Cross-Layer Packet Retry Limit Adaptation for Video Transport over Wireless LANs

Chih-Ming Chen, *Member, IEEE,* Chia-Wen Lin, *Senior Member, IEEE,* and Yung-Chang Chen, *Fellow, IEEE*

***Abstract*—Video transport over wireless networks requires retransmissions to successfully deliver video data to a receiver in case of packet loss, leading to increased delay time for the data to arrive at the receiver. Delay constraint is 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. In this paper, we propose a cross-layer content-aware retry limit adaptation scheme for video streaming over IEEE 802.11 wireless LANs. Video packets of different importance are unequally protected with different retry limits at the media access control (MAC) layer. The error propagation effect of each packet is estimated to guide the determination 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 the proposed adaptation scheme can effectively mitigate the error propagation due to packet loss and assure the on-time arrival of packets for presentation, so as to improve video quality significantly.**

***Index Terms*—Cross-layer error protection, packet retransmission, retry limit adaptation, video streaming, wireless video.**

I. Introduction

WITH LOW COST, easy deployment, and flexible connectivity, deployments of wireless LANs (WLANs) [1] have become widespread and fast-growing everywhere. However, the challenges of coping with the time-varying error rate and fluctuating bandwidth of a wireless network bring out the need of error resilient video transport. Transmitting video data over error prone networks can be very unreliable due to packet loss, which still presents many challenges to streaming video applications, especially for wireless video. In a video streaming system, a server sends coded video data to client terminals for decoding and playback. There are several standard video coding techniques developed to compress a video sequence into a coded bitstream to reduce its data size. These video encoding techniques exploit spatial and temporal redundancy to achieve a high compression ratio, while making the compressed data very sensitive to transmission error. Packet loss due to transmission error often leads to serious video quality degradation, which not only degrades the quality of current frame, but also leads to error propagation to subsequent frames due to the motion-compensated prediction technique used in standard video codecs. Forward error correction (FEC) [2]–[4] and automatic retransmission request (ARQ) [3]–[5] are the two most commonly used channel coding schemes for error protection. FEC is relatively effective in multicast sessions 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 video streaming over wireless networks [5]–[16].

Video transport over wireless networks usually requires retransmissions to successfully deliver video data to a 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 its presentation time will become useless for a client, making packet scheduling important in retransmission-based error control for wireless video streaming. There have been a few research works addressing the scheduling problem of retransmitted packets. For example, a simple scheduling-oriented retransmission method which takes into account the play-out deadline based on the early-deadline-first (EDF) principle was presented in [6] and [7]. The EDF method gives priority to the lost packets with earliest play-out deadlines in transmission, which may lead to the overriding of regular packets by the retransmitted packets. The variants of EDF have been shown to be delay-optimal for deadline-constrained scheduling [7]. Dua and Bambos [8] proposed a downlink packet scheduler, namely, channel-aware earliest due date (CA-EDD) that takes into account the wireless channel conditions, packet deadlines, and application layer importance of packets. The CA-EDD algorithm achieves low-complexity implementation by using analytically computable approximations to the dynamic programming (DP) based optimal scheduling policy to avoid explicitly solving any DP equations, while achieving nearly optimal performance. Chou *et al.* [3] proposed a hybrid FEC/ARQ error protection scheme for receiver-driven layered multicast of audio-visual data. The method uses an iterative descent algorithm in a Lagrangian framework to find optimal transmission policies for video packets, provided that the media source layers are given by arbitrary directed acyclic graphs. It, however, does

Manuscript received June 19, 2008; revised February 3, 2009 and January 1, 2010; accepted June 8, 2010. Date of publication September 20, 2010; date of current version November 5, 2010. This paper was recommended by Associate Editor E. Steinbach.

C.-M. Chen is with Telecommunications Laboratories, Chunghwa Telecom Company, Ministry of Transportation and Communications, Taoyuan 326, Taiwan (e-mail: cmchen2@cht.com.tw).

C.-W. Lin and Y.-C. Chen are with the Department of Electrical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan (e-mail: cwlin@ee.nthu.edu.tw; ycchen@ee.nthu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSVT.2010.2077475

not consider packet delay jitter and loss of retransmission requests. A more general framework for rate-distortion optimized streaming of packetized media, RaDiO, was then proposed in [9] by generalizing the work in [3]. With RaDiO, the problem of rate-distortion optimized streaming of entire media presentation, in which data units (e.g., video packets) generally depend on each other according to a directed acyclic graph, can be reduced to the problem of error-cost optimized transmission of an isolated data unit. For the common application scenario of sender-driven streaming over a best-effort network using retransmission with feedback, a Markov decision processes framework was proposed to formulate the error-cost optimization problem so that the problem can be solved by existing algorithms such as DP or branch and bound algorithms. Ren *et al.* [10] addressed the problem of scheduling constant bit rate traffic over deadline-constrained wireless networks by deriving an optimal scheduling policy to minimize long-term packet loss due to deadline expiration, considering the relative impact of deadline constrains and channel conditions. The conditional retransmission scheme proposed in [12] uses the concealment error and the channel feedbacks to determine whether a packet is worthwhile to retransmit. It provides a rate-distortion analysis of the tradeoff between the saved bits due to the reduced retransmission and the increased distortion resulting from the concealment error of not-retransmitted packets. Kang and Zakhor [13] proposed a class of rate-distortion optimized 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, thereby achieving unequal error protection. Chakareski and Frossard [14] proposed a distributed streaming strategy to allow for trading rate for distortion, not only within a single video stream, but also across different streams by employing rate-distortion information including packets' sizes and their importance for the reconstruction quality of the corresponding stream. The cross-layer approach proposed in [15] provides adaptive quality of services to a layered coded video by utilizing priority queuing at the network layer and retry-limit adaptation at the link layer. It considers the media access control (MAC) retry limit, application-layer FEC, and adaptive packet size selection in an integrated manner such that the throughput efficiency be maximized for a given channel condition. In [16], a receiver-based optimization framework was proposed for maximizing the time to receiver underflow and associated undesirable video freeze by pre-emptively evicting frames from the receiver buffer.

IEEE 802.11-like networks adopt a MAC-layer retry mechanism [1], in which a lost packet is retransmitted several times until a retry limit is reached. It has been shown in [18]–[23] that an adaptive MAC retry mechanism can achieve significant performance improvement over the static-retry-limit approach. The layered prioritized queuing (L-PQ) presented in [18] takes into account together the MAC retry limit and sender buffer occupancy to determine the operating point that minimizes the overall packet loss rate of video layers due to buffer overflow and link errors according to the priorities of video layers. It, however, does not consider the backoff delay of retransmission that may lead to presentation timeout. The multi-user packet scheduling algorithm proposed in [19] slows down the transmission of streams to users with favorable channel states until their deadline is approaching, leading to a fairer distribution of the achievable video quality among all users. This multi-user scheduling algorithm incorporates information about video stream structure and future channel behavior to determine an optimal schedule based on a simplified rate-distortion model. In [20], a learning-based approach was applied, where content, coder-specific, and channel features were used to predict the optimal cross-layer strategy for assigning MAC-layer retry limits for video transmissions under delay constraints. Similar to the approach proposed in [13], a time-based adaptive retry (TAR) mechanism was proposed in [22] and [23] to improve video streaming over 802.11 WLANs. In the TAR scheme, the influence (i.e., the effect of error propagation) of each frame in a GOP is evaluated according to the number of frames inter-coded with respect to the frame. It then dynamically determines whether to send or discard a packet in one frame according to the influence and retransmission deadline of this frame. Significant performance improvement over the traditional static-retry-limit approach in terms of packet loss, channel utilization, and user-perceived visual quality was reported.

Although time-based adaptive retry limit adaptation [13], [22], [23] has proven to be a promising scheme, simply exploiting the temporal prediction relationship among frames may not be able to well capture the error propagation of a lost packet. The performance of time-based MAC retry adaptation can be further improved by incorporating better video content analysis and packet retry scheduling with the knowledge on the arrival times of packets. In this paper, we propose a cross-layer content-aware retry limit adaptation (CA-RLA) scheme that dynamically adapts the retry limit for each packet based on its loss impact. We consider the application scenario that the compressed video bitstream is either pre-stored in the transmitter or sent to the transmitter through a reliable access network. The system framework of CA-RLA is illustrated in Fig. 1. In the off-line encoding process, the encoder estimates the amount of error propagation caused by each packet should it be lost during transmission. This side information is stored as metadata in the streaming server for guiding the decision on the retry limit of each packet. While performing real-time streaming, the transmitter retrieves the side information associated with the video bitstream as well as estimates the client channel conditions according to channel feedbacks and actual backoff waiting statistics. The proposed CA-RLA scheme increases the retry limits of packets of higher loss impacts, while reducing the retry limits of packets of lower loss impacts so as to minimize the overall error propagation in a GOP under the delay constraint of video presentation. Our scheme also analyzes the backoff time for each retry and then takes into account the estimated backoff time for retransmission scheduling.

The contribution of this paper is twofold. First, we develop for IEEE 802.11-like networks a systematic analysis to derive a closed-form formula for accurately estimating the mean

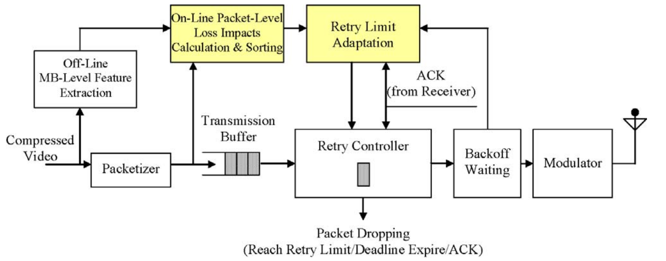

On-Line Packet-Level Loss Impacts Calculation & Sorting
Retry Limit Adaptation
Off-Line MB-Level Feature Extraction
ACK (from Receiver)
Transmission Buffer
Compressed Video
Packetizer
Retry Controller
Backoff Waiting
Modulator
Packet Dropping (Reach Retry Limit/Deadline Expire/ACK)

Fig. 1. System framework of the proposed CA-RLA.

transmission time of a packet for a given retry limit, which is an important parameter in delay-constrained video streaming applications. The second contribution of this paper is to address the retry limit adaptation problem under a constrained optimization framework by appropriately incorporating cross-layer parameters including the loss impact (application-layer) and estimated transmission time (MAC-layer) of each packet.

The rest of this paper is organized as follows. Section II presents the estimation of error propagation of each packet to assign the appropriate priority level to data packets. In Section III, the estimation of backoff time for each retry in wireless networks is presented. The proposed CA-RLA with retransmission-based packet scheduling is presented in Section IV. Section V reports the experimental results of the proposed algorithms and the comparison with other methods. Finally, conclusions are drawn in Section VI.

II. Estimation of Packet-Level Loss Impacts

The most accurate method of estimating the impact of a single packet loss is to delete the packet and then decode a GOP by recovering the lost data (e.g., using the zero-motion error concealment). The mean squared error (MSE) can thus be calculated by comparing the error-free video and the reconstructed video with single packet loss. This approach, however, is computationally too expensive. In this paper, we adopt our previous method presented in [24] that characterizes the pixel-level loss-impact (LI) metric as the product of two parameters: pixel reference count (PRC) and pixel-wise concealment error (PCE) by

$$LI(t, x, y) = PCE(t, x, y) \cdot PRC(t, x, y) \tag{1}$$

where $PRC(t, x, y)$ represents the frequency of pixel (x,y) of frame t being referenced by pixels in the following frames within a GOP in the motion-compensated prediction (MCP) process as illustrated in Fig. 2. It can be calculated recursively by summing up the individual reference counts of pixels from frame $t + 1$ to the last frame of the GOP which directly or indirectly reference pixel (x,y) of frame t in an inverse tracking order as in

$$PRC(t, x, y) = \begin{cases} \sum\limits_{(x',y')\in\Omega_{(x,y)}} PRC(t+1, x', y') & 1 \le t < N_{\text{GOP}} \\ 1 & t = N_{\text{GOP}} \end{cases} \tag{2}$$

where $\Omega_{(\mathrm{x,y})}$ denotes the set of pixels in frame $t + 1$ that reference to pixl (x, y) of frame t, N_{GOP} represents the GOP size.

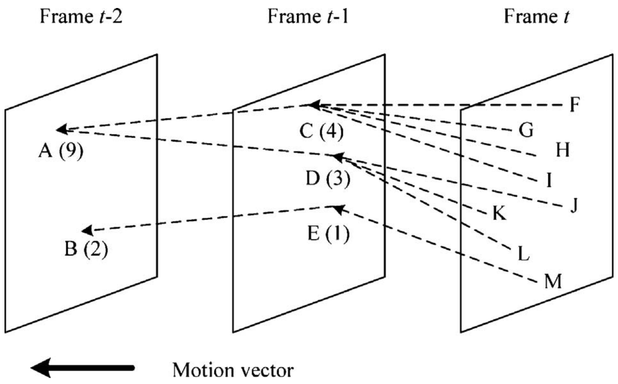


Fig. 2. Illustration of calculating the PRC. Assume frame u is the last frame of a GOP, the numbers in the braces indicate the PRCs of pixels.

$PCE(t, x, y)$, as defined in (3), denotes the norm of concealment error of pixel (x, y) of frame t should this pixel be corrupted

$$PCE(t, x, y) = |f(t, x, y) - f(t-1, x, y)|^2 \tag{3}$$

where $f(t, x, y)$ represents the pixel value of pixel (x, y) in frame t, assuming the zero-motion error concealment scheme is adopted.

The macroblock-level loss-impact is then calculated by

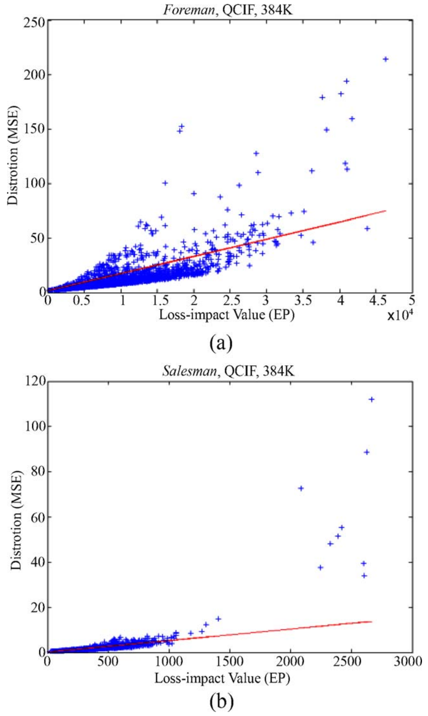
$$LI_{\text{MB}}(t, v) = \sum_{(x,y)\in MB_v} LI(t, x, y) \tag{4}$$

where v represents the macroblock index in a frame and t represents the time index of frame, (x, y) denotes the pixel coordinate, (MV_x, MV_y) represents the associated motion vector of pixel (x, y). As a result, all LI_{MB}s in one packet are summed up to estimate the packet-level error-propagation (EP) as follows:

$$EP_{\text{pkt}}^{(k)} = \sqrt{\sum_{MB_v \in packet\ k} LI_{\text{MB}}(t, v)} \tag{5}$$

where k denotes the packet index of a frame.

The proposed scheme is simple yet effective. Fig. 3 illustrates the relationship between packets' loss impact values and the corresponding MSE caused by packet loss for two test sequences. In the experiment, each packet contains one row of macroblocks and the loss estimation is based on the assumption that there is only a single packet loss in a GOP. Although the MSE value is not a strictly linear function of loss-impact, the estimated loss-impact linearly approximates the amount of MSE with a fairly good accuracy when the loss-impact value is not very high. In the proposed CA-RLA method, most computation for loss-impact estimation is done off-line. The loss-impact values are pre-stored as side information for use in real-time streaming. Other distortion estimation models, such as the ROPE model proposed in [22] and the parametric models proposed in [23] and [24] also can be used to estimate the loss-impacts of packets. If scalable video coding (SVC) is employed, the layer-dependent models for SVC distortion estimation proposed in [9] and [25] can be used.



(a)

(b)

Fig. 3. MSE distortion versus estimated loss-impact value (EP), and the corresponding linear least squares fitting for three test sequences. (a) *Foreman*. (b) *Salesman*.

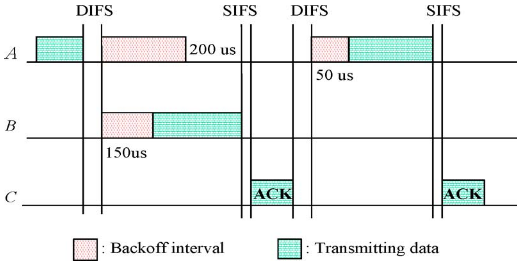


Fig. 4. Backoff example of IEEE 802.11.

TABLE I
SYSTEM PARAMETERS USED IN BACK-OFF TIME ANALYSIS

n_{ms}	Number of stations contending for 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 [1]
W	$W = CW_{\min}$ is the minimal contention window size
m	Contention window CW of the rth retry is $CW = W_r = 2^r(CW_{\min}+1)$. $CW_{\max} = 2^m(CW_{\min}+1)$. When the retry reaches m, W_i will stay constant at $CW_{\max}$ [1].

III. ESTIMATION OF BACKOFF TIME OF EACH MAC-LAYER RETRY

In IEEE 802.11 [1], prior to sending data packets, a mobile station has to probe the channel to make sure whether another one is transmitting. If the medium is sensed idle for an interval that exceeds the distributed interframe space (DIFS), the station will start its transmission. If the medium is busy, it will continuously monitor the channel until the channel becomes idle for a DIFS, and then the station initiates a random backoff interval that is used to initiate the backoff timer for the station. The timer counts down as the medium is sensed idle, but stops counting when any transmission is detected, and will continue counting down when the medium is sensed idle again for more than a DIFS. When the backoff timer reaches zero, the station will start its transmission. When the receiver receives the data frame successfully and waits for a short interframe space (SIFS), it will send back an acknowledgement (ACK) to the sender. If the sender does not receive 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 the value $CW = 2^r(CW_{\min}+1)$, for the rth retry attempt, is called the contention window with an initial minimal value of $CW_{\min}+1$ (the default value of $CW_{\min}$ is 15) at the first transmission attempt. Each time the retry counter increases, the CW will be doubled, up to a maximal value $CW_{\max} = 2^m(CW_{\min}+1)$ (the default value of m is six). A packet will be dropped after its retry limit has been reached. The current IEEE 802.11 standard allows a default of a maximum of seven retries before the packet is dropped. A backoff example of IEEE 802.11 is illustrated in Fig. 4, where users A and B try to send data to user C. After 150 μs backoff, user B is granted the opportunity to transmit data to user C. Since the medium becomes busy when user B is sending data, user A needs to stop counting. After user C sends back an ACK to user B, the medium is sensed idle again. After waiting for a DIFS, user A continues counting down for the remaining 50 μs, then begins to transmit data to user C should there be no another station sending data in the slot.

We present here a mathematical analysis of estimating the backoff time for each retry in an IEEE 802.11 WLAN based on the Markov chain model proposed in [29] that is an extension of the throughput analysis presented in [30]. Table I lists the system parameters used in the analysis. These parameters are either known *a priori* or can be derived from other known parameters.

We consider that every station always has a packet available for transmission (i.e., the saturated condition discussed in [26]), in which the probability of a packet being transmitted from a station in a slot, as derived in [29], is given by

$$\tau = \frac{2(1-2p_c)(1-p_c)}{(1-2p_c)(W+1)+p_c W(1-(2p_c)^m)} \tag{6}$$

where p_c stands for the probability of detecting the channel busy (also the collision probability of transmission) as follows:

$$p_c = 1-(1-\tau)^{n_{\text{ms}}-1}. \tag{7}$$

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 follows:

$$P_{tr} = 1-(1-\tau)^{n_{\text{ms}}} \tag{8}$$

and

$$P_s = n_{\text{ms}}\tau(1-\tau)^{n_{\text{ms}}-1}. \tag{9}$$

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 will stop counting when any transmission is detected. Therefore, we first define $P_w(k)$ as the probability of a station finishing w backoff slots with k slot times as follows:

$$P_w(k) = \binom{k-1}{w-1} P_{tr}^{k-w}(1 - P_{tr})^w \qquad (10)$$

where $k \geq w$. As shown in [28], $P_w(k)$ is a negative binomial variable with parameters w and P_{tr}, and has the following property:

$$E_k[P_w(k)] = \frac{w}{(1 - P_{tr})} \text{ with } \sum_{k=w}^{\infty} P_w(k) = 1 \qquad (11)$$

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

The time duration $\delta_w(k)$ of finishing w backoff slots with k slot times can be expressed as

$$\begin{aligned}\delta_w(k) = {} & w \cdot tSlotTime + (k - w) \\ & \cdot [P(P_s|P_{tr}) \cdot T_s + P(P_{tr} - P_s|P_{tr}) \cdot T_c]\end{aligned}$$

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

$$T_s = T_{\text{hdr}} + T_{\text{data}} + DIFS + \delta + T_{\text{ACK}} + SIFS + \delta \qquad (12)$$

and

$$T_c = T_{\text{hdr}} + T_{\text{data}} + DIFS + \delta \qquad (13)$$

where δ denotes the transmission propagation delay, $SIFS$ and $DIFS$ are the time interval parameters defined in the access mechanism of IEEE 802.11, and T_{data}, T_{hdr}, and T_{ACK} represent the durations for sending a packet with size of $E[LEN_{\text{pkt}}]$, the packet header, and the corresponding ACK frame, respectively. With the IEEE 802.11 frequency-hopping spread spectrum (FHSS), these durations are as follows:

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

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 $Header = PHY_{\text{hdr}} + MAC_{\text{hdr}}$. According to (11), the expected value $E_k[\delta_{\text{w}}(k)]$ is

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

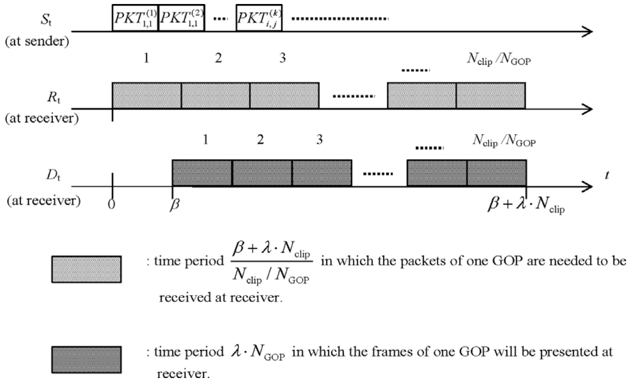

S_t
(at sender)
$PKT_{1,1}^{(1)}$
$PKT_{1,1}^{(2)}$
$PKT_{i,j}^{(k)}$
1
2
3
$N_{\text{clip}}/N_{\text{GOP}}$
R_t
(at receiver)
1
2
3
$N_{\text{clip}}/N_{\text{GOP}}$
D_t
(at receiver)
0
β
$\beta + \lambda \cdot N_{\text{clip}}$
t
: time period $\frac{\beta + \lambda \cdot N_{\text{clip}}}{N_{\text{clip}} / N_{\text{GOP}}}$ in which the packets of one GOP are needed to be received at receiver.
: time period $\lambda \cdot N_{\text{GOP}}$ in which the frames of one GOP will be presented at receiver.

Fig. 5. Time schedule for packet transmission.

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

$$t_{\text{back}}(r) = E_w[\Delta(w)] = \left\lceil 2^{r-1}(CW_{\text{min}} + 1) - \frac{1}{2} \right\rceil \cdot K \qquad (14)$$

where

$$K = tSlotTime + \frac{P_{tr}}{(1 - P_{tr})} \cdot [P(P_s|P_{tr}) \cdot T_s + P(P_{tr} - P_s|P_{tr}) \cdot T_c]. \qquad (15)$$

IV. Content-Aware Retry Limit Adaptation and Packet Scheduling

A. Content-Aware Retry Limit Adaptation

Consider an N_{clip}-frame video clip with an interframe interval λ. As illustrated in Fig. 5, we formulate the presentation deadline for packet $PKT_{i,j}^{(k)}$ as follows:

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

where $PKT_{i,j}^{(k)}$ denotes the kth packet of frame j of GOP i.

In (16), we assume a startup delay of receiver buffer β that is set intentionally for accommodating a reasonable amount of packet delay. The larger the value of β is selected, the longer retry deadline the sender can deploy, but leading to a larger-size buffer at the decoder and a longer delay for video presentation. For the sake of simplicity, we uniformly distribute the initial delay β to each GOP. According to Fig. 5, the time period T_{GOP} during which the packets of one GOP should all arrive at the receiver is

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

Let L_r be the retry limit of a packet, and P_e the packet loss rate of the link of each transmission (without retransmission). The average number of transmissions for a packet to be received successfully, or reach its retry limit, can be computed by

$$s(L_r, P_e) = 1 \cdot (1 - P_e) + 2 \cdot P_e(1 - P_e) + \ldots + (L+1)P_e^{L_r} = \frac{1 - P_e^{L_r+1}}{1 - P_e}$$

and

$$p_e = p_c + p_f$$

where p_c represents the packet loss rate caused by data collision as defined in (7) and p_f is the rate of packet loss

TABLE II
IEEE 802.11B SYSTEM PARAMETERS FOR FHSS

MAC header	281 bits
PHY header	1351 bits
ACK	240 bits
Channel bit rate (*BR*)	11 Mb/s
Propagation delay (δ)	1 μs
tSlotTime	50 μs
SIFS	28 μs
DIFS	128 μs
$CW_{\min}$	15
$CW_{\max}$	1023

due to signal fading which can be obtained from the receiver through a feedback channel. In a heavily congested channel, typically p_c is significantly larger than p_f.

Assuming a packet is transmitted with a retry limit L_r and packet loss rate P_e, we can calculate the mean value of a packet's transmission time as

$$\begin{aligned}
T(L_r, P_e) &= (t_{\text{back}}(0) + T_s) \cdot (1 - P_e) \\
&\quad + (t_{\text{back}}(0) + T_c + t_{\text{back}}(1) + T_s) \cdot P_e(1 - P_e) \\
&\quad + (t_{\text{back}}(0) + T_c + t_{\text{back}}(1) + T_c + t_{\text{back}}(2) + T_s) \\
&\quad \cdot P_e^2(1 - P_e) \\
&\quad + \ldots \\
&\quad + (t_{\text{back}}(0) + T_c + t_{\text{back}}(1) + T_c + \ldots \\
&\quad + t_{\text{back}}(L_r - 1) + T_s) \cdot P_e^{L_r-1}(1 - P_e) \\
&\quad + (t_{\text{back}}(0) + T_c + t_{\text{back}}(1) + T_c + \ldots + t_{\text{back}}(L_r - 1) \\
&\quad + T_c + t_{\text{back}}(L_r) + T_s) \cdot P_e^{L_r} \\
&= (1 - P_e^{L_r}) \cdot \left(T_s + \frac{P_e}{1 - P_e} \cdot T_c \right) \\
&\quad + \sum_{r=0}^{L_r} P_e^r \cdot t_{back}(r) \qquad (18)
\end{aligned}$$

where T_s and T_c are defined in (12) and (13), respectively.

Plugging (14) into (18), we have

$$\begin{aligned}
T(L_r, P_e) &= \left(1 - P_e^{L_r}\right) \cdot \left(T_s + \frac{P_e}{1 - P_e} \cdot T_c \right) \\
&\quad + K \cdot \sum_{r=0}^{L_r} P_e^r \cdot \left[2^{r-1}(CW_{\min} + 1) - \frac{1}{2} \right]. \qquad (19)
\end{aligned}$$

Suppose the wireless link is a memoryless packet erasure channel and the packets are dropped independently. Since the kth packet is dropped after $L_r^{(k)}$ unsuccessful retries, we can obtain the packet erasure rate as

$$PLR^{(k)}(L_r^{(k)}, P_e) = P_e^{L_r^{(k)}+1}. \qquad (20)$$

```
Algorithm 1 Content-Aware Retry Limit Adaptation (CA-RLA)

Definitions:
  $G_{\text{RL}}(l)$:         the $l$th sub-group of a GOP in which all packets have a same retry limit
  $PKT_{\text{low}}(l)$:      the packet with the lowest $EP_{\text{pkt}}$ in $G_{\text{RL}}(l)$
  $PKT_{\text{high}}(l)$:     the packet with the highest $EP_{\text{pkt}}$ in $G_{\text{RL}}(l)$
  Cost$^-$($l$):              $\Delta^- EP_{\text{pkt}} / \Delta^- T$, a ratio of the amount of increased $EP_{\text{pkt}}$
                              to the amount of released time for packet $PKT_{\text{low}}(l)$ should its retry limit
                              be reduced from $L_r$ to $L_r$–1, where
                              $\Delta^- EP_{\text{pkt}} = \left[(P_e^{L_r-1} - P_e^{L_r}) \cdot EP_{\text{pkt}}^{PKT_{\text{low}}(l)}\right]$,
                              and $\Delta^- T = T(L_r, P_e) - T(L_r - 1, P_e)$.
  Value$^+$($l$):             $\Delta^+ EP_{\text{pkt}} / \Delta^+ T$, a ratio of the amount of reduced $EP_{\text{pkt}}$
                              to the amount of consumed time for packet $PKT_{\text{high}}(l)$ should
                              its retry limit be increased from $L_r$ to $L_r$+1, where
                              $\Delta^+ EP_{\text{pkt}} = \left[(P_e^{L_r} - P_e^{L_r+1}) \cdot EP_{\text{pkt}}^{PKT_{\text{high}}(l)}\right]$ and $\Delta^+ T =$
                              $T(L_r + 1, P_e) - T(L_r, P_e)$.

Initiation:
                              Sort the packets in a GOP by their loss impact values to obtain
                              $\left\{EP_{\text{pkt}}^{(1)}, EP_{\text{pkt}}^{(2)}, \ldots, EP_{\text{pkt}}^{(k)}, \ldots, EP_{\text{pkt}}^{(N_{\text{pkt}})}\right\}$,
                              $EP_{\text{pkt}}^{(k-1)} \geq EP_{\text{pkt}}^{(k)}$ and uniformly distribute the corresponding retry
                              limit to each packet under the timing constraint imposed in (21),
                              $\left\{L_r^{(1)}, L_r^{(2)}, \ldots, L_r^{(k)}, \ldots, L_r^{(N_{\text{pkt}})}\right\}$, $L_r^{(k-1)} \geq L_r^{(k)}$

Adaptation:
  Step 1.   Try to reduce the retry limit of the $PKT_{\text{low}}(l)$ with the lowest Cost$^-$ to release
            free timebudget
  Step 2.   According to the free time budget, try to increase the retry limit of the $PKT_{\text{high}}(l)$
            with the highest Value$^+$
  Step 3.   Accept the adaptation of retry limit assignment if the overall error propagation can be
            reduced under the timing constraint imposed in (21).
  Step 4.   Go to Step 1 until the overall error propagation reduced in an iteration is less than
            a predefined threshold
```

Let $PLR^{(k)}$) represent the packet loss probability of the kth packet in a GOP with retry limit $L_r^{(k)}$ and packet loss rate P_e, and $EP_{\text{pkt}}^{(k)}$ its packet-level error propagation as defined in (5). With the delay constraint, our goal is to find a set of retry limits $\{L_r^{(1)}, L_r^{(2)}, \ldots, L_r^{(k)}, \ldots, L_r^{(N_{\text{pkt}})}\}$ for the packets in a GOP to minimize the overall error propagation of the whole GOP which can be formulated as the following constrained optimization problem:

$$\min_{L_r^{(1)}, L_r^{(2)}, \ldots, L_r^{(N_{\text{pkt}})}} \left\{ \sum_{k=1}^{N_{\text{pkt}}} PLR^{(k)} \cdot EP_{\text{pkt}}^{(k)} = \sum_{k=1}^{N_{\text{pkt}}} P_e^{L_r^{(k)}+1} \cdot EP_{\text{pkt}}^{(k)} \right\}$$

$$\text{subject to} \sum_{k=1}^{N_{\text{pkt}}} T(L_r^{(k)}, P_e) \leq T_{\text{GOP}} \qquad (21)$$

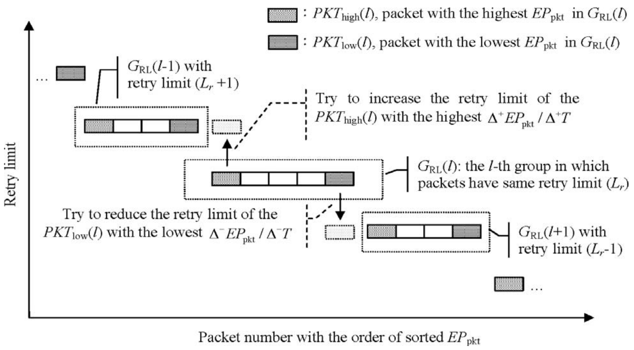

▨ : $PKT_{\text{high}}(l)$, packet with the highest EP_{pkt} in $G_{\text{RL}}(l)$
▨ : $PKT_{\text{low}}(l)$, packet with the lowest EP_{pkt} in $G_{\text{RL}}(l)$
$G_{\text{RL}}(l\text{-}1)$ with retry limit (L_r +1)
Try to increase the retry limit of the $PKT_{\text{high}}(l)$ with the highest $\Delta^+EP_{\text{pkt}} / \Delta^+T$
$G_{\text{RL}}(l)$: the l-th group in which packets have same retry limit (L_r)
Try to reduce the retry limit of the $PKT_{\text{low}}(l)$ with the lowest $\Delta^-EP_{\text{pkt}} / \Delta^-T$
$G_{\text{RL}}(l$+1) with retry limit (L_r-1)
Retry limit
Packet number with the order of sorted EP_{pkt}

Fig. 6. Greedy-based retry limit adaptation in the proposed CA-RLA.

where N_{pkt} denotes the number of packets in a GOP.

Table II lists the typical values of important system parameters for IEEE 802.11 FHSS, among which the propagation delay of data transmission is relatively negligible compared to other system parameters.

From (21), as shown in the Appendix, we can derive a DP recursion, that is usually used to solve the 0/1 knapsack problem [16], [32], to determine the optimal solution. The DP-base approach, however, is computationally very expensive because of its large state space as analyzed in Appendix, thereby making it impractical for real-time applications. Instead, the integer-rounding-based complexity scaling scheme proposed in [33] can be used in the DP recursion to trade video quality for complexity. For the sake of simplicity, we propose a low-complexity Greedy heuristic to reduce the complexity without sacrificing visual quality significantly. The proposed Greedy adaptation algorithm is illustrated in Fig. 6 and summarized in Algorithm 1.

Under the delay constraint T_{GOP}, the proposed Greedy-based adaptation scheme tends to reduce the retry limits of the packets of low loss impacts to release more free time budget so as to increase the retry limits of the packets of high loss impacts to achieve unequal error protection of packets. The ratio of the amount of reduced (increased) EP_{pkt} to the amount of consumed (released) time for a packet is used to determine which packets' retry limits need to be adjusted. The proposed scheme is a suboptimal solution; however it is efficient to meet the real-time requirement without sacrificing performance significantly.

B. Dynamic Reallocation of Packet Retry Limits

In the realizations of packet transmissions, it is probable that a packet will be received successfully in fewer transmissions than its allocated retry limit. The above-mentioned adaptation algorithm can be further improved if the saved transmission budget can be used in the transmissions of flowing packets. Here, we propose a dynamic Greedy heuristic to reallocate the extra retry budget to the remaining most important packets. The retry limit adaptation in dynamic Greedy algorithm is divided into GOP and packet levels. Similar to the rate control algorithms used in video coding, at the GOP level, the remaining time budget, that is the total time budget deducts what has been used till now, is uniformly allocated to the remaining GOPs. The allocation is updated on per GOP basis.

Algorithm 2 Packet-Level Dynamic CA-RLA (Greedy-Based)

```
A packet in sub-group $G_{\text{RL}}(L_r)$ with retry limit $l$ is received
successfully after $(L_r - s)$ retries.
if $s > 0$
    for $k = 1, 2, \ldots, s$
    {
        if there is any packet waiting to be transmitted in the
        sub-group $G_{\text{RL}}(L_r - k)$
        The retry limit of the packet in $G_{\text{RL}}(L_r - k)$ with the
        largest loss impact will be increased by 1
        (i.e., $L_r - k + 1$) and the packet is moved to
        $G_{\text{RL}}(L_r - k + 1)$.
        else
            if $(L_r - k) > m$ ($m$ is the saturation retry limit
        defined in Table I) The retry limit of the packet in
        $G_{\text{RL}}(L_r - k–1)$ with the largest EP is increased by 1
        (i.e., $L_r - k$) and the packet is moved to $G_{\text{RL}}(L_r - k)$.
        else
        The retry number of the two most important packets
        in $G_{\text{RL}}(L_r - k–1)$ will be increased by 1 and the packets
        are moved to $G_{\text{RL}}(L_r - k)$.
        end if
    end if
    }
end if
```

For a video clip with N_{clip} frames, the time budget for the ith GOP T^i_{GOP} is calculated by

$$T^i_{\text{GOP}} = \frac{T^{\text{all}}_{\text{budget}} - \sum_{n=1}^{i-1} T^n_{\text{used}}}{M/N_{\text{GOP}} - i + 1} \quad i = 1, 2, ..., N_{\text{clip}}/N_{\text{GOP}} \quad (22)$$

where $T^{\text{all}}_{\text{budget}}$ represents the total time budget for the whole video clip, T^n_{used} denotes the actual time used to transmit the nth GOP. There are $N_{\text{clip}}/N_{\text{GOP}} - i + 1$ remaining GOPs prior to sending the ith GOP.

Given the GOP-level time budget, the packet-level allocation follows. Should a packet be received successfully with a retry number that is less than its allocated retry limit, the saved time will be reallocated to the highest loss-impact ones out of those packets of the GOP that have not yet been sent to the receiver. The packets waiting to be sent in a GOP are divided into sub-groups according to their retry limits, that is, packets in the rth sub-group $G_{\text{RL}}(r)$ have the same retry limit $L_r = r$. Since different rounds of retry consume significantly different mean transmission time [see (14) and (15)], the time saved from not taking the rth retry for a packet is on average only enough for adding an extra rth retry to another packet in sub-group $G_{\text{RL}}(r–1)$ [or adding an extra $(r–1)$th retry to two packets in $G_{\text{RL}}(r–1)$]. For example, if a packet is received successfully after $(r–s)$ retries, where $s \geq 0$ is the saved number of retries, s can be reallocated sequentially to add one more retry to the packets in the sub-groups $G_{\text{RL}}(r–1)$, $G_{\text{RL}}(r–2)$, $\ldots$, $G_{\text{RL}}(r–s)$ with the most loss impact in the same sub-group. The proposed reallocation algorithm is summarized as follows:

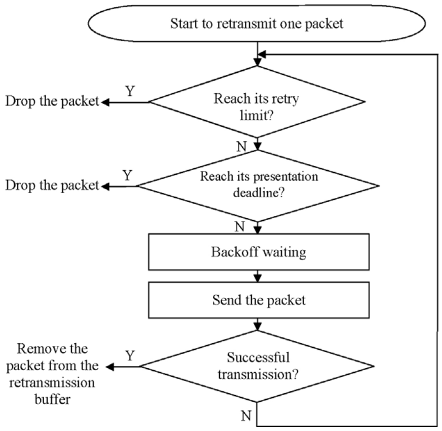
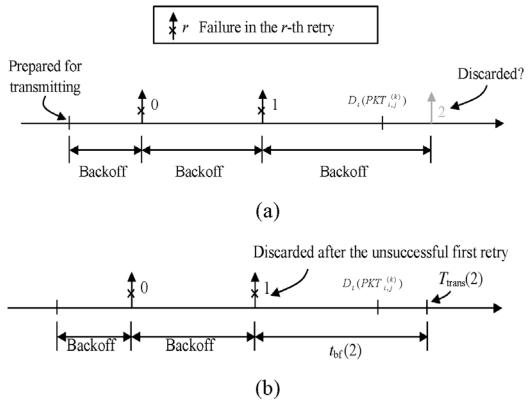


Fig. 7. Flowchart of timeout-based packet scheduling in CA-RLA.



(a)

(b)

Fig. 8. Illustrations of two packet scheduling scheme. (a) Traditional method without backoff time estimation. (b) Proposed scheduling method based on backoff time estimation.

V. Retransmission-Based Packet Scheduling

On top of the proposed retry limit adaptation scheme, we propose a retransmission packet scheduler based on timeout estimation to prevent useless backoff waiting during the preparation for a transmission. In the flowchart shown in Fig. 7, a packet will be discarded before a retry when the number of retries exceeds its retry limit or its estimated arrival time is late for presentation; otherwise, it will continue to take a backoff process for another retry. When the packet is transmitted successfully, it will be removed from the retransmission queue.

To prevent useless backoff waiting for a retry, it is reasonable to discard a packet if the estimated arrival time of a packet is later than the packet's presentation deadline calculated by (16). Fig. 8 illustrates two packet scheduling schemes: the traditional approach without backoff time estimation and the proposed scheme. The traditional approach in Fig. 8(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. However, it becomes too late for the second retry after finishing the backoff process, making the backoff waiting useless for this non-performed transmission. The estimated time to take the next retry (the rth retry) can be drawn by

$$T_{\text{trans}}(r) = T_{\text{cur}} + t_{\text{back}}(r) \tag{23}$$

TABLE III

Two Test Scenarios Used for Experiments

Test Scenario	No. of IBSS	No. of Mobile Station (n_{ms})	Initial Delay (β)
Scenario 1 (slight congestion)	1	6	1 s
Scenario 2 (heavy congestion)	1	8	9 s

where T_{cur} represents the time beginning to prepare the rth retry and $t_{\text{back}}(r)$ represents the estimated backoff time for this new retry which can be estimated by (14).

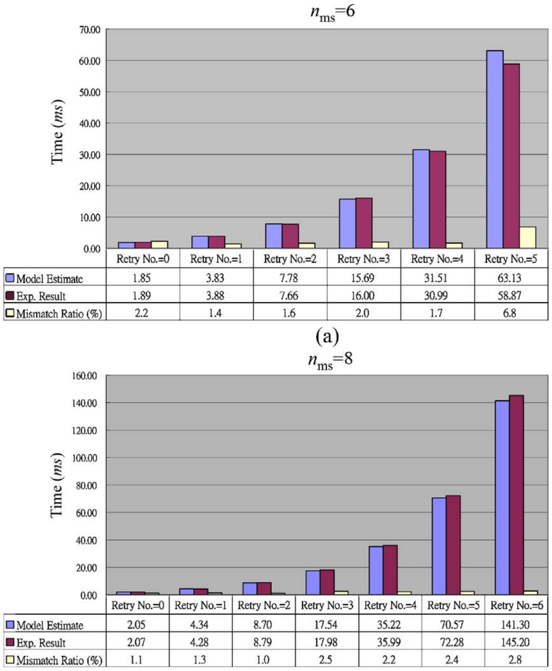
In our method, a packet will be discarded early after the unsuccessful (r–1)th retry if $T_{\text{trans}}(r) + \delta \geq D_t(PKT_{i,j}^{(k)})$, where δ denotes the transmission delay as illustrated in Fig. 8(b).

VI. Experimental Results

We used the OPNET v.8.3 network simulator [34] to simulate an 11 Mb.s 802.11b network configuration which includes an independent basic service set (IBSS) and a number of mobile stations. In our experiments, as shown in Table III, two test scenarios with two different numbers of mobile stations (n_{ms} = 6 and 8) and startup delay settings (β = 1 and 9 s) at the decoders were evaluated, respectively. The numbers of stations used in our experiments were selected to simulate the situations with overloaded traffics. In the test scenarios, 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 that contend for the channel. Three 300-frame QCIF (176 × 144) test sequences, *Foreman*, *Coastguard*, and *Salesman*, were pre-encoded at 30 frames/s and 384 kb/s using a public-domain MPEG-4 software encoder [35]. The GOP structure was (N_{GOP}, M) = (30, 1), where N_{GOP} represents the GOP size, and M denotes the distance between two anchor I/P-frames. Each row of macroblocks is 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 to make the network overloaded. We set for all the background packets a fixed packet size of 180 bytes, which is close to the average 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 the number of retires reaches the packet's retry limit as shown in Fig. 7. Besides, a packet will be dropped if the arrival time of the packet is later than its presentation time. It is usually required to set an appropriate startup delay to extend the retransmission deadline to accommodate more retries due to channel contention caused by excessive traffics. Note, the appropriate length of startup delay depends on the degree of network congestion and the length of the video affected by congestion. In our experiments, as shown in Table III, the startup delay β is set to 1 s for scenario 1 (6 stations) and 9 s for scenario 2 (8 stations), respectively. The impact of startup delay on the performance of the proposed method shall be discussed later.

We first evaluate the accuracy of the proposed backoff time estimation scheme. Fig. 9 shows the accuracy of the proposed



	Retry No.=0	Retry No.=1	Retry No.=2	Retry No.=3	Retry No.=4	Retry No.=5
Model Estimate	1.85	3.83	7.78	15.69	31.51	63.13
Exp. Result	1.89	3.88	7.66	16.00	30.99	58.87
Mismatch Ratio (%)	2.2	1.4	1.6	2.0	1.7	6.8

(a)

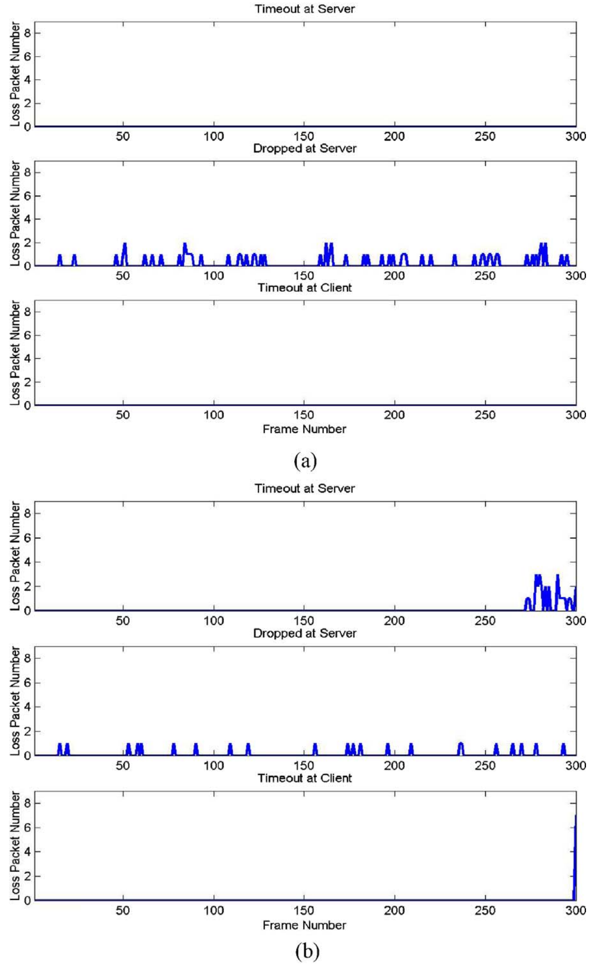
	Retry No.=0	Retry No.=1	Retry No.=2	Retry No.=3	Retry No.=4	Retry No.=5	Retry No.=6
Model Estimate	2.05	4.34	8.70	17.54	35.22	70.57	141.30
Exp. Result	2.07	4.28	8.79	17.98	35.99	72.28	145.20
Mismatch Ratio (%)	1.1	1.3	1.0	2.5	2.2	2.4	2.8

(b)

Fig. 9. Accuracy evaluation of the backoff time (in ms) estimated by the model in (14) compared to the experimental result obtained from the OPNET simulator with two different numbers of stations. (a) $n_{\mathrm{ms}} = 6$. (b) $n_{\mathrm{ms}} = 8$.

model in (14) compared to the actual experimental results using OPNET under two test scenarios ($n_{\mathrm{ms}} = 6$ and 8). The mismatch ratio of the model estimates obtained by (14) ranges from 1.4% to 6.8%, mostly falling in 2–3%. The good accuracy of the proposed model makes it useful in estimating the delay time of packet retry so as to solve the delay-constrained scheduling optimization problem in (21). In our method, the number of stations needs to be inferred prior to the backoff delay estimation. Since packet scheduling is performed on per GOP basis. It is reasonable to assume that the number of stations involved in a GOP duration (about 0.5–1 s) is stable and close to those numbers in the neighboring GOPs. Since the number of stations contending for the channel in previous GOPs can be monitored and recorded, the statistical data collected in previous GOPs can be used by a sender to estimate the number of stations involved in the current GOP. Our method simply uses the number of stations in the previous GOP as an estimate of the current GOP's station number.

With the proposed packet scheduling method, there are three classes of packet dropping. The first class of packet dropping is caused by an insufficient number of retries due to the retry limit. The second is dropping of packets at the sender due to insufficient (estimated) time for sending the packet to the receiver by the presentation deadline. The final is dropping of packets at the receiver due to their late arrival (later than their presentation deadline) even they are successfully received. As a result, the less the retry limit, the more the number of dropped packets due to insufficient retries. Oppositely, the more the retry limit, the more the number of dropped packets due to late packet arrivals (i.e., later than the presentation deadline). For example, as shown in Fig. 10(a), if the retry limit is too few (e.g., $L_r = 2$ in this case) for a certain channel condition, there will be many packets reaching the retry limit and thereby being dropped at the sender. On the other hand, Fig. 10(b) shows the case that a higher number of retry limit ($L_r = 3$) for packets runs the risk of packet dropping due to late packet arrivals caused by excessive retries.



(a)

(b)

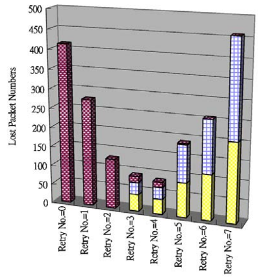
Fig. 10. Numbers of packets dropped at the sender and the receiver under test scenario 1 for two fixed retry limits. (a) Two retries. (b) Three retries.

TABLE IV

Average PSNR Performance Comparisons With Various Retry Limits Under Two Test Scenarios

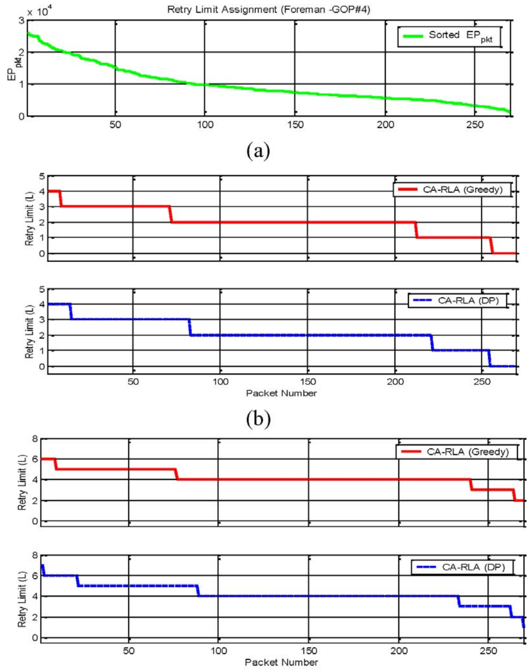
PSNR	Retry Limit Number							
(dB)	0	1	2	3	4	5	6	7
Scenario 1 ($n_{\mathrm{ms}} = 6$, $\beta = 1$)	19.86	26.53	31.17	32.74	25.15	21.63	21.63	21.63
Scenario 2 ($n_{\mathrm{ms}} = 8$, $\beta = 9$)	18.79	24.59	28.77	31.82	32.12	28.49	24.23	15.00

The average PSNR performances with different fixed retry limits under two test scenarios are compared in Table IV, in which the best retry numbers for these two cases are three (32.74 dB) and four (32.12 dB), respectively. Our observation from the experiment is, since more retries can mitigate the packet loss due to insufficient retry limit, the PSNR performance is improved when the number of retry limit is increased initially. However, due to the backoff waiting mechanism in 802.11, a later retry consumes exponentially growing backoff



	Retry No.=0	Retry No.=1	Retry No.=2	Retry No.=3	Retry No.=4	Retry No.=5	Retry No.=6	Retry No.=7
▨ Exceed Retry Limit	413	277	129	16	14	3	0	0
▨ Timeout at Sender	0	0	0	31	31	97	138	252
🟨 Timeout at Receiver	0	0	0	44	39	89	118	206

Fig. 11. Numbers of packets dropped at the sender and the receiver under test scenario 2 (n_{ms} = 8, β = 9) at different retry limits.



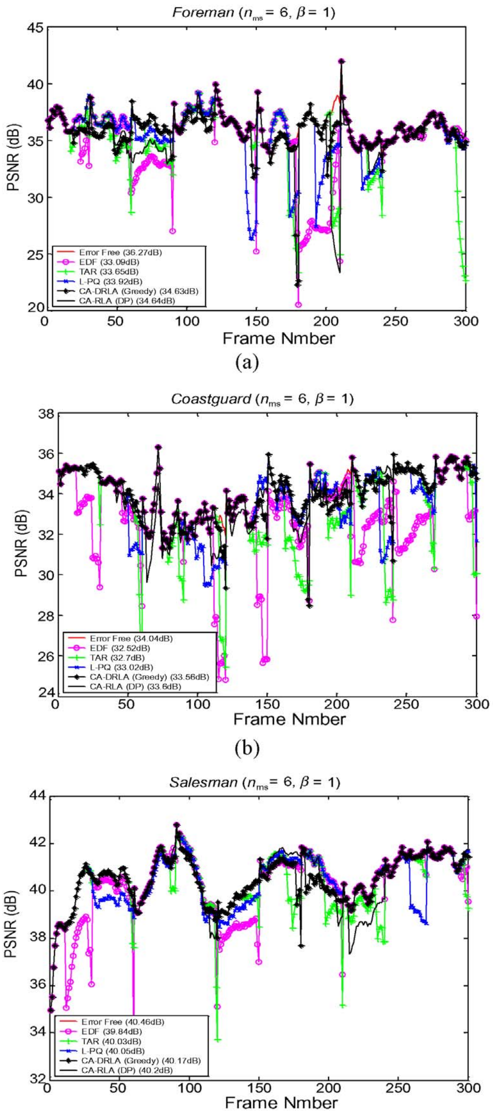
(a)

(b)

(c)

Fig. 12. Example of retry limit assignment. (a) Sorted EP_{pkt} values of packets in a GOP. (b) Retry limits assigned to the packets by CA-RLA with Greedy-based optimization (upper) and by DP-based optimization (lower) for scenario 1 (n_{ms} = 6, β = 1), and (c) for scenario 2 (n_{ms} = 8, β = 9).

time compared to its previous retry. The backoff time growth saturates after the sixth retry as specified in [1]. Therefore, such exponential growth of backoff time due to excessive retries of a packet does not only make the possibility of timeout for the packet itself high, but also blocks the transmission of other unsent packets during the long waiting period, thereby significantly increasing the number of the second and third classes of lost packets. Fig. 11 shows the distribution of the three kinds of packet dropping with different fixed retry limits under test scenario 2 (n_{ms} = 8 and β = 9 s). We can observe that retry limits of 3–4 lead to the fewest numbers of lost packets, thereby achieving the highest PSNR quality. The fixed retry limit mechanism is kind of equal error protection schemes, whereas adapting the retry limits of packets achieves unequal error protection for the packets. Fig. 12(a) depicts an example of sorted EP_{pkt} values of packets in a GOP and the packets' corresponding retry limits assigned by the proposed



(a)

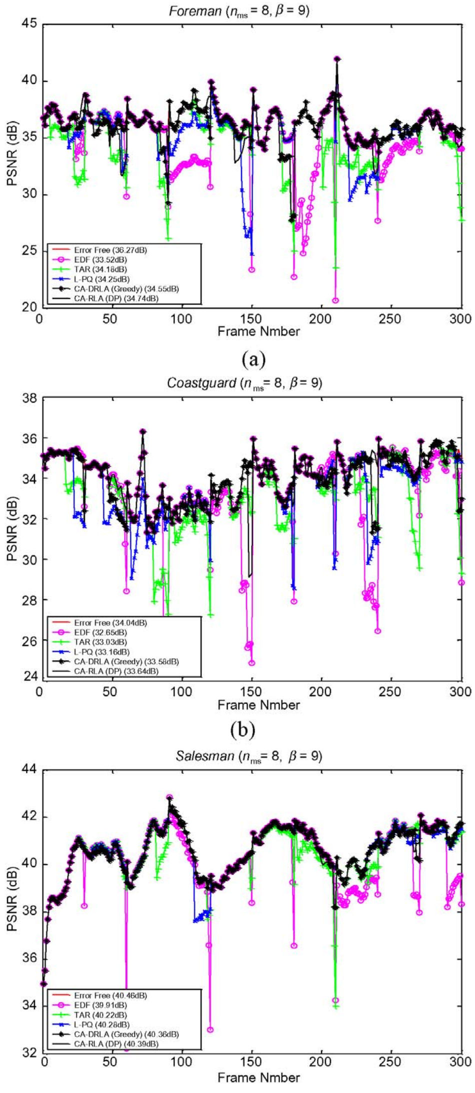
(b)

(c)

Fig. 13. Frame-by-frame PSNR performance comparisons of four methods for test scenario 1 (i.e., n_{ms} = 6 and β = 1) for three test sequences. (a) *Foreman*. (b) *Coastguard*. (c) *Salesman*.

CA-RLA scheme. In this example, the retry limits assigned to the packets range from 0 to 4 for test scenario 1 and from 1 to 7 for test scenario 2, as illustrated in Fig. 12(b) and (c), respectively. The allocation maps of DP recursion and Greedy heuristics are close, thereby achieving similar video quality.

As discussed above, the increase of retry number leads to exponential growth of time cost. Simply setting the retry limit of a packet according to the packet's importance without taking into account the time cost cannot achieve the optimal performance. For performance evaluation, the fixed retry-limit scheme with two different retry limits, the EDF scheme [6], the L-PQ scheme proposed in [18], and the TAR scheme proposed in [22] and [23] were also implemented and compared with the proposed method. The above schemes represent three state-of-the-art strategies of retry limit adaptation in terms of presentation deadlines of packets, packet loss rate of video layers, and frame/packet-level influence, respectively. But none of them considers the estimation of backoff delay and takes



(a)

(b)

(c)

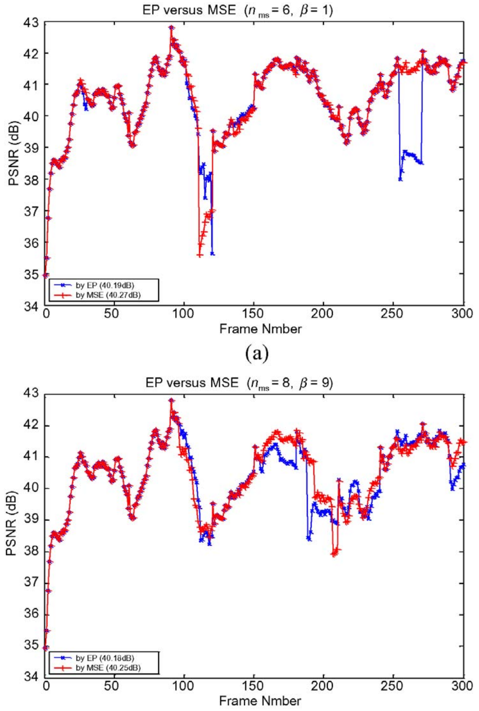
Fig. 14. Frame-by-frame PSNR performance comparisons of four methods for test scenario 2 (i.e., $n_{\text{ms}} = 8$ and $\beta = 9$) for three test sequences. (a) *Foreman*. (b) *Coastguard*. (c) *Salesman*.

into account the backoff delay for packet scheduling. To compare our method with L-PQ, we replaced the layer priority used in L-PQ with our proposed packet loss impact since layered coding is not used in our paper. In our implementation of TAR, the retry extension deadline thresholds are set as

$$R(PKT_{i,j}^{(k)}) = \frac{\beta}{N_{\text{clip}}/N_{\text{GOP}}} \cdot \frac{P_{\text{num}}(F_{i,j}) + 1}{\sum_{j=1}^{N_{\text{GOP}}}(P_{\text{num}}(F_{i,j}) + 1)} \quad (24)$$

where $R(PKT_{i,j}^{(k)})$ stands for the retry extension deadline threshold for the kth packet of the jth frame in the ith GOP, $F_{i,}$ j denotes the jth frame in the ith GOP, and $P_{\text{num}}(F_{i,}\ j)$ stands for the number of frames inter-coded with respect to $F_i,\ j$. The deadline for a retry in the TAR scheme, $D_{\text{TAR}}(PKT_{i,j}^{(k)})$, is then formulated as

$$\begin{aligned} D_{\text{TAR}}(PKT_{i,j}^{(k)}) &= ((i-1)N_{\text{GOP}} + (j-1)) \cdot \lambda \\ &+ \frac{\beta}{N_{\text{clip}}/N_{\text{GOP}}}(i-1) + R(PKT_{i,j}^{(k)}). \quad (25) \end{aligned}$$



(a)

(b)

Fig. 15. Frame-by-frame PSNR performance comparisons of using MSE and the proposed EP for *Salesman* under (a) test scenario 1 ($n_{\text{ms}} = 6$, $\beta = 1$) and (b) test scenario 2 ($n_{\text{ms}} = 8$, $\beta = 9$).

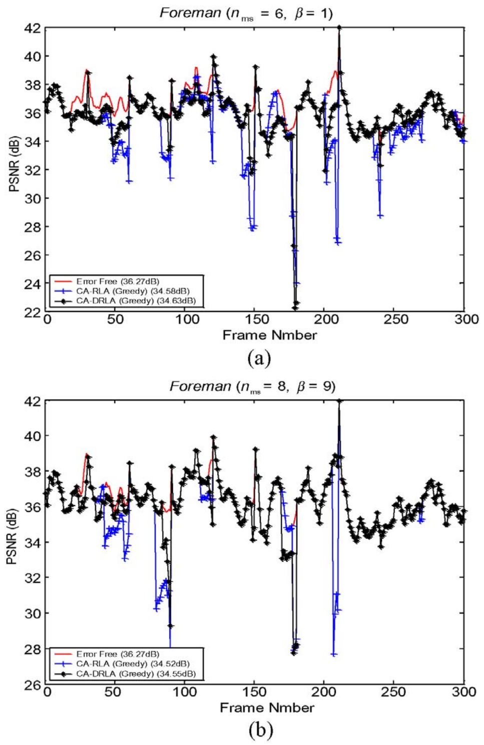
Table V shows the average PSNR performance comparisons of the proposed dynamic CA-RLA methods (with the DP recursion and dynamic Greedy heuristic), the fixed retry limit method and our implementations of TAR and L-PQ for three test sequences under the two test scenarios. In this experiment, ten test channel patterns generated by the OPNET simulator were used to evaluate the average PSNR qualities of each test sequence statistically. The results show that the proposed methods achieve significant PSNR performance improvement over the other methods. The results also show that the PSNR performance difference between the Dynamic-Greedy-based CA-DRLA and the DP-based DP-RLA (DP), is about 0.05–0.22 dB, which is not significant. The frame-by-frame PSNR performances of various adaptive packet retry strategies with one channel pattern under the two test scenarios are compared in Figs. 13 and 14, respectively. By simply evaluating the influence of each frame in a GOP according to the number of frames inter-coded with respect to the frame, the TAR method tends to drop the packets closer to the end of GOP according to the retransmission deadline adaptation within a GOP. The proposed CA-DRLA 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 severe error propagation. The proposed method achieves significant PSNR performance improvements on many corrupted frames by up to several dBs compared to the other schemes.

Since the retry limit of a packet is quantized to a limited range of integers, the estimate of a packet's loss impact is not necessarily as accurate as the actual MSE. As long as

TABLE V
AVERAGE PSNR PERFORMANCE COMPARISONS USING DIFFERENT PACKET RETRY STRATEGIES FOR THREE TEST SEQUENCES UNDER TWO TEST SCENARIOS

		$n_{\mathrm{ms}} = 6, \beta = 1$						
PSNR (dB)	Error Free	2-retry	3-retry	EDF	TAR	L-PQ	CA-DRLA (Greedy)	CA-RLA (DP)
Foreman	36.27	31.17	32.74	33.09	33.65	33.92	34.63	34.64
Coastguard	34.04	30.13	32.39	32.52	32.70	33.02	33.56	33.60
Salesman	40.46	37.29	38.30	39.84	40.03	40.05	40.17	40.20
		$n_{\mathrm{ms}} = 8, \beta = 9$						
PSNR (dB)	Error Free	3-retry	4-retry	EDF	TAR	L-PQ	CA-DRLA (Greedy)	CA-RLA (DP)
Foreman	36.27	31.82	32.12	33.52	34.18	34.25	34.55	34.74
Coastguard	34.04	31.44	31.98	32.65	33.03	33.16	33.58	33.64
Salesman	40.46	39.71	39.81	39.91	40.22	40.28	40.36	30.39

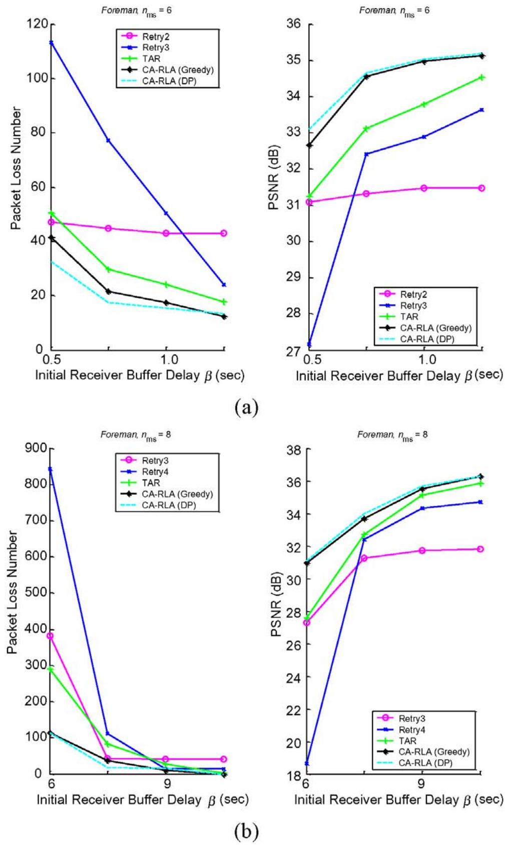
Ten test channel patterns are used to evaluate the average PSNR performance of each test sequence statistically.



(a)

(b)

Fig. 16. Frame-by-frame PSNR performance comparisons of the Greedy-based CA-RLA with and without dynamic reallocation of retry limit under two test scenarios. (a) Scenario 1 (n_{ms} = 6, β = 1), and (b) for scenario 2 (n_{ms} = 8, β = 9) *Foreman*.



(a)

(b)

Fig. 17. Impact of the initial receiver buffer delay β on the number of lost packets and the resulting PSNR performance under two test scenarios. (a) n_{ms} = 6. (b) n_{ms} = 8.

the estimate provides a reasonably good measure of packet's relative importance, it would do a good job in the retry limit adaptation. To evaluate the performance of the proposed EP compared to the MSE, Fig. 15 shows the PSNR performance comparison of retry limit adaptation using the proposed EP and MSE. The average PSNR with the proposed EP is lower than that with MSE by only 0.07–0.08 dB. This justifies that the proposed EP achieves reasonably good performance.

We also compare the PSNR performances of the static Greedy (CA-RLA) and the dynamic Greedy (CA-DRLA) for *Foreman* under the two test scenarios as shown in Fig. 16. Under the two test scenarios, the CA-DRLA achieves slight average PSNR improvement over CA-RLA. The reason why CA-DRLA cannot achieve significant improvement over CA-RLA is that, as shown in the histogram plots of retry numbers for the two test scenarios, most packets can be received successfully within 2 reties under scenario 1 (n_{ms} = 6, β = 1) and 3 reties under scenario 2 (n_{ms} = 8, β = 9), only a small number of packets require more retries than their retry limit. Therefore the number of packets that require to extend their retry limit to get through the channel is few. The improvement from the dynamic reallocation of CA-DRLA will be more significant should the network become more congested.

The impact of the startup delay β on the number of lost packets and the resulting PSNR performance are shown in Fig. 17 for n_{ms} = 6 and n_{ms} = 8, respectively. From the results we can observe that the more the β allocated, the less the number of lost packets caused by expiration of packet presentation deadline. Furthermore, with the TAR and CA-RLA schemes, the extended deadline also allows allocating more retries for packets, thereby further reducing the packet loss rate. The results show that $\beta = 1$ and 9 s are reasonable choices for $n_{\text{ms}} = 6$ and 8 considering the cost with the prolonged initial delay, assuming that the burst length of congested video was assumed to be up to 300 frames (the length of test sequences), corresponding to a 10 s congestion duration. In practical applications, when the network is congested, the startup delay will grow up as the channel bandwidth decreases or as the congestion duration increases.

VII. Conclusion

We have proposed a novel CA-RLA optimization framework 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 set of retry limits 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. We have proposed and evaluated two algorithms based on DP recursion and static/dynamic Greedy heuristics, respectively. 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 traditional static-retry-limit mechanism and the state-of-the-art time-based retry adaptation method in terms of packet loss and visual quality. Besides, the proposed packet scheduling based timeout estimation can further improve the performance.

Appendix

In this appendix, we derive a dynamic programming recursion similar to the approach presented in [16] to determine the optimal retry number assignment. At first, we sort all the EP_{pkt} values in a GOP in a monotonically decreasing order

$$\left\{EP_{\text{pkt}}^{(1)}, EP_{\text{pkt}}^{(2)}, \ldots, EP_{\text{pkt}}^{(k)}, \ldots, EP_{\text{pkt}}^{(N_{\text{pkt}})}\right\} \quad EP_{\text{pkt}}^{(k-1)} \geq EP_{\text{pkt}}^{(k)}.$$

Let x represent the available time budget allowed to send the packets of a GOP, $0 \leq x \leq T_{\text{GOP}}$, and n the maximal retry limit so that $0 \leq L_r^{(k)} \leq n$. We define $f_n(x)$ as the minimal total error propagation under the time constraint x

$$f_n(x) = \min_{L_r^{(1)}, L_r^{(2)}, \ldots, L_r^{(k)}, \ldots, L_r^{(N_{\text{pkt}})}} \left\{ \sum_{k=1}^{N_{\text{pkt}}} P_e^{L_r^{(k)}+1} \cdot EP_{\text{pkt}}^{(k)} \right\}$$

subject to $\sum_k T(L_r^{(k)}, P_e) \leq x$ and $0 \leq L_r^{(k)} \leq n$

where $T(L_r^{(k)}, P_e)$, as defined in (19), is the average transmission time of kth packet with a retry limit $L_r^{(k)}$ and packet loss rate P_e.

When $n = 0$, we have

$$f_0(x) = \min \left\{ \sum_k P_e \cdot EP_{\text{pkt}}^{(k)} + \sum_{k'} EP_{\text{pkt}}^{(k')} : \sum_k T(0, P_e) \leq x \right\}$$

with $k = 1, 2, \ldots, \min\left\{N_{\text{pkt}}, (x/T(0, P_e))\right\}$ and $k' = (x/T(0, P_e)) + 1, \ldots, N_{\text{pkt}}$. Then we can evaluate f_1 in terms of f_0

$$f_1(x) = \min \left\{ \sum_k P_e^2 \cdot EP_{\text{pkt}}^{(k)} + f_0(x - \sum_k T(1, P_e)) \right\}$$

with $k = 0, 1, 2, \ldots, \min\left\{N_{\text{pkt}}, (x/T(1, P_e))\right\}$.

The dynamic programming recursion becomes

$$f_n(x) = \min \left\{ \sum_k P_e^{n+1} \cdot EP_{\text{pkt}}^{(k)} + f_{n-1}(x - \sum_k T(n, P_e)) \right\}$$

where $k = k_n(x)$ with a range of $0 \leq k \leq \min\left\{N_{\text{pkt}}, (x/T(n, P_e))\right\}$. Under the constraint $\sum_{k=1}^{N_{\text{pkt}}} T(L_r^{(k)}, P_e) \leq T_{\text{GOP}}$, and the maximum retry limit n, the dynamic programming approach determines the optimal solution $f_n(T_{\text{GOP}})$ in a coherent way by evaluation $f_0(x)$, $f_1(x)$, $\ldots$, $f_{n-1}(x)$ with a time complexity of $O\left(\sum_{i=0}^{n} \prod_{j=0}^{i} \left(\min\left(N_{\text{pkt}}, T_{\text{GOP}}/T(n-j, P_e)\right)\right)\right)$.

References

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

[2] W. Kumwilaisak, J. Kim, and C.-C. J. Kuo, "Reliable wireless video transmission via fading channel estimation and adaptation," in *Proc. IEEE Wirel. Commun. Netw. Conf.*, Sep. 2000, pp. 185–190.

[3] P. A. Chou, A. E. Mohr, A. Wang, and S. Mehrotra, "Error control for receiver-driven layered multicast of audio and video," *IEEE Trans. Multimedia*, vol. 3, no. 1, pp. 108–122, Mar. 2001.

[4] 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, Jun. 2002.

[5] M. Zorzi, R. R. Rao, and L. B. Milstein, "ARQ error control for fading mobile radio channels," *IEEE Trans. Vehicular Technol.*, vol. 46, no. 2, pp. 445–455, May 1997.

[6] T. Hasegewa, T. Kato, and K. Suzuki, "A video retrieval protocol with video data prefetch and packet retransmission considering playout deadline," in *Proc. IEEE Int. Conf. Network Protocols*, 1996, pp. 32–39.

[7] L. Georgiadis, R. Guerin, and A. Parehk, "Optimal multiplexing on a single link: Delay and buffer requirements," *IEEE Trans. Information Theory*, vol. 43, no. 5, pp. 1518–1535, Sep. 1997.

[8] A. Dua and N. Bambos, "Downlink wireless packet scheduling with deadlines," *IEEE Trans. Mobile Comput.*, vol. 6, no. 12, pp. 1410–1425, Dec. 2007.

[9] P. A. Chou and Z. Miao, "Rate-distortion optimized streaming of packetized media," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 390–404, Apr. 2006.

[10] T. Ren, I. Koutsopolous, and L. Tassiulas, "QoS provisioning for real-time traffic in wireless packet networks," in *Proc. IEEE Global Telecomm. Conf.*, Nov. 2002, pp. 1673–1677.

[11] 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, Jun. 2002.

[12] S. H. Kang and A. Zakhor, "Effective bandwidth based scheduling for streaming media," *IEEE Trans. Multimedia*, vol. 7, no. 6, pp. 1139–1148, Dec. 2005.

[13] J. Chakareski and P. Frossard, "Rate-distortion optimized distributed packet scheduling of multiple video streams over shared communication resources," *IEEE Trans. Multimedia*, vol. 8, no. 2, pp. 207–218, Apr. 2006.

[14] M. van der Schaar, S. Krishnamachari, S. Choi, and X. Xu, "Adaptive cross-layer protection strategies for robust scalable video transmission over 802.11 WLANs," *IEEE J. Sel. Areas Commun.* vol. 21, no. 10, pp. 1752–1763, Dec. 2003.

[15] C. Chan, J. Apostolopoulos, Y. Li, and N. Bambos, "Receiver-based optimization for video delivery over wireless links," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, pp. 861–864.

[16] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video streaming over wireless local area networks through real-time retry limit adaptation," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 278–290, Apr. 2004.

[17] G. Liebl, M. Kalman, and B. Girod, "Deadline-aware scheduling for wireless video streaming," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jul. 2005, p. 4.

[18] M. van der Schaar, D. S. Turaga, and R. Wong, "Classification-based system for cross-layer optimized wireless video transmission," *IEEE Trans. Multimedia*, vol. 8, no. 5, pp. 1082–1095, Oct. 2006.

[19] 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*, Jul. 2005, pp. 723–726.

[20] M.-H. Lu, P. Steenkiste, and T. Chen, "A time-based adaptive retry strategy for video streaming in 802.11 WLANs," *Wirel. Commun. Mob. Comput.*, vol. 7, pp. 187-203, Feb. 2007.

[21] 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, nos. 4–5, pp. 563–588, Aug.–Oct. 2005.

[22] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 966–976, Jun. 2000.

[23] K. Stuhlmüller, N. Färber, M. Link, and B. Girod, "Analysis of video transmission over lossy channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 6, pp. 1012–1032, Jun. 2000.

[24] S. Rane, P. Baccichet, and B. Girod, "Systematic lossy error protection of video signals," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 10, pp. 1347–1360, Oct. 2008.

[25] W. Kumwilaisak, Y. T. Hou, Q. Zhang, W. Zhu, C.-C. J. Kuo, and Y. Q. Zhang, "A cross-layer quality of service mapping architecture for video delivery in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 21, no. 10, pp. 1685–1698, Dec. 2003.

[26] M. Ergen and P. Varaiya, "Throughput analysis and admission control for IEEE 802.11a," *Mobile Netw. Applicat.*, vol. 10, pp. 705–716, Oct. 2005.

[27] G. Bianchi, "Performance analysis of the IEEE 802.11 distributed coordinated function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, pp. 353–547, Mar. 2000.

[28] S. Ross, *Probability*, 5th ed. Englewood Cliffs, NJ: Prentice-Hall, 1998.

[29] J. Bather, *Decision Theory*. New York: Wiley, 2000.

[30] G. Cheung, W.-T. Tan, and C. Chan, "Reference frame optimization for multipath video streaming with complexity scaling," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 6, pp. 649–662, Jun. 2007.

[31] OPNET [Online]. Available: http://www.opnet.com

[32] *ISO/IEC 14496-7 Information Technology—Coding of Audio-Visual Objects—Part 7: Optimized Reference Software for Coding of Audio-Visual Objects*, ISO/IEC Std., Ver. Microsoft-FDAM1-2.3-001213.

Chih-Ming Chen (M'08) was born in Changhua, Taiwan, in 1969. He received the B.S. degree in electronic engineering from Feng Chia University, Taichung, Taiwan, in 1991, and the M.S. and Ph.D. degrees in electrical engineering from National Tsing Hua University, Hsinchu, Taiwan, in 1993 and 2007, respectively.

Since 1993, he has been with the Telecommunications Laboratories, Chunghwa Telecom Company, Ministry of Transportation and Communications, Taoyuan, Taiwan. He has authored or co-authored about 30 technical papers and holds 17 patents. His current research interests include video networking and wireless sensor networks.

Chia-Wen Lin (S'94–M'00–SM'04) received the Ph.D. degree in electrical engineering from National Tsing Hua University (NTHU), Hsinchu, Taiwan, in 2000.

He is currently an Associate Professor with the Department of Electrical Engineering, NTHU. From 2000 to 2007, he was with the Department of Computer Science and Information Engineering, National Chung Cheng University (CCU), Minhsiung, Taiwan. Prior to joining academia, he was with the Information and Communications Research Laboratories, Industrial Technology Research Institute, Hsinchu, from 1992 to 2000, where his final post was Section Manager. From April 2000 to August 2000, he was a Visiting Scholar with the Information Processing Laboratory, Department of Electrical Engineering, University of Washington, Seattle. From July 2002 to August 2002, he was a Visiting Professor with Microsoft Research Asia, Beijing, China. He has authored or co-authored over 90 technical papers. He holds more than 20 patents. His current research interests include video networking and video content analysis.

Dr. Lin is an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY and the *Journal of Visual Communication and Image Representation*. He has served as a Guest Co-Editor of two special issues for the *EURASIP Journal on Advances in Signal Processing* and the *Journal of Visual Communication and Image Representation*. He is a Technical Program Co-Chair of the IEEE International Conference on Multimedia and Expo (ICME) in 2010, and was a Special Session Co-Chair of the IEEE ICME in 2009. He was a recipient of the 2001 Ph.D. Thesis Award presented by the Ministry of Education, Taiwan. His paper won the Young Investigator Award presented by SPIE VCIP 2005. He received the Young Faculty Award presented by CCU in 2005 and the Young Investigator Award presented by the National Science Council, Taiwan, in 2006.

Yung-Chang Chen (F'05) received the BSEE and MSEE degrees from National Taiwan University, Taipei, Taiwan, in 1968 and 1970, respectively, and the Dr.Ing. degree from Technische Universitaet, Berlin, Germany, in 1978.

Since 1978, he has been with the Department of Electrical Engineering, National Tsing Hua University (NTHU), Hsinchu, Taiwan, where he is currently a Professor and the Director of the Multimedia Research Center of NTHU. He served as the Founding Chairman of the Department of Electrical Engineering, National Central University, Chungli, Taiwan, from 1980 to 1983. From 1992 to 1994, he served as the Chairman of the EE Department, NTHU. He served as the Dean of the College of Engineering, National Chung Cheng University, Chiayi, Taiwan, from February 2002 to July 2003.

Dr. Chen received the Outstanding Research Award from the National Science Council from 1991 to 1994, the Outstanding Electrical Engineering Professor Award from the Chinese Institute of Electrical Engineers in 2000, and the Industrial Collaboration Award from the Ministry of Education in 2001. From 2001 to 2003, he served as the Chair of the IEEE Consumer Electronics Society, Taipei Chapter, and the Representative of the Greater Chinese Region for the IEEE International Conference on Consumer Electronics. He was the General Chair of the 2002 IEEE Pacific-Rim Conference on Multimedia and the IEEE Pacific-Rim Symposium on Image and Video Technology in 2006.