Packet Scheduling for Video Streaming over Wireless with Content-Aware Packet Retry Limit

Chih-Ming Chen[1,2], Chia-Wen Lin[3] and Yung-Chang Chen[1]
[1]Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan 30013, R.O.C.
ycchen@ee.nthu.edu.tw
[2]Chunghwa Telecom Co., Ltd.
[3]Department of Computer Science & Information Engineering
National Chung Cheng University
Chiayi, Taiwan 621, R.O.C.
cwlin@cs.ccu.edu.tw

***Abstract*—In this paper, we propose a content-aware retry limit adaptation scheme for video streaming over IEEE 802.11 wireless LANs (WLANs). Video packets of different importance are unequally protected with different retry limits at the MAC layer. The loss impact of each packet is estimated to guide the selection of its retry limit. More retry numbers are allocated to packets of higher loss impact to achieve unequal error protection. Our scheme also analyzes the backoff time for each retry and then takes into account the estimated backoff time for retransmission scheduling. Experimental results show that our adaptation scheme can effectively mitigate the error propagation due to packet loss and assure the on-time arrival of packets for presentation, thereby improving video quality significantly.**

Keywords— wireless video; packet retransmission; packet scheduling; error control; video streaming

Topic area—multimedia networking.

1. Introduction

With low cost, easy deployment, and flexible connectivity, WLAN is becoming widespread and leading to fast-growing deployments in consumer homes. However, the challenges as to cope with the time-varying error rate and fluctuating bandwidth of a wireless network bring out the need of error resilient video transport. Forward Error Correction (FEC) and Automatic Retransmission reQuest (ARQ) are the two most commonly used channel coding schemes for error protection. FEC is more effective in multicast sessions [1] and applications with large end-to-end delay, whereas ARQ is particularly useful for non-interactive unicast applications with bursty packet loss and has been adopted in several existing packet protection methods for wireless video [2]-[5].

Video transport over wireless networks usually requires retransmissions to successfully send the video data to the receiver in case of packet loss, leading to increased delay time for the data to arrive at the receiver side. Delay constraint is, however, one of the most important requirements in real-time applications. A video packet arriving later than the presentation time will become useless for the client, making packet scheduling important in retransmission-based error control for wireless video streaming. In [2] the authors proposed a class of packet scheduling algorithms for wireless video streaming by applying different deadline thresholds to video packets of different importance. The importance of a packet is determined by its relative position within its group of pictures (GOP) and motion-texture context. The conditional retransmission scheme proposed in [3] uses the concealment error and the channel condition to determine whether a packet is worthwhile to retransmit. It provides a rate-distortion analysis of the trade-off between the saved-bits due to the reduced retransmission and the increased distortion resulting from the concealment error of not-retransmitted packets. The multi-user packet scheduling scheme proposed in [4] slows down the transmission of streams to users with favorite channel states until their deadline is approaching, leading to a fairer distribution of the achievable video quality among all users. The timestamp based Content-Aware Retry (CAR) mechanism proposed in [5] evaluates the influence (i.e., the effect of error propagation) of each frame in a GOP according to the number of frames inter-coded with respect to the frame. The CAR scheme then dynamically determines whether to send or discard a packet in one frame according to the influence and retransmission deadline of this frame.

In IEEE 802.11 WLAN networks, when a station wants to send data, it needs to take a backoff process to prepare for transmission. After data are sent, sender will wait for an ACK from receiver to confirm the data is arrived successfully at receiver. However, if the sender does not get the ACK within a specified timeout interval or detects another transmission in the channel, the sender will retransmit the frame again according to the backoff rule. For any transmission, the backoff interval is uniformly chosen in [0, CW-1], where CW is the contention window that will be doubled at each retransmission. A packet will be dropped after its retry limit has been reached. The standard allows a default of a maximum of transmission before the data is dropped [6].

Instead of adopting a static retry limit in IEEE 802.11, we propose a Content-Aware Retry Limit Adaptation (CA-RLA) scheme to dynamically adapt retry limit for each packet based on its loss impact. The key idea of CA-RLA is shown in Fig. 1. During the encoding process, the encoder estimates the loss impact (e.g., the amount of error propagation) of each packet if it is lost during transmission, and then generates side information as a hint for use in making retry decision for each packet. The proposed scheme tries to increase the retry limit of the packets of higher loss impacts, and to reduce the retry limit of packets of lower loss impacts so as to minimize the overall error propagation in a GOP under the delay constraint of video presentation.

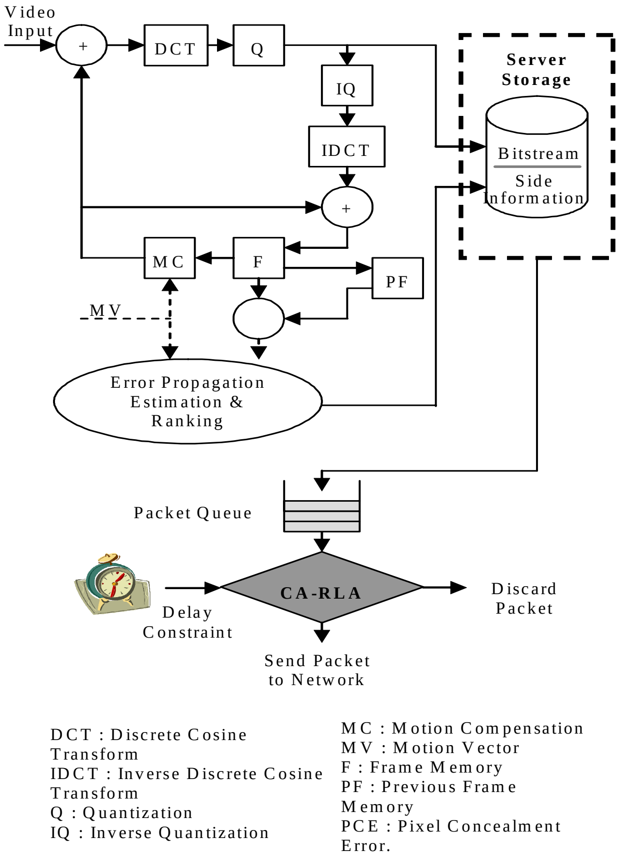


Fig. 1. Content-Aware Retry Limit Adaptation (CA-RLA) architecture.

The rest of this paper is organized as follows. Section 2 the estimation of backoff time for each retry in wireless networks is derived. In Section 3, the estimation of error propagation of each packet and CA-RLA are proposed. The retransmission-based packet scheduling with CA-RLA is presented in Section 4. Section 5 reports the experimental results of the proposed algorithms and the comparison with other methods. Finally, conclusions are drawn in Section 6.

2. Estimation of Backoff Time

In this paper, we present mathematical analyses of estimating the backoff time for each retry in IEEE 802.11 WLANs based on the Markov chain model proposed in [7]. Table I lists the system parameters used in the analyses. These parameters are either known a priori or can be derived from other known parameters.

TABLE I. System Parameters Defined in [7]

n_{ms}	Number of stations contending on the channel
T_s	time spent in a successful transmission
T_c	time spent in a collision
tSlotTime	time duration of one slot as defined in [6]
W	W=$CW_{\min}$ is the minimum contention window as defined in [6]
m	Contention window CW of the i-th retry is $CW = W_i = 2^i(CW_{\min}+1)$-1. $CW_{\max} = 2^m(CW_{\min}+1)$-1. When the retry reaches m, W_i will stay constant at $CW_{\max}$ as defined in [6]

We consider that every station always has a packet available for transmission (i.e., the saturated condition discussed in [7]), in which the probability for a packet to be transmitted in a slot, is derived in [7]:

$$\tau = \frac{2(1-2p)(1-p)}{(1-2p)(W+1)+pW(1-(2p)^m)}\ , \tag{1}$$

where p stands for the probability of detecting the channel busy as given in (2), which is also the collision probability of transmission.

$$p = 1-(1-\tau)^{n_{\text{ms}}-1} \tag{2}$$

Let P_{tr} denote the probability that there is at least one transmission in a slot, and P_s the probability that a transmission is successful, as shown below:

$$P_{tr} = 1-(1-\tau)^{n_{\text{ms}}}\ , \tag{3}$$

$$P_s = n_{\text{ms}}\tau(1-\tau)^{n_{\text{ms}}-1}\ . \tag{4}$$

According to the access mechanism of IEEE 802.11, the backoff timer for a station will count down as the medium is sensed idle, but stop counting when any transmission is detected. Therefore, at first, we define $P_w(k)$ as the probability of the station finishing w backoff slots with k slot times:

$$P_w(k) = \binom{k-1}{w-1} P_{tr}^{k-w}(1-P_{tr})^w, \tag{5}$$

where $k \geq w$. $P_w(k)$ is a negative binomial variable with parameter w and P_{tr}, and has the following property:

$$E_k[P_w(k)] = \frac{w}{(1-P_{tr})} \quad \text{with} \quad \sum_{k=w}^{\infty} P_w(k) = 1, \tag{6}$$

where $E_k[P_w(k)]$ represents the mean of $P_w(k)$ subject to k.

The time duration $\delta_w(k)$ of finishing w backoff slots with k slot times can be obtained using the model presented in [7]:

$$\delta_w(k) = w\cdot tSlotTime + (k-w)\cdot[P(P_s \mid P_{tr})\cdot T_s + P(P_{tr}-P_s \mid P_{tr})\cdot T_c],$$

where *tSlotTime*, T_s, and T_c are known a priori. Considering a system completely managed via the basic access mechanism, we obtain

$$T_s = T_{\text{hdr}} + T_{\text{data}} + DIFS + \delta + T_{\text{ACK}} + SIFS + \delta\,, \quad (7)$$

and

$$T_c = T_{\text{hdr}} + T_{\text{data}} + DIFS + \delta\,, \quad (8)$$

where δ denotes the propagation delay of trasmission. *SIFS* and *DIFS* are the time intervals defined for the access mechanism of IEEE 802.11. T_{data} represnts the duration to transmit a packet with size of $E[LEN_{\text{pktr}}]$, T_{hdr} is the duration of packet header, and T_{ACK} of the corresponding frame ACK. With IEEE 802.11 FHSS, these durations are given below:

$$\begin{cases} T_{hdr} = Header / R_{ch} \\ T_{data} = E[LEN_{pkt}] / R_{ch} \\ T_{ACK} = ACK / R_{ch} \end{cases},$$

where R_{ch} represents the channel bit rate. The overhead for sending the packet header include the header costs of physical and MAC layers, that is $Header = PHY_{\text{hdr}} + MAC_{\text{hdr}}$. We can calculate the expected value $E_k[\delta_w(k)]$ as

$$\begin{aligned}\Delta(w) &= E_k[\delta_w(k)] = \sum\nolimits_{k=w}^{\infty} \delta_w(k) \cdot P_w(k) \\ &= w \cdot tSlotTime + \left(E_k[P_w(k)] - w\right) \cdot \left[P(P_s \mid P_{tr}) \cdot T_s + P(P_{tr} - P_s \mid P_{tr}) \cdot T_c\right] \\ &= w \cdot tSlotTime + \left(\frac{w}{(1-P_{tr})} - w\right) \cdot \left[P(P_s \mid P_{tr}) \cdot T_s + P(P_{tr} - P_s \mid P_{tr}) \cdot T_c\right]\end{aligned}$$

As defined in [6], the backoff interval w of any transmission is uniformly chosen from [0, CW-1], where $CW = 2^r(CW_{\min}+1)-1$ for the r-th retry. As a result, we can derive the backoff time for the r-th retry as follows:

$$t_{\text{bf}}(r) = E_w[\Delta(w)] = \left[2^{r-1}(CW_{\min}+1) - \frac{1}{2}\right] \cdot K\,, \quad (9)$$

where

$$K = tSlotTime + \frac{P_{tr}}{(1-P_{tr})} \cdot \left[P(P_s \mid P_{tr}) \cdot T_s + P(P_{tr} - P_s \mid P_{tr}) \cdot T_c\right]. \quad (10)$$

3. Content-Aware Retry Limit Adaptation

The proposed CA-RLA allows the packet with higher error propagation to have more retry opportunity to reduce the probability of loss at receiver based on the timing constraint of transmission, so that packets are prioritized by the estimated amount of error propagation.

A. Estimation of error propagation

To estimate the error propagation impact of each lost packet, we first define the pixel-level loss-impact (LI) metric as the product of two parameters: PRC (Pixel Reference Count) and PCE (Pixel Concealment Error), to characterize the amount of pixel-wise error propagation as follows [8]:

$$LI(x,y,u) = PCE(x,y,u) \times PRC(x,y,u)\,, \quad (11)$$

where $PRC(x,y,u)$ represents the frequency of pixel (x,y) of frame u being referenced by pixels in the following frames within a GOP in the motion-compensated prediction (MCP) process. It can be calculated recursively by summing up the individual reference counts of pixels in frame u+1 which reference to pixel (x,y) of frame u in the reverse tracking order from the last frame to the first frame of a GOP as in (12), where N_{GOP} is the GOP size. In (13), $PCE(x,y,u)$ denotes the norm of concealment error of pixel (x,y) of frame, where $f(x,y,u)$ is the pixel value of pixel (x,y) in frame u, assuming the zero-motion error concealment scheme is adopted.

$$PRC(x,y,u) = \begin{cases} \sum\limits_{(x',y',u+1)\to(x,y,u)} PRC(x',y',u+1) & 1 \le u < N_{\text{GOP}} \\ 1 & u = N_{\text{GOP}} \end{cases} \quad (12)$$

$$PCE(x,y,u) = \left| f(x,y,n) - f(x,y,u-1) \right|^2. \quad (13)$$

We then use the motion information to calculate the current frame's macroblock-level error propagation by

$$EP_{\text{MB}}(u,v) = \sum_{(x,y)\in MB_v} LI(x+MV_x, y+MV_y, u-1)\,, \quad (14)$$

where v denotes the macroblock index in a frame; (x,y) denotes the pixel coordinate; u represents the time index; (MV_x, MV_y) represents the associated motion vector of pixel $(x,$y). Finally, all EP_{MB}'s in one packet are summed up to estimate the packet-level error-propagation as follows:

$$EP_{\text{pkt}}(u,q) = \sum_{v=1}^{N_{\text{MB}}} EP_{\text{MB}}(u,v)\,, \quad (15)$$

where q denotes the packet index of a frame, and N_{MB} denotes the number of macroblocks in the packet.

B. Content-Aware Retry Limit Adaptation

Consider a video sequence with M frames, inter-coded frame interval λ, and GOP size N_{GOP}. We formulate the deadline of presentation for video packet $PKT_{i,j}^q$ as follows:

$$D_t(PKT_{i,j}^q) = \beta + ((i-1) \cdot N_{GOP} + (j-1)) \cdot \lambda\,, \quad (16)$$

where we assume an initial delay β at the receiver, and $PKT_{i,j}^q$ denotes the q-th packet of the j-th frame within the i-th GOP. The lager the value of β is selected, the longer retry deadline the sender can deploy, but the receiver requires a larger-size buffer and a longer delay for video presentation. We uniformly assign the initial delay β to each GOP. We can formulate the time period T_{GOP} during which all the packets of one GOP are all received at the receiver as follows:

$$T_{\text{GOP}} = \frac{\beta + \lambda \cdot M}{M / N_{\text{GOP}}}. \quad (17)$$

From (9), while a packet is transmitted with a retry limit L_r and packet loss rate P_e, we can calculate the mean value of backoff time as

$$T(L_r,P_e)=(1-P_e^{L_r})\cdot\left(T_s+\frac{P_e}{1-P_e}\cdot T_c\right)+\sum_{i=0}^{L_r}P_e^{i}\cdot t_{\mathrm{bf}}(i)\cdot \quad (18)$$

Suppose the wireless link is a memoryless packet erasure channel and the packets are dropped independently. If the packet is dropped after L_r unsuccessful retries, we can obtain the packet erasure rate as:

$$p_L^k(L_r,P_e)=P_e^{L_r+1}. \quad (19)$$

Let p_L^k denote the packet loss probability of the k-th packet in a GOP with retry limit L_r^k, and EP_{pkt}^k its packet-level error-propagation as in (15). With the delay constraint, our goal is to find a set of retry limits $\{L_r^1, L_r^2,\ldots,L_r^k,\ldots,L_r^{N_{\mathrm{GOP}}^{\mathrm{pkt}}}\}$ for the packets in a GOP to minimize the total error propagation of the GOP as follows:

$$\min_{L_r^1,L_r^2,\ldots,L_r^{N_{\mathrm{GOP}}^{\mathrm{pkt}}}}\left\{\sum_{k=1}^{N_{GOP}^{pkt}}p_L^k\cdot EP_{\mathrm{pkt}}^k=\sum_{k=1}^{N_{\mathrm{GOP}}^{\mathrm{pkt}}}P_e^{L_r^k+1}\cdot EP_{\mathrm{pkt}}^k\right\}, \quad (20)$$

$$\text{subject to } \sum_{k=1}^{N_{\mathrm{GOP}}^{\mathrm{pkt}}}T(L_r^k,P_e)\le T_{\mathrm{GOP}}.$$

Based on the formulation shown in (20), the flowchart of the proposed CA-RLA algorithm is depicted in Fig. 2. In our method, the proposed CA-RLA tries to increase the retry limit of the packets with higher *EP*, and to reduce the retry limit of packets with lower *EP* under the delay constraint T_{GOP} and the requirement to minimize total *EP* in a GOP.

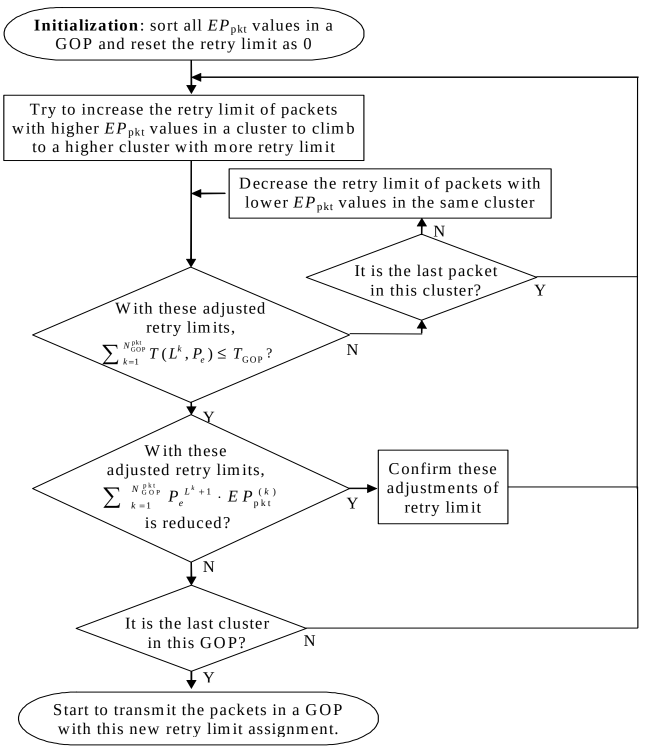


Fig. 2. Flowchart of CA-RLA.

4. Retransmission-Based Packet Scheduling

In addition to the proposed content-aware adaptation of MAC-layer retry limits, we also propose to schedule retransmission packets based on timeout estimation to prevent useless backoff waiting during the preparation for a transmission. The flowchart of our proposed packet scheduling algorithm based on the retry limit adaptation and backoff time estimation is shown in Fig. 3. In the scheduling, a packet will be discarded in a retry when the number of retries reaches its retry limit or its estimated arrival time is late for presentation; otherwise it will continues to take a backoff process for another retry. When the packet is transmitted successfully, it will be removed from the retransmission buffer.

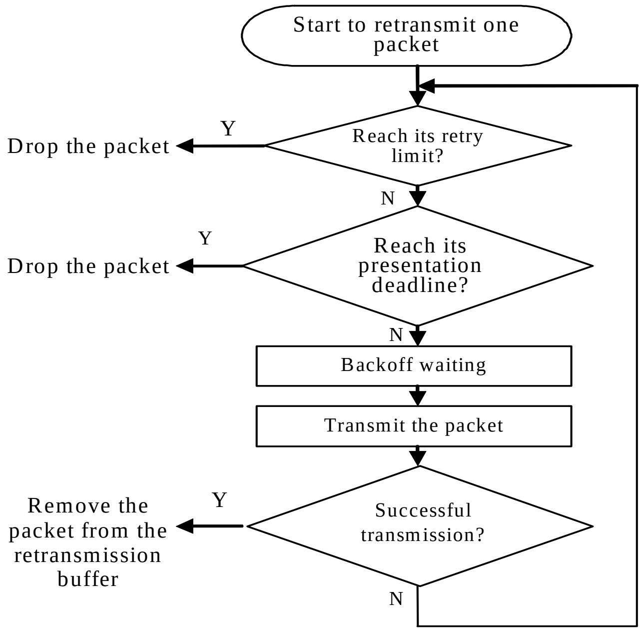


Fig. 3. Flowchart of retransmission-based packet scheduling in CA-RLA.

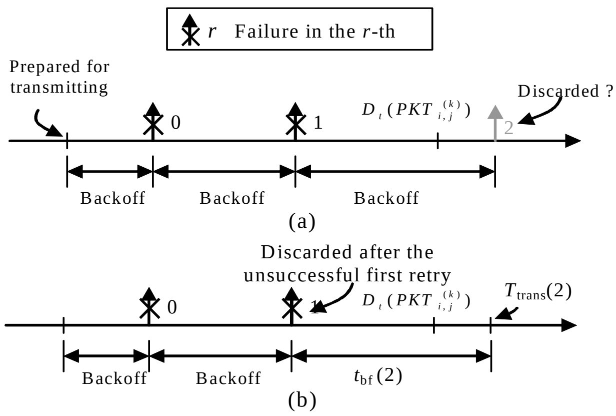


Fig. 4. Illustrations of two packet scheduling scheme: (a) the traditional method without backoff time estimation and (b) the proposed scheduling method based on backoff time estimation.

To prevent useless backoff waiting during the preparation for a transmission, it is reasonable to discard a packet should the estimated arrival time of the packet be later than the corresponding presentation deadline $D_t(PKT_{i,j}^{(k)})$ calculated by (16). Fig. 4 illustrates two packet scheduling schemes: the traditional approach and the proposed CA-RLA-based packet scheduling scheme. The tradition approach, as illustrated in Fig. 4(a), takes the backoff process for the second retry since the time right after the failed transmission of the first retry does not exceed the deadline $D_t(PKT_{i,j}^{(k)})$. However, it becomes too late for the second retry after finishing the backoff process. At this point the packet needs to be discarded, making the backoff waiting useless for this non-performed transmission. With the proposed backoff time estimation, the estimated time to pick up the next retransmission (the r-th retry) can be drawn as follows:

$$T_{\text{trans}}(r) = T_{\text{cur}} + t_{\text{bf}}(r), \tag{21}$$

where T_{cur} represents the time beginning to prepare the r-th retry and $t_{\text{bf}}(r)$ is the backoff time for this new retry which can be estimated by (9). In our method, a packet will be discarded early after the unsuccessful (r-1)-th retry if $T_{\text{trans}}(r) + \delta \geq D_t(PKT_{i,j}^{(k)})$, where δ denotes the propagation delay of trasmission. As shown in Fig. 4(b), the packet will be discarded after the unsuccessful first retry as $T_{\text{trans}}(2)+\delta$ is more than its presentation deadline $D_t(PKT_{i,j}^{(k)})$.

5. Simulation Results

We used the OPNET network simulator to simulate the network configuration which includes an independent basic service set (IBSS) and six mobile stations. In the test scenario, station 1 (the video sender) transmits an MPEG-4 video stream to station 2 (a video receiver), while the other stations simultaneously generate background traffic packets which contend for the channel. Two 300-frame QCIF (176×144) test sequences, *Foreman* and *Coastguard*, are respectively pre-encoded at 30 fps and 384 Kbps using an MPEG-4 software encoder. The structure of group of pictures (GOP) is (N_{GOP}, M) = (30,2), where N_{GOP} represents the GOP size, and M denotes the distance between two anchor frames. Each row of macroblocks are encoded as a slice and each slice is encapsulated into one packet. The background traffic packets were generated with a geometric distribution with parameter λ = 0.999. We set for all the background packets a fixed packet size of 180 bytes, same as the averaged packet length of video packets collected from our experiments.

According to the retry limit, a packet will be transmitted over and over until a transmission gets through or it reaches its retry limit. A packet will also be dropped in the case that the packet arrival time is later than its presentation deadline. It is reasonable to set an appropriate initial delay to extend the retransmission deadline to accommodate more retries due to channel contention caused by excessive traffics. In our simulations, the initail delay β is set to 1 s. A statistical analysis of extra retries beyond the retransmission deadline for various channel conditions was presented in [5]. For performance evaluation, the fixed retry limit scheme with different upper limits of retries and the CAR scheme proposed in [5] were also implemented and compared with the proposed method.

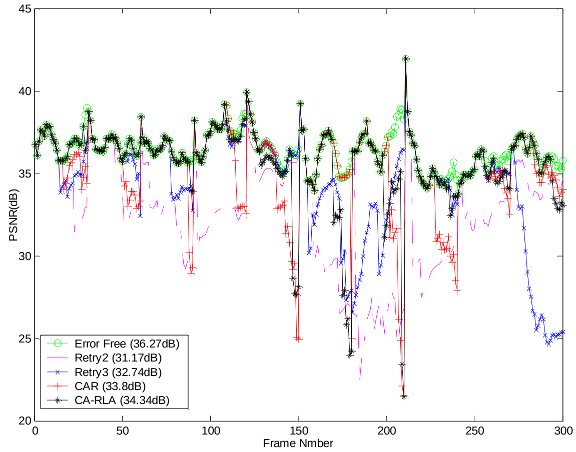
Table II shows the accuracy of the backoff time estimation based on the model in (9) compared to the actual experimental results using OPNET in the case that the number of mobile stations is six. The ratios of inaccuracy of the model estimates obtained by (9) are all less than 6.8 %. In the simulations, all packets can be transmitted successfully within five retries under our test scenario.

TABLE II. Accuracy Evaluation of the Backoff Time Estimated by (9) Compared to the Statistics Obtained from OPNET

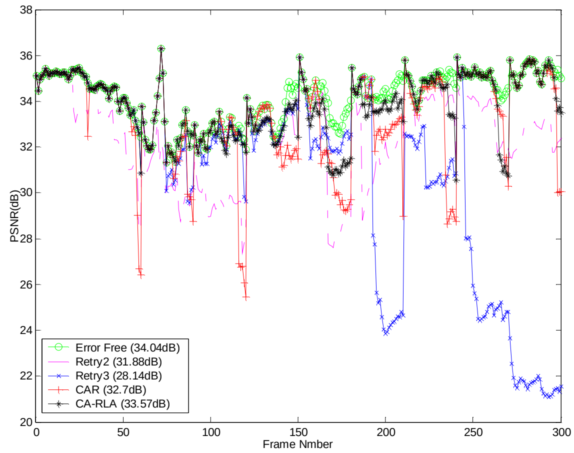
Retry No.	Estimated by (9)	Measured from OPNET	Inaccuracy Ratio
0	1.853 ms	1.893 ms	2.2 %
1	3.830 ms	3.884 ms	1.4 %
2	7.784 ms	7.663 ms	1.6 %
3	15.69 ms	16.00 ms	2.0 %
4	31.51 ms	30.99 ms	1.7 %
5	63.13 ms	58.87 ms	6.8 %

Fig. 5 shows the PSNR performance comparison of CA-RLA, the fixed retry limit method, and our implementation of the CAR method proposed in [5]. It shows that the method of fixed 2-retry limit causes excessive packet losses due to insufficient numbers of retries, thereby degrading the video quality severely. On the other hand, the fixed 3-retry limit leads to a relative large number of packets being dropped due to timeout for presentation, although there are almost no packets dropped due to an insufficient number of retries. With the CAR method, without taking into account the importance of each retransmitted packet, packets closer to the end of GOP always have higher possibility to be dropped based on the retransmission deadline adaptation within a GOP. The proposed CA-RLA takes into account the importance of each retransmitted packet for MAC-layer retry number adaptation as well as the estimated backoff time for retransmission scheduling, so as to recover video quality quickly from packet losses without causing too much error propagation.

Fig. 6 compares the frame-by-frame PNSR performances of CR-RLA with and without timeout estimation for Foreman (frame #270~#300) under the test case of n_{ms} = 8 and β = 9. The comparison indicates that without timeout estimation the sender still sends the packets after finishing the backoff waiting for the packets. Part of the packets may be dropped at the receiver due to timeout for presentation, thereby leading to more packet losses at the rear of the sequence and thus larger visual quality degradation.



(a)



(b)

Fig. 5. Frame-by-frame PSNR performance comparisons of four methods for test scenario 1 (i.e., n_{ms} = 6 and β = 1) for two test sequences: (a) *Foreman* and (b) *Coastguard*.

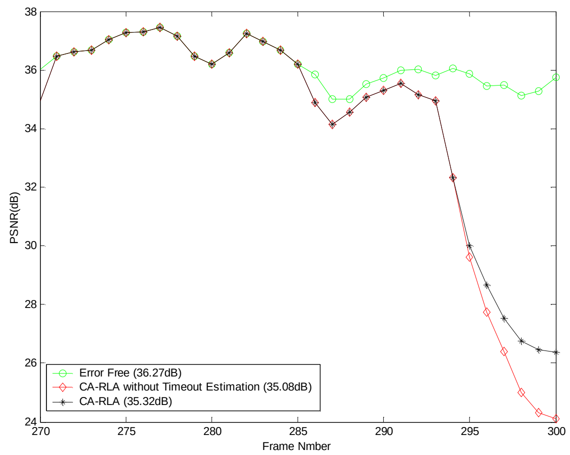


Fig. 6. Frame-by-frame PSNR performance comparisons of the CA-RLA with and timeout estimation at n_{ms} = 8 and β = 9.

6. Conclusion

We have proposed a novel CA-RLA scheme to adaptively set the retry limits of packets according to its error propagation characteristics for video streaming over WLANs. The CA-RLA scheme analyzes the backoff time for each retry, so as to find a retry limit set for packets in a GOP to minimize the total error-propagation of the GOP according to the delay constraint of packets for presentation at the receiver. The proposed method also takes into account the estimated backoff time for retransmission scheduling. Simulation results show that the proposed retry adaptation scheme significantly outperforms the conventional fixed retry limit mechanism in terms of visual quality. Besides, the proposed packet scheduling based timeout estimation can further improve the visual quality.

References

[1] A. Majumdar, D. Sachs, I. Kozintsev, K. Ramchandran, and M. Yeung, "Multicast and unicast real-time video streaming over wireless LANs," *IEEE Trans. Circuits Syst. Video Technol*, vol. 12, no. 6, pp.524-534, June 2002.

[2] S. H. Kang and A. Zakhor, "Packet scheduling algorithm for wireless video streaming," in *Proc. Workshop Packet Video*, Apr. 2002, Pittsburgh.

[3] S. Aramvith, C.-W. Lin, S. Roy, and M.-T. Sun, "Wireless video transport using conditional retransmission and low-delay interleaving," *IEEE Trans. Circuits Syst. Video Technol*., vol. 12, no. 6, pp. 558-565, June 2002.

[4] G. Liebl, M. Kalman, and B. Girod, "Deadline-aware scheduling for wireless video streaming," in *Proc. IEEE Int. Conf. Multimedia & Expo*., July 2005, Amsterdam, the Netherlands.

[5] M.-H. Lu, P. Steenkiste, and T. Chen, "Video streaming over 802.11 WLAN with content-aware adaptive retry," in *Proc. IEEE Int. Conf. Multimedia & Expo*., July 2005, Amsterdam, the Netherlands.

[6] IEEE P802.11. Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY). Nov. 1997

[7] M. Ergen and P. Varaiya, "Throughput analysis and admission control for IEEE 802.11a," *Mobile Networks and Applications* (*MONET*), vol. 10, pp. 705-716, Oct. 2005.

[8] H.-J. Chiou, Y.-R. Lee, and C.-W. Lin, "Content-aware error resilient transcoding using prioritized intra-refresh for video streaming," *J. Visual Commun. Image Represent*., vol. 16, no. 4-5, pp. 563-588, Aug.-Oct. 2005.