]. The mechanism allocates video frame packet to the ACs with minimum relative queuing delay to reduce the transmission delay. The authors also presented a dynamic frame assignment algorithm (DFAA) to guarantee the efficient transmission of high-priority frames efficiently. *Xiao* et al. proposed a dynamic queue scheduling mechanism to improve video transmission over WLANs, which adopts the concept of "relative queuing delay" to select the AC with minimum delay to transmit the video frame packet [7].

We previously proposed a mapping mechanism NMM [8], which dynamically maps MPEG-4 video packets to appropriate ACs, according to the importance of the video frame, the traffic loads of ACs and the actions performed on previous packets of the same video frame. The proposed mechanism in this work develops a mechanism to avoid the transmission of useless data, if the previous video frame packet is dropped, then an incoming packet of the same video frame is also dropped. The reason is that if one of the fragmented packets of a frame is dropped in the sender node owing to the overflow of the AC queue in WLANs, then the decoder at the receiver may not have sufficient information to reconstruct the original frame. For this purpose, the NMM develops a *flow* table in the sender node to record the action of the video frame packet. Figure 2 shows the data structure of *flow* table. In this work, the proposed mechanism considers the action that was taken on packet of the dependent video frame that was transmitted previously.

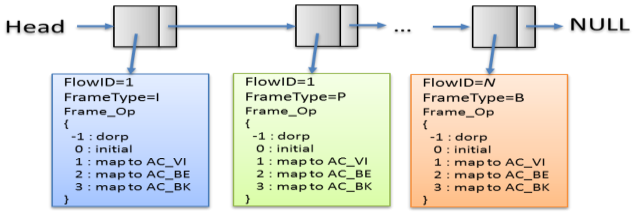

Figure 2. The data structure of the *flow* table

III. THE PROPOSED FRAME-BASED MAPPING MECHANISM

The proposed mechanism *HPPD* adopts a strategies to realize the efficient transmission of MPEG-4 video over IEEE 802.11e networks. The *HPPD* considers the conditions that the receiver could not decode the video frame. Such as the referencing video frame was damage while the receiver is decoding the receiving video frame. For example, if the *I* frame packet was dropped in the sender node, then the receiver may not reconstruct the original *I* frame. Hence, the receiver may not reconstruct the *P/B* frame of the same GOP, too. Therefore, the received video frame packets become useless so transmission resources are wasted.

In Figure 1, if the fragmented packets of P_4 were dropped in the sender node owing to the overflow of the AC queue in WLANs, then the decoder at the receiver may not have sufficient information to reconstruct the original frame. Hence, the following video frame B_2, B_3 and P_7 may also not have sufficient information to reconstruct the video frame even if all of the fragmented packets of video frames B_2, B_3 and P_7 were received successfully. In addition, B_5 and B_6 may also not reconstruct the video frame due to the video frame P_7 is undecodable. Therefore, the received video frame packets of B_2, B_3, P_7, B_5, B_6 become useless so transmission resources are wasted.

Accordingly, to well-utilize the network resources to transmit the useful video frame packets, the proposed mechanism will not transmit the useless video frame packets to avoid the waste of access channel resource. For example, if the *I* frame packet was dropped in the sender node, then the HPPD mechanism directly dropped the *P/B* frame packets in the same GOP, too. If the video frame P_4 was dropped in the sender node, then the proposed mechanism HPPD dropped the following video frame P_7, B_2, B_3, B_5 and B_6 in the same GOP.

For this purpose, the proposed mechanism develops a *Frame_Op* table in the sender node to record the action of the video frame. Figure 3 shows the data structure of *Frame_Op* table.

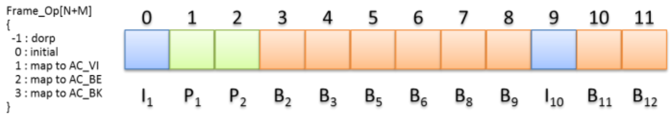

Figure 3. The data structure of *Frame_Op* table

Table 1. Parameter notations in our proposed HPPD mechanism

Parameter	Description
limit	The limit value of queue length of AC
threshold	The threshold of queue length
getlen(AC_VI)	The queue length of AC_VI
N	The distance between two adjacent I frames in a GOP
M	The distance from I frame to P frame
$currI$	The sequence number of last arrival I frame
Loc_{frame}	The location in the GOP of the arrival video frame packet of the GOP
GOP_{no}	The location in the GOP of the arrival video frame packet of the GOP
$frameno_{packet}$	The sequence number of arrival video frame packet
N_P	The number of P frame in a GOP
P_1	The first P frame in a GOP
GOP_1	The first GOP of a video sequence
$P_m^{GOP_n}$	The m-th P frame in the n-th GOP

The sequence number of the GOP is defined as equation (1).

$$GOP_{no} = \frac{frameno_{packet} - 1 - \left(N - \frac{N}{M} * (M-1) - 1\right) * M - 1}{N} + 2 \qquad (1)$$

The sequence number of last arrival I frame is defined as equation (2).

$$currI = 1 + \left(N - \frac{N}{M} \times (M-1) - 1\right) \times M + 1 \qquad (2)$$

The index of *Frame_Op* table of the arrival video frame packet is defined as equation (3).

$$Loc_{frame} = \begin{cases} (frameno_{packet} - currI) \bmod (N+M), if\ type\ of\ frame\ is\ I\ or\ B \\ \left(frameno_{packet} - 1 - \left(N - \frac{N}{M} \times (M-1) - 1\right) \times M - 1\right) \bmod N, if\ type\ of\ frame\ is\ P \end{cases} \qquad (3)$$

where $frameno_{packet}$ is the sequence number of an arrival video frame packet in the sender node.

For example, if the packet is associated with a video frame of type *P*, the GOP structure is GOP(9, 3) and the number of video frame is 11. The HPPD calculates the value of GOP_{no} is 2 by using equation (1). The HPPD calculates the value of *currI* is 8 by using equation (2) and the index of the arrival video frame in *Frame_Op* table is 3. Therefore, the HPPD read/record the action of the video frame by accessing *Frame_Op[3]*.

In addition, the location in the GOP of the arrival I/P/B frame packet of the GOP is defined as equation (4)(5)(6).

$$Loc_{I^{GOP_{no}}} = \begin{cases} 1, \ if\ GOP_{no} = 1 \\ (GOP_{no} - 1) \times N - 1, \ if\ GOP_{no} \geq 2 \end{cases} \qquad (4)$$

$$Loc_{P_m^{GOP_n}} = Loc_{P_1^{GOP_1}} + (GOP_n - 1) \times N + (m-1) \times M \qquad (5)$$

$$Loc_{B_{j,k}^{GOP_n}} = Loc_{P_j^{GOP_n}} + k \times 1, where\ 1 \leq j \leq N_p, 1 \leq k \leq M - 1 \qquad (6)$$

Figure 4 shows the pseudo code of the proposed HPPD algorithm. The HPPD uses different strategies for processing I/P/B video frame packets.

```
When (a video packet arrives){
   if (packet_type is 'I'){
   $Loc_{I^{GOP_{no}}} = (GOP_{no} - 1) \times N - 1$
      if Frame_Op[Loc] is 'drop'
            drop( video packet)
      else
         video packet -> AC[flow->frame_op]
      if AC_VI is not full
         video packet -> AC_VI
      else if AC_BE is not full and AC_BK is not full
         if queue length of AC_BE is shorter than AC_BK
            video packet -> AC_BE
         else
            video packet -> AC_BK
      else if AC_BE is not full and AC_BK is full
            video packet -> AC_BE
      else if AC_BE if full and AC_BK is not full
            video packet -> AC_BK
      else
            drop( video packet)
            sets the frame operation of P/B frame of the same GOP to be
            dropped
   }
   Else if (packet_type is 'P') {
   $Loc_{P_m^{GOP_n}} = Loc_{P_1^{GOP_1}} + (GOP_n - 1) \times N + (m-1) \times M$
      if Frame_Op[Loc] is 'drop'
            drop( video packet)
      else
         video packet -> AC[flow->frame_op]
      if queue length of AC_VI is less than threshold
         video packet -> AC[2]
      else if AC_VI is not full {
         $Prob = \frac{getlen(AC_{VI}) - threshold}{limit \quad - threshold}$
         RN= uniform(0,1)
         if(RN < Prob_New)
            video packet -> AC[1]
         else
            video packet -> AC[2]
      }
      else if AC_BE is not full and AC_BK is not full
         if queue length of AC_BE is shorter than AC_BK
            video packet -> AC_BE
         else
            video packet -> AC_BK
      else if AC_BE is not full and AC_BK is full
            video packet -> AC_BE
      else if AC_BE if full and AC_BK is not full
            video packet -> AC_BK
      else
            drop( video packet)
          sets the frame operation of subsequently P/B frame of the same
          GOP to be dropped
   }
   Else if (packet_type is 'B') {
   $Loc_{B_{j,k}^{GOP_n}} = Loc_{P_j^{GOP_n}} + k \times 1$
      if Frame_Op[Loc] is 'drop'
            drop( video packet)
      else
         video packet -> AC[flow->frame_op]
      if queue length of AC_VI is less than threshold
         video packet -> AC[2]
      else if AC_BE is not full and AC_BK is not full
         if queue length of AC_BE is shorter than AC_BK
            video packet -> AC_BE
         else
            video packet -> AC_BK
      else if AC_BE is not full and AC_BK is full
            video packet -> AC_BE
      else
         drop(video packet)
   }
record the action of the video frame packet of specific video flow
}
```

Figure 4. The pseudo code of the proposed mechanism HPPD

The proposed mechanism searches for the actions of the video frame in the predefined frame operation table. In order to search the value of the frame operation table, the HPPD found the index by using equation (4)(5)(6). Therefore, the mechanism directly performs the previously taken action on the current packet if the information was found. Otherwise, the following strategies are implemented.

- If the packet of video frame type is *I* frame, HPPD checks the queue length of AC_VI and directly maps the packet into AC_VI when the queue length of AC_VI is not full. However, while the AC_VI is full and the AC_BE is not full, the mapping decision maps the packet into AC_BE. Subsequently, the video packet will be mapped into AC_BK when AC_BK is not full. Otherwise, the video packet will be dropped. In addition, the proposed mechanism HPPD set the frame operation of P/B frame of the same GOP to be dropped in the frame operation table. The index of the frame operation table could be calculated by equation (5)(6).
- If the packet of video frame type is *P* frame, HPPD checks the queue length of AC_VI and compares it with threshold. If the queue length of AC_VI is less than *threshold*, the HPPD directly maps the packet into AC_VI. If the queue length of AC_VI is greater than *threshold* but not full, the probability for the frame packet mapping is defined as follows:

$$Prob = \frac{qlen(AC\_VI) - threshold}{limit_{AC\_VI} - threshold} \qquad (7)$$

where *qlen(AC_VI)* is the function to calculate the queue length of AC_VI, and $limit_{AC\_VI}$ is the limiting buffer size of AC_VI. Equation (7) provides the probability of *P* frame as *Prob* to be mapped to AC_VI or AC_BE. A higher *Prob* indicates that a packet has a higher priority to be processed in AC_BE. In other words, a lower *Prob* indicates that a packet has a higher priority to be mapped to AC_VI. When the *qlen(AC_VI)* is close to $limit_{AC\_VI}$, the *P* frame packet has higher probability of being mapping to AC_BE. On the contrary, when *qlen(AC_VI)* is close to *threshold*, the *P* frame packet has higher probability of being mapping to AC_VI. However, while the AC_VI is full and the AC_BE is not full, the HPPD directly maps the packet of the *P* frame to AC_BK when the queue length of AC_BK is shorter than AC_BE. Finally, HPPD dropped the frame packet and set the frame operation of subsequently P/B frame of the same GOP to be dropped in the frame operation table, while AC_VI, AC_BE and AC_BK are full.

- Finally, if the packet of video frame type is *B*, the mechanism more aggressively protects the important frame packets from being dropped. The *B* frame packets will be early dropped to reserve more queue space for *I* or *P* frame packets while the network is heavy loaded. If the *qlen(AC_VI)* is lower than *threshold*, the video packet is mapped into AC_VI. If not, the video packet is mapped into AC_BE when AC_BE is not full and the queue length of AC_BE is shorter than AC_BK. But if AC_BE is also full, the HPPD directly drops the video packet.
- Lastly, HPPD records the action of video packet based on its frame type.

IV. SIMULATION RESULTS AND DISCUSSION

A. 4.1 Simulation Model

NS-2 [9] was conducted to evaluate the performance of the proposed mechanism. The proposed mechanism was tested by using video sequences “Football” [10], which is in YUV CIF (352 × 288) format. The data rate of wireless link is 2 Mbps, and the maximum size of a packet that is transmitted over the simulated network is 1000 bytes. We used different loading cases, including different numbers of voice traffic (64 k), TCP flows, and UDP flows. The *threshold* of the access category queues is 40 packets, and the limit number of packets in the access category queues is 50 packets.

Four different scenarios are simulated in the evaluation of video transmission performance. In the first scenario, one voice flow, one TCP flow, one UDP flow, and two video streams are transmitted from the sender node to the receiver node. In the second scenario, two flows of each kind of traffic are transmitted from the sender node. In the third scenario, three voice flows, three TCP flows, three UDP flows, and two video streams are transmitted. In the fourth scenario, four voice flows, four TCP flows, four UDP flows, and two video streams are transmitted. Table 1 shows the number of video frames and packets of the video source.

Table 1. The number of video frames and packets of video sources

Video	Format	Frame number				Packets number			
		I	*P*	*B*	*Total*	*I*	*P*	*B*	*Total*
Football	CIF	34	54	172	260	282	350	725	1357

B. Simulation results and discussions

Table 2 show the average number of frame loss result of video Football under various loading cases by using different mechanisms.

The EDCA directly allocates all of the video packets to AC_VI even if the queue is in overflow. Therefore, the important video packets may be dropped while the AC_VI is full. Consequently, the AC_VI queue for the Football video more easily becomes full and drops the incoming packets. The adaptive mapping method dynamically allocates video packets to the appropriate ACs, based on the importance of video frame and the traffic loads of ACs. The adaptive mapping method may transmit unnecessary frame packets when the preceding frame packet has been dropped, wasting transmission resources. In Table 2, adaptive mapping drops many I frames in tests 3 and test 4.

Table 2. The average PSNR in the scenario

Loads case	Average PSNR (db)			
	802.11e EDCA	Adaptive mapping	NMM	HPPD
Test 1	16.37	20.91	28.70	**28.85**
Test 2	16.05	20.40	27.45	**27.72**
Test 3	15.82	19.03	26.56	**26.72**
Test 4	15.88	18.24	25.54	**25.70**

V. CONCLUSIONS AND FUTURE WORK

A Hierarchical packet pre-dropping (HPPD) approach is developed to improve the quality of the MPEG-4 video that is streamed over IEEE 802.11e wireless networks. HPPD dynamically allocates video packets to the most appropriate

ACs based on the network traffic load, frame type and the action that was taken on packet of the dependent video frame that was transmitted previously. Simulations verify that the HPPD successfully improves the quality of the transmitted video. The simulation results demonstrate that, under heavy network loading, the HPPD scheme outperforms the 802.11e EDCA, adaptive mapping scheme and novel mapping mechanism in terms of the quality of the transmitted MPEG-4 video.

ACKNOWLEDGEMENTS

Foundation item: The National Project of Taiwan (No.: MOST 103-2221-E507-001). Authors are grateful to Ministry of Science and Technology, Government of Taiwan for financial support to carry out this work.

REFERENCES

[1] Siris, V. A., & Courcoubetis, C. (2006, June). Resource control for the EDCA mechanism in multi-rate IEEE 802.11e networks. In *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks* (pp. 419-428). IEEE Computer Society.

[2] Foh, C. H., Zhang, Y., Ni, Z., Cai, J., & Ngan, K. N. (2007). Optimized cross-layer design for scalable video transmission over the IEEE 802.11e networks. *Circuits and Systems for Video Technology, IEEE Transactions on*, *17*(12), 1665-1678.

[3] Chen, H. L., Lee, P. C., & Hu, S. H. (2008, July). Improving scalable video transmission over IEEE 802.11e through a cross-layer architecture. In *Wireless and Mobile Communications, 2008. ICWMC'08. The Fourth International Conference on* (pp. 241-246). IEEE.

[4] Lin, C. H., Shieh, C. K., Ke, C. H., Chilamkurti, N. K., & Zeadally, S. (2009). An adaptive cross-layer mapping algorithm for MPEG-4 video transmission over IEEE 802.11e WLAN. *Telecommunication Systems*, *42*(3-4), 223-234.

[5] Ke, C. H., Yang, C. Y., & Chen, J. L. (2014, June). A Stream-Based Mapping Mechanism for MPEG-4 Video Transmission over IEEE 802.11e. In *Computer, Consumer and Control (IS3C), 2014 International Symposium on* (pp. 512-515). IEEE.

[6] Wan, Z., Xiong, N., Ghani, N., Peng, M., Vasilakos, A. V., & Zhou, L. (2011, April). Adaptive scheduling for wireless video transmission in high-speed networks. In *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on* (pp. 180-185). IEEE.

[7] Xiao, X., & Yanlin, X. (2013, December). A dynamic queue assignment algorithm under ieee 802.11e. In *Instrumentation and Measurement, Sensor Network and Automation (IMSNA), 2013 2nd International Symposium on* (pp. 186-189). IEEE

[8] Ke, C. H., Yang, C. Y., & Chen, J. L. (2015). A novel mapping mechanism for MPEG-4 video delivery over IEEE 802.11e networks. *The Journal of Multimedia Tools and Applications*. doi: 10.1007/s11042-015-2791-2

[9] The Network Simulator, NS2.29. Available at: http://www.isi.edu/nsnam/ns/.

[10] http://trace.eas.asu.edu/yuv..