

MAC-Layer Active Dropping for Real-time Video Streaming in IEEE 802.16 Networks

James She, Fen Hou, Basem Shihada and Pin-Han Ho

Abstract— This paper introduces a MAC-layer Active Dropping (AD) scheme for achieving effective resource utilization and maintaining application-level quality of services in real-time video streaming over the emerging wireless broadband access networks based on time-division multiple access (TDMA). By proactively dropping the MAC-layer protocol data units (MPDUs) of a video frame that are unlikely to meet its application-layer delay bound, the proposed scheme releases the precious transmission resources to the subsequent frames or other competing service streams. This leads to more efficient resource utilization than that by the conventional prioritization-based cross-layer approaches which simply manipulate transmission and/or retransmission priority of each MPDU. Furthermore, an analytical model is developed to evaluate how confident a video frame can be effectively delivered within its application-layer delay bound, where the effects of time-varying wireless channel, requirement of bitstreams decodability in each video frame, retransmission of lost data, and playback buffer at the recipient software in a WiMAX network are jointly considered. Extensive simulations are conducted to illustrate the effectiveness of the proposed scheme.

Index Terms— Dropping, Application-layer Delay, IEEE 802.16d/e, WiMAX, Real-time Video Streaming, Cross-Layer Design

I. INTRODUCTION

THE advancements in miniaturization of powerful mobile devices, affordable wireless communication equipments, and improved small-scale energy supplies, have been integrated with the rapid development of wireless broadband technologies, such as IEEE 802.16d/e (also known as WiMAX), and WCDMA, to make broadband wireless access (BWA) networks possible [1]. Among the emerging BWA technologies, the IEEE 802.16 technology has attracted extensive attentions from both industry and academia due to its strong ability of provisioning real-time multimedia services through the OFDAM/TDMA multiple access technique, and has been in a leading position in its early deployments around the world. With the capability of broadband wireless capacity,

QoS supports and mobility, WiMAX has been recognized as a promising and/or complementary access network technology when compared to legacy cable wirelines, Digital Subscriber Lines (DSL) and 802.11a/b/g (or Wi-Fi) for all-IP heterogeneous broadband wireless access services in the metropolitan area networks. Such BWA platform can further extend many existing IP-based streaming video applications into wireless and mobile realms. Many emerging video services using wireless or even with the mobility are cultivated such as IPTV over WiMAX, mobileTV, in-vehicle IPTV, on-demand roadside video training, etc.

Real-time video streaming is a widely employed yet highly demanding component in the abovementioned multimedia services due to the delay-sensitive and bandwidth-intensive video data. With a stringent application-layer delay bound on the arrival of each video frame imposed by the recipient, a late arrival frame yields an equivalent impact to the recipient as if the frame is lost. The problem becomes more serious and challenging in such BWA scenarios given the demanding expectations of high data rate and mobility supports, where MAC-layer Protocol Data Units (MPDU) of consecutive video frames in a real-time video stream could be lost or delayed. This is caused by the time-varying capacity and bit error rate of a wireless channel due to the fading and mobility effects. Although specific traffic classes of services (i.e. UGS and rTPS) for dedicated access control and radio resources allocation are defined in WiMAX, the base station (BS) may still launch a MPDU of which the corresponding video frame is unlikely to arrive at the recipient software within the required application-layer delay bound. In this case, not only the precious radio transmission resources are wasted for sending a late MPDU, but also the subsequent MPDUs would experience the delay propagations, which could eventually lead to the disruptions on the perceived video quality of more than just one video frame and also all the existing real-time streaming video services.

To reduce the late delivery of video frames in real-time wireless video streaming, numerous studies through a cross-layer design approach have been reported in the literature. The study in [2] proposed to schedule packet transmissions over orthogonal frequency-division multiplexing (OFDM) channels by giving a higher priority to more important packets (such as those belonging to I-frames which affect the quality significantly in a video stream). It also allocates an appropriate number of OFDM sub-carriers to the user by jointly prioritizing with the size of the queues. In [3], an opportunistic scheduling algorithm for multiple video streams using a

Manuscript received March 1, 2010. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada. James She is with the Computer Laboratory, William Gates Building, University of Cambridge, 15 JJ Thomson Avenue, Cambridge, CB3 0FD. He was with the the Electrical & Computer Engineering, University of Waterloo, 200 University Ave. W., Waterloo, ON, N2L3G1, Canada (phone: +1 519-888-4567; e-mail: james@bcr.uwaterloo.ca). Basem Shihada is with the Computer Science, King Abdullah University of Science & Technology, Jeddah 21534, Saudi Arabia. Fen Hou and Pin-Han Ho are with the Electrical & Computer Engineering, University of Waterloo, 200 University Ave. W., Waterloo, ON, N2L3G1, Canada.

priority function was developed based on channel conditions, importance of frames, queue size, and multiplexing gain. A packetization scheme was introduced in [4], which incorporates with forward error correction (FEC) codes at radio link protocol (RLP) packet level rather than across different application packets. A priority-based automatic repeat request (ARQ) scheme was applied at the application layer to retransmit the corrupted RLP packets only. The scheme in [5] prioritizes retransmission opportunities for each MPDU based on the joint weighting of the perceptual importance and urgency of each individual MPDU upon the video frame. In [6], the best source and channel coding pair is selected to encode and transmit the video signals with efficient packetization and error concealment techniques altogether according to the channel information.

It was pointed out that a cross-layer QoS mapping architecture for video delivery in wireless networks is critical, in order to coordinate and achieve effective adaptation of QoS parameters in the application layer and priority transmission system [7]. However, the abovementioned cross-layer approaches suffer from the committed consideration on the application-layer delay bound of each video frame. This task is very complicated because the QoS mapping onto the subjective video quality with an effective utility functions could be computationally intractable and impractical. Even though the delay of each MPDU is considered in a cross-layer approach such as in [5], the result by a priority-based optimization would still consume transmission resources on sending MPDUs for a late arrival video frame. In such situation, it is natural to consider dropping those MPDUs to preserve the resources rather than prioritizing their transmission through a cross-layer approach. Some previous attempts of dropping packets for meeting real-time requirements are proposed. A scheme of priority-drop for on-demand streaming video is introduced in [8] for the best effort computing and networking environments, which drop packets gracefully with priorities based on a pre-proceed mapping between each packet and its impacts towards the overall video quality. However, the upfront processing for mapping those priorities of video packets and their impacts to the overall video quality is not computationally efficient and limited by many practical system deployments, especially for real-time video streaming in a BWA network. In [9], excess real-time packets are attempted to be transmitted with the delay guarantee through an efficient coordinated buffer and scheduling management scheme, in which expired packets will be simply dropped in the queues when the buffer is too full to accommodate newly arrived real-time traffic. Although this can relax the constraint of limiting the unpredictable rate of multimedia traffic while maintain the committed statistical delay, the dropping is not done proactively until the buffer is full, where the characteristic of tolerable loss in video coding has never been explored. In [10], a dropping scheme was proposed, where its effectiveness for streaming video delivery with fine-grained adaptation to the transmission channel is

demonstrated by introducing a delayed time-window as little as 400 milliseconds at the sender before the transmission. All data packets with timestamps within a certain period of time are placed in the time-window and reordered into priority order for transmission. It then transmits these packets for the time duration of the window only. At the end of the window duration, it discards unsent packets and moves on to the next window. In this way, the available bandwidth from the channel is used to send the most important elements of the video stream and the least important elements are dropped. However, the scheme is designed to apply only on the sender side in a wired network without the consideration of the playback-buffer at the recipient software as well as the impact of retransmitting lost packets.

It is clear that aforementioned strategies have never considered the impacts on the delay performance and video quality induced by wireless channel fluctuation, and very few of them have jointly taken the effects of recipient-side playback-buffer on the application-level delay bound into account, which however, is known to play a critical role in any video streaming system. Note that the application-layer delay bound and the maximum loss tolerance of a frame are considered as two common parameters easily available from the adopted video coding, which are envisioned to take an important factor in the design of any future scalable video coding technique and other cross-layer strategies. An initial active dropping approach using a cross-layer design without any intended delay at the video source is introduced in our previous work [11] to address the missing issues above. The scheme is designed to work in the transmission buffer at a wireless broadband transmitter, instead of the video source in a wired network. Promising results from simulations have showed the effectiveness of this approach. We are interested to extend the work into a more completed framework with a video quality measure using the number of decodable frames as well as the discussions of the possible implementation.

In this paper, we present a MAC-layer Active Dropping (AD) scheme for real-time video streaming in WiMAX-based BWA networks with a stringent delay bound of each video frame. The developed AD scheme has the following features - 1) The scheme guarantees an application-level delay bound on each video frame with minimal consumed resources; 2) the scheme enforces a graceful video quality degradation by manipulating a probabilistic confidence threshold, which is taken for making a dropping decision on a possibly late video frame; and 3) the scheme makes the MPDUs of a video frame delivered without changing their orders. To achieve the above design features, an analytical model is developed and employed in the proposed scheme in order to determine the confidence of meeting the application-layer delay bound of each video frame. Based on a probabilistic evaluation, the proposed AD scheme can proactively drop the MPDUs of a video frame when it is considered not confident enough to arrive at the recipient by its application-layer delay bound for decoding. Thus, the precious transmission resources can be

released for the subsequent frames or other competing video streams, by which the revenue potential is maximized without losing the same application-layer QoS conditions such as frame loss rate. In addition to the application-layer delay bound of each video frame, the wireless channel condition, common video coding parameters, the recipient-side playback buffer and any possible MAC-layer retransmission mechanism, such as Automatic Retransmission reQuest (ARQ), are the essential cross-layer design factors being jointly considered in a dropping decision for real-time streaming video in WiMAX networks. Extensive simulations are conducted to demonstrate the efficiency of the proposed scheme in terms of application-level QoS satisfaction and resource utilization.

The rest of the paper is organized as follows. The system model is described in Section II. Section III presents the proposed AD scheme with a comprehensive analytical framework, and then the implementability is also discussed. The effectiveness of the proposed scheme is proved with simulations and numerical results in Section IV, followed by the conclusions.

II. SYSTEM MODEL

A. System Architecture

Without loss of generality, the proposed scheme is developed based on IEEE 802.16d/e or WiMAX (PMP mode), in which transmission resources are in terms of slotted times by way of time-division multiple access (TDMA) with all sub-carriers allocated in each timeslot. The traffic class of UGS in WiMAX is associated for the data of a video stream, where dedicated number of transmission resources are assigned in each physical downlink sub-frame. Fig. 1 illustrates the system architecture, which includes three components: (1) a media server; (2) the base station (BS) with a set of logical queues corresponding to each type of service for all mobile and fixed subscriber stations (SSs); and (3) the recipient with a built-in playback buffer at the corresponding mobile or fixed SS. They are defined with the following characteristics: i) Real-time video is encoded at the media server into streaming video bitstreams, in which the bitstream of each video frame is generated every P timeslots with possibly a variable size, and is further packetized into IP packets; ii) IP packets of a video frame are immediately available to a corresponding transmission queue in the BS with negligible latency, and are further packetized into a number of L MPDUs for sending over the wireless channel; iii) At the beginning of the video transmission, a small number of video frames, denoted as Δ , can be received accumulatively in the playback buffer at the recipient before the first one is displayed; iv) Each video frame generated by the media server should be played at the recipient in D timeslots later, where D is a time duration determined by Δ and P , and is the maximum application-layer delay bound of each video frame that can be tolerated before its playback. In other words, any frame received by the recipient longer than D timeslots after it was produced at the media server will be classified as a late-arrival frame for playback and treated as a frame loss. Obviously, both Δ and P are the system design

parameters, and the larger Δ and P are, the larger D could be tolerated. v) The received video frames are expected to be available in the playback buffer and retrieved by the recipient at a scheduled playback rate (i.e., every P timeslots per frame); otherwise, a frame loss event is perceived; vi) A perfect and instantaneous feedback is in place to initiate a retransmission if the transmission of a MPDU from the BS failed and the retransmission is required. This feature is optional for real-time video streaming but is included in the proposed scheme for the sake of completeness.

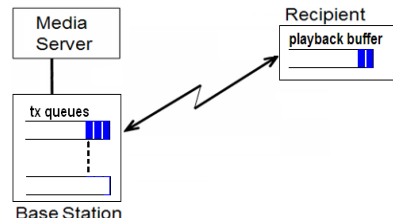


Fig. 1. The considered system model.

B. Recipient-side Playback Buffer

Without loss of generality, we assume the frame playback rate at the recipient is the same as the frame-producing rate at the media server (i.e., every P timeslots per frame). A playback buffer is built in the recipient media software and serves as a reservoir for mitigating the vicious impact due to fluctuations of the communication latency [16]. With such playback buffer, a number of frames, Δ , can be accumulated in it before the head of line (HoL) frame is played back. The value of Δ is set very small for real-time video applications, such as 1-5 frame(s), and represents the maximum delay bound, D , that the system can tolerate for waiting a video frame newly transmitted from the BS. In case no playback buffer is available (or $\Delta=0$), D will be equal to P , which means a frame can only spend a duration no longer than P to reach the recipient for its playback without a frame loss event. In summary, the maximum application-layer delay for a video frame, denoted as D , can be formulated as,

$$D = (\Delta + 1) \times P \quad (1)$$

where Δ is the number of frames already buffered in the playback buffer when the first frame is being played back at the recipient.

C. Packetizations of Video Frames, Decodable Frame Rate and Video Quality

In the MPEG standard [12] for video streaming, the generated video frames do not have the same significance with respect to the video quality because some frames are dependent on the others. Standard MPEG encoders generate three types of compressed video frames, referred to as I-, P-, and B frames. An I-frame is intra-coded without any dependence on the other frames, while P- and B-frames are coded with forward and bidirectional predictions respectively. Undoubtedly, I-frames are the most important, followed by P-frames and B-frames. After being generated, a video frame is packetized into multiple Real-Time Protocol (RTP) datagrams,

each being further packetized again into multiple IP packets. The use of RTP is optional. Each IP packet is then segmented into a set of MPDUs at the BS.

Given a scheduling policy, one or multiple MPDUs will be scheduled in the time of a downlink sub-frame for transmitting over the wireless channel. The lossy wireless channel can introduce unpredictable bandwidth and loss of MPDUs due to the fluctuating channel transmission capacity and bit error rates, which may impair the arrival time of video frames and their video quality perceived by the recipient. The impairment on the perceptual video quality becomes more serious when the channel condition gets worse and/or when the system transmission resources are heavily demanded. To evaluate the quality of a video source, we adopt an evaluation model of video quality (Q) from [13], which is defined by the number of decodable frames over the total number of frames originally in the video source:

$$Q = \frac{N_{dec}}{N_{total-I} + N_{total-P} + N_{total-B}}, \quad (2)$$

where $0 < Q < 1$, N_{dec} is the expected total number of decodable frames including all the types of frame (i.e., $N_{dec} = N_{dec-I} + N_{dec-P} + N_{dec-B}$). Noted that the dependencies between different types of frames (i.e., I, P, and B frames) are already considered in the derivation of the numbers of different decodable frames (i.e., N_{dec-I} , N_{dec-P} , and N_{dec-B}), which is given in Appendix A.1-3. $N_{total-I}$, $N_{total-P}$ and $N_{total-B}$ is the total number of I-, P-, B-frames in the video source, respectively. Q is an objective measure to evaluate the video quality. The larger Q means the better video quality is perceived by the recipient. Although evaluating video quality through an actual human perception, such as MOS (mean opinion scores), can be adaptive to the content of the video and yield more persuasive results, it is however not the focus of our study here. According to [14], an objective quality measure, such as Eq. (2) employed here, can still effectively provide a lower bound of the quality measure because MPEG streams tend to recover from partial losses of frames provided by their temporal and spatial redundancy.

Furthermore, a frame is only considered to be decodable at the recipient with the acceptable video quality when at least a minimum fraction of the total MPDUs of the frame is received [15] due to the video decoding requirement and adopted error recovery mechanisms. Otherwise, the frame is not acceptable by the recipient even if it is received within the application-layer delay bound. Let $x\%$ be the statistical maximum fraction of bitstreams in a video frame that can be tolerated to lose. This parameter depends on the video coding technique and the video quality required by the real-time streaming video applications. Hence, the statistical minimal amount of MPDUs of a video frame, which must be received by the recipient within its application-layer delay bound, is denoted as L_i' , and can be formulated as:

$$L_i' = L_i \times (100 - x)\% \quad (3)$$

where L_i is the total number of MPDUs of frame i . In other words, receiving a number of L_i' out of the total L_i MPDUs of frame i within its application-layer delay bound serve as the decodable requirement for an acceptable video quality and a successful delivery of the frame. If the available transmission resources can send all the L_i MPDUs of frame i , the recipient will perceive the optimal video quality; otherwise, at least L_i' MPDUs must be delivered within the corresponding delay bound in order to make the frame decodable with the minimal video quality. Assumed that each MPDU consumes a single timeslot to transmit, at least L_i' timeslots are required for sending frame i . Number of timeslots available for actual transmissions within the corresponding delay bound of frame i is indeed affected by various factors in a WiMAX-based BWA network, such as the wireless channel condition, the size of frame, the recipient's playback buffer size etc., which will be further described in the rest of this section.

D. Modeling of Available Transmission Timeslots

A BS has a scheduling cycle for allocating the time resources to different queues, which is denoted as S for the duration of each cycle. The duration of a timeslot is assumed to be long enough to send the data of each complete MPDU. When MPDUs of a frame arrive at the BS, they will be buffered in a transmission queue first. If no MPDU(s) of the previous frame exists in the queue, the MPDUs of this newly arrived frame will be served and transmitted in the coming scheduling cycle with the fixed amount of allocated transmission timeslots. Otherwise, they will wait for transmissions until they become the HoL in the queue. Each MPDU of a frame will therefore experience a delay within each scheduling cycle due to three factors: (1) the inter-service time T_{in} , where the scheduler is serving other queues; (2) the waiting time when the scheduler is transmitting the MPDUs of previous frames in the queue; (3) the time for transmitting its own MPDUs. Therefore, the actual available transmission timeslots for sending the MPDUs of frame i currently remained in the queue at time t in the scheduling cycle j can be derived as:

$$m(i, t, j) = D - T_w(i, t, j) \quad (4)$$

where t is the universal time counted from zero in the BS, j is the index of the current scheduling cycle starting from one, D is the application-layer delay bound of frame i defined in Eq. (1), $T_w(i, t, j)$ represents the total waiting time that the remaining MPDUs of frame i need to experience in the queue estimated at time t in the scheduling cycle j , and can be derived by:

$$T_w(i, t, j) = (j - i) \times S + (\Delta + 1 - (j - i)) \times T_{in} + T_{leftover}(i, t, j) + T_x(i, t, j) \quad (5)$$

where the 1st term on the right hand side is the number of scheduling cycles that frame i has experienced in the queue at time t if it has taken more than one scheduling cycle to send preceding MPDUs of frame i ; the second term describes the total of inter-service time, T_{in} , that frame i has to wait; $T_{leftover}(i, t, j)$ is the time consumed for transmitting MPDUs of previous frames in the scheduling cycle j , and $T_x(i, t, j)$ is the

total timeslots expended so far for successful/unsuccessful transmissions of MPDUs of frame i in the scheduling cycle j . Both $T_{\text{leftover}}(i, t, j)$ and $T_x(i, t, j)$ are parameters commonly tracked in the system after each transmission, whereas, T_{in} and S are known from the adopted scheduling policy. The following is an example showing their relationships. In case $\Delta=1$, the application-layer delay bound, D , of the current frame is equal to $2P$ according to Eq. (1). To further clarify Eqs. (4) and (5), Fig. 2 presents all possible scenarios at time instants t_1 , t_2 and t_3 that all MPDUs of a frame can be sent successfully with the transmission timeslots available within D . Since the rate of a real-time video streaming traffic is varying, it is not trivial that the duration S of a scheduling cycle is optimal with a particular value, and also it will be adjusted regularly depending on the scheduling, call-admission and resources provisioning policies based on the changing services commitments. For the simplicity but without loss of generality, we assume that the duration of S can be equal to the duration of frames-generation at the media server (i.e., $S=P$) in Fig. 2.

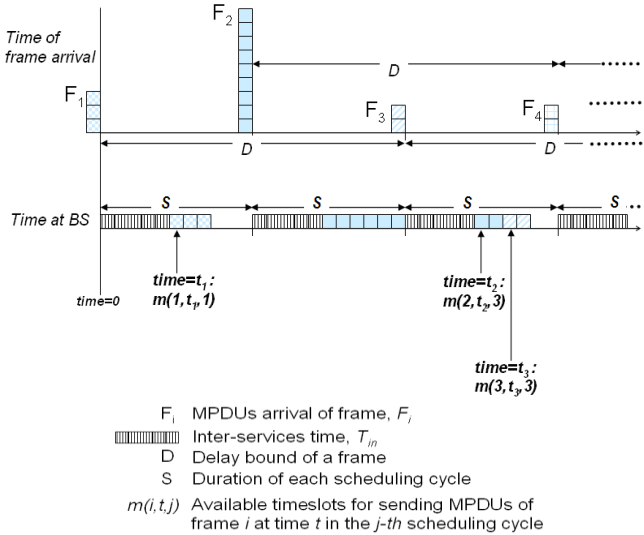


Fig. 2. All possible scenarios that MPDUs of a frame are sent completely

At time t_1 : No MPDU of previous frames are in the transmission queue when video frame F_1 arrives at the BS, the waiting time within the delay bound, D , of F_1 is simply consumed by the inter-service time T_{in} in the 1st and 2nd scheduling cycles. The total transmission timeslots available for sending the MPDUs of F_1 at time t_1 in the 1st scheduling cycle is:

$$m(1, t_1, 1) = D - T_w(1, t_1, 1) = D - 2T_{in}$$

At time t_2 : All MPDUs of video frame F_2 could not be sent within the 2nd scheduling cycle. Since the frame has a delay bound, D , with the duration of $2P$, it allows the remaining MPDUs of F_2 to be sent at time t_2 in the 3rd scheduling cycle with the available transmission timeslots derived below:

$$m(2, t_2, 3) = D - T_w(2, t_2, 3) = D - (S + T_{in})$$

At time t_3 : The MPDUs of video frame F_3 will be sent completely right after all the leftover MPDUs of F_2 are successfully sent over first two timeslots in the 3rd scheduling cycle. The total available transmission timeslots for sending MPDUs of F_3 at time t_3 in the 3rd scheduling cycle is derived below with the consideration of the leftover MPDUs of F_2 :

$$m(3, t_3, 3) = D - T_w(3, t_3, 3) = D - (2T_{in} + 2)$$

These three scenarios summarized the dynamics between the waiting time, $T_w(i, t, j)$ and the total available transmission timeslots $m(i, t, j)$ within the corresponding delay bound, D , of a frame F_i , as well as the impact of the playback buffer.

III. THE ACTIVE DROPPING MECHANISM

By Eq. (3), L_i' is the minimal number of MPDUs of frame i required by the recipient within the delay bound D for the decodable and minimal video quality requirements. Although the arrival of L_i' MPDUs of frame i can be guaranteed by a particular link-layer retransmission policy such as ARQ, retransmitting a lost MPDU certainly take over the precious transmission opportunities from the later MPDUs. It increases the risk of late arrival not only for the current frame, but also the subsequent frames due to the delay propagation. Even though transmission/retransmission of these L_i' MPDUs can be normally optimized through prioritizations with other factors in a cross-layer approach, it is however reasonable to drop all the rest of MPDUs of the frame from the transmission queue when it is believed that sending these L_i' MPDUs successfully within D timeslots is not likely to happen. This is taken as the fundamental motivation of the proposed Active Dropping scheme. We further develop the idea with an analytical approach, where a probabilistic confidence value is manipulated by jointly considering the factors in a WiMAX network that have not been comprehensively addressed in the previous literatures.

A. Active Dropping Mechanism

The dropping decision is based on the confidence of successfully sending $L_i(t)$ MPDUs of frame i in the queue within $m(i, t, j)$ available transmission timeslots at time t . In other words, before sending a MPDU, a confidence value at time t , denoted as $C(L_i(t), m(i, t, j))$, is evaluated by two deterministic information - the current values of $m(i, t, j)$ and $L_i(t)$. The value of $L_i(t)$ is the number of MPDUs that is still required to be sent for frame i at time t . Note that the initial value of $L_i(t)$ is set as the constant L_i' defined in Eq. (3) for frame i . Transmitting MPDUs at time t will continue only if the confidence value, $C(L_i(t), m(i, t, j))$, is not below a confidence threshold, C_{th} . The retransmission limit is assumed infinite for simplicity, but it can also be a finite value for our modeling here. For every MPDU transmission/retransmission, the value of $m(i, t, j)$ is reduced by one at time $t + 1$, and each successful MPDU transmission/retransmission updates $L_i(t + 1)$ by reducing one MPDU from $L_i(t)$ at time $t + 1$, i.e.,

$$L_i(t+1) = \begin{cases} L_i(t) - 1, & \text{for successful delivery of a} \\ & \text{MPDU at the timeslot } t; \\ L_i(t), & \text{otherwise.} \end{cases}$$

When $L_i(t)$ becomes zero, it will be set as the value of $L_i - L_i'$ for the purpose of sending more than L_i' MPDUs for frame i if the resource allows. However, regardless of the value of $L_i(t)$, when the confidence value $C(L_i(t), m(i, t, j))$ is less than C_{th} at any time instant, it is taken as a strong statistical indication that frame i will not be delivered and decodable for the minimal video quality with the available $m(i, t, j)$ transmission timeslots in the associated delay bound. The BS therefore immediately drops all the remaining MPDUs of frame i in the queue and the remaining $m(i, t, j)$ transmission timeslots can be released to any subsequent frames or other queues. Fig. 3 illustrates a flowchart of the proposed AD scheme. We justified that it is a reasonable assumption that there is no difference leading to a frame loss event between the cases that required number of MPDU are dropped or delayed due to the lack of transmission time resources and that those MPDU are dropped proactively due to a lower-than-threshold probabilistic confidence for sending them before the delay bound.

In summary, the proposed AD scheme proactively drops those MPDUs of a possibly late arrival frame and releases precious transmission opportunities such that the delivery of each video frame can be bounded within its application-layer delay bound with a probabilistic confidence threshold.

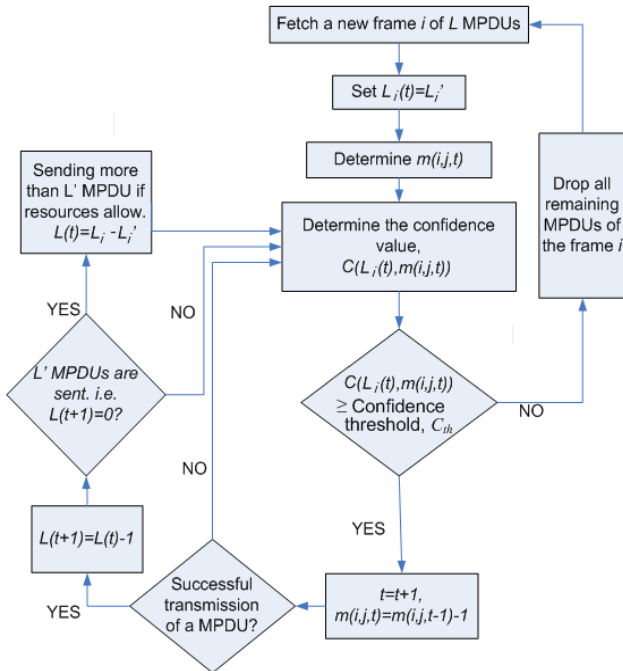


Fig. 3. The Active dropping mechanism.

B. Analysis of Confidence

An analytical framework based on an embedded Markov-chain model is developed to quantify the confidence of successfully delivering a frame within the given application-layer delay bound. Specifically, the Markov-chain evaluates

the probability of successfully sending $L_i(t)$ MPDUs of frame i in the queue within $m(i, t, j)$ available transmission timeslots. For simplicity, a two-state wireless channel model is employed and described below. On the other hand, a more complicated channel model could be also adopted in the proposed analytical framework at the expense of higher computation complexity.

Wireless Channel Model: A common two-state Gilbert–Elliot model [17] is employed as the underlying wireless channel model, where the wireless link is modeled as a discrete time Markov Chain with “Good” and “Bad” states. The error probability of a MPDU transmission is 0 and 1 when the channel is “Good” and “Bad”, respectively. The probabilities P_{GB} and P_{GG} represent the transition probabilities from “Good” state to “Bad” and “Good” states, respectively. Similarly, P_{BB} and P_{BG} represent the transition probabilities from “Bad” state to “Bad” and “Good” states, respectively. P_{BG} , P_{GB} , P_{BB} and P_{GG} are derived by the average MPDU-error-rate (PER) ε and error burst length (EBL) here:

$$P_{BG} = \frac{1}{EBL}, \quad P_{GB} = \frac{\varepsilon}{EBL(1-\varepsilon)}, \quad (6)$$

$$P_{BB} = 1 - P_{BG}, \quad P_{GG} = 1 - P_{GB}$$

Calculation of Confidence: Based on the wireless channel model, the delivery of each MPDU in a frame along with the corresponding confidence is modeled through a discrete time Markov-chain as shown in Fig. 4. States G_1 and B_1 represent that the 1st MPDU of the frame to be transmitted when the channel is in “Good” state and “Bad” state, respectively. Similarly, states G_2 and B_2 represent the 2nd MPDU of the frame to be transmitted when the channel is in “Good” and “Bad” states, respectively. If the system is currently in the state “ G_i ” or “ B_i ”, it means that the system has successfully sent $i - 1$ MPDUs already and going to send the i th MPDU while it is at the “Good” or “Bad” state respectively. This model captures the transmission of every successfully sent MPDU of a frame until the states G_L and B_L , where L is the number of MPDUs of a particular frame. An unsuccessful transmission of a MPDU could (optionally) initiate a retransmission, where the maximum number of retransmission attempts is defined by the retransmission policy. Our analytical model on the probabilistic confidence, nonetheless, is independent of the retransmission limit, and allows our scheme to incorporate with any retransmission policy in a cross-layer design approach.

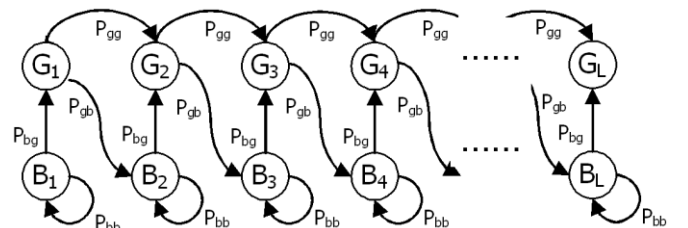


Fig. 4. Markov model for successfully transmitted MPDUs of a frame.

Transitions between “G_i” states and “B_i” states are described by the two-state wireless channel model through a one-step transition probability matrix, $Tr(L')$ with the size of

$$Tr(L)|_{L'=1} = \begin{bmatrix} 0 & 0 \\ P_{BG} & P_{BB} \end{bmatrix}, \quad Tr(L')|_{L'=2} = \begin{bmatrix} 0 & 0 & P_{GG} & P_{GB} \\ P_{BG} & P_{BB} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & P_{BG} & P_{BB} \end{bmatrix}, \quad \dots, \quad (7)$$

$$Tr(L') = \begin{bmatrix} 0 & 0 & P_{GG} & P_{GB} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ P_{BG} & P_{BB} & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{GG} & P_{GB} & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{BG} & P_{BB} & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{BG} & P_{BB} & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & P_{GG} & P_{GB} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & P_{BG} & P_{BB} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & P_{GG} & P_{GB} \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & P_{BG} & P_{BB} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & P_{BG} & P_{BB} \end{bmatrix}$$

Assuming that $m(i, t, j)$ transmission timeslots will be completely consumed for sending frame i at time t in the j th scheduling cycle, if the BS still has any number of remaining MPDUs of the frame in the queue at the end, the system would definitely fall into one of the states in Fig. 4. This indicates that those remaining MPDUs in the queue should be proactively dropped in the BS back then during the moment at time t instead of consuming any single timeslot. In other words, the interested metric is *the probability that the system will not stay in any of these states after $m(i, t, j)$ transmission timeslots*. Therefore, the confidence of the BS for sending $L_i(t)$ MPDUs successfully within $m(i, t, j)$ timeslots at time t in the j th scheduling cycle can be defined as

$$C(L_i(t), m(i, t, j)) = 1 - \Pi_0 Tr(L_i(t))^{m(i, t, j)} e \quad (8)$$

where $L_i(t)$ is the number of MPDUs of frame i to be transmitted at time t , $m(i, t, j)$ is the available transmission timeslots for sending the remaining MPDUs of frame i in the queue, Π_0 is the initial state probability vector, $Tr(L_i(t))$ is the one step transition probability matrix with the size of $2L_i(t) \times 2L_i(t)$, and $e = [1 \dots 1]^T$ is a column vector with a size of $1 \times 2L_i(t)$. Fig. 5 shows the relationship between the confidence value, and a difference between $m(i, t, j)$ and $L_i(t)$ which is defined as $Diff_mL = m(i, t, j) - L_i(t)$. The value of $Diff_mL$ can be interpreted as the extra transmission timeslots available for retransmitting a MPDU of a frame due to a MPDU loss event. Note that the confidence becomes smaller for a frame with a larger L even the two video frames have the same value of $Diff_mL$. This is because with the more MPDUs in a frame, there is a higher chance of unsuccessful transmission of MPDUs, which expects additional transmission opportunities for retransmission. On the other

hand, every successful MPDU transmission leaves the same extra $Diff_mL$ timeslots for the next MPDU and decreases the remaining number of MPDUs, $L_i(t)$, in the queue, the system will be therefore escalated to follow a tendency of a higher confidence value as shown in Fig. 5. This gracefully balances between the tradeoff of video quality and the aggressiveness of proactive dropping decisions. Such relationship also provides us an insight about the impact of confidence thresholds on different frame types, since I-, P- and B-frame typically generate different amounts of MPDUs. With the same extra resources $Diff_mL$, we may like to assign an I-frame with a smaller confidence threshold than that for a P- or B-frame because (i) a successful MPDU transmission belonging to an I-frame is much favored to the video quality than that from a B- or P-frame; and (ii) an I-frame usually has a much larger value of L . In other words, the dropping mechanism on different video frames can be differentiated by using different confidence thresholds, where I-frames is assigned a smaller confidence threshold in order to factor its importance to the video playback quality.

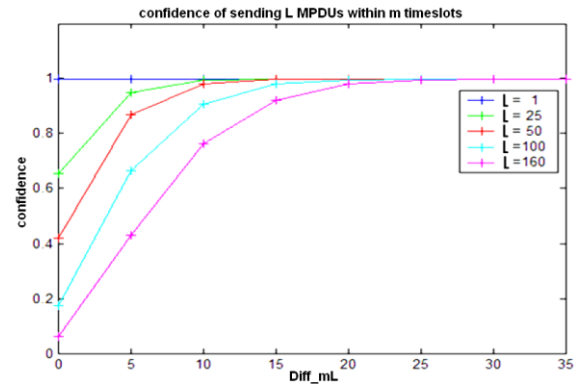


Fig. 5. The relationship between the confidence value and the difference of $m(i, t, j) - L(t)$.

A. Implementation Issues

One of our major design goals is the simplicity and practicality to integrate the proposed scheme into a real world system. This subsection provides detailed discussions on the complexity and implementability of the proposed scheme. As illustrated in Fig. 3, only $m(i, t, j)$, and $C(L_i(t), m(i, t, j))$ are required to evaluate a dropping decision. Determining $m(i, t, j)$ is relatively straightforward since the required involved in Eq. (4) and Eq. (5) are either constants such as D , Q , Δ , and T_{in} , whereas $T_{leftover}(i, t, j)$ and $T_x(i, t, j)$ are directly tracked by the system after each MPDU transmission. To determine the value of $C(L_i(t), m(i, t, j))$, on the other hand, requires the calculation of the one-step transition probability, $Tr(L_i(t))$ from Eq. (7) and Eq. (8). The complexity of the state space of $Tr(L_i(t))$ is $O((2L)^2)$ and is bounded by the maximum video frame size. The current value of $L_i(t)$ reflects the number of remaining MPDUs of frame i awaiting in the queue to be transmitted, which is updated along the time after each successful transmission. The following is an illustration of the complexity of the proposed scheme. We check the dropping criteria for every MPDU. For an I-frame with a size of 8400 bytes plus the overhead of RTP/IP headers and a MPDU size of 52 bytes in the BS, about 160 MPDUs on average are generated to form a 320-by-320 matrix of $Tr(L_i(t))$ initially. After the first successful MPDU transmission, $Tr(L_i(t))$ is reduced to a 318-by-318 matrix, which is in turn used to evaluate the second MPDU transmission of the frame, and so on. With a single 2 GHz processor, calculating 320×320 elements (i.e., around 102,400 operations) in $Tr(L_i(t))$ takes about 102.4 ms by software to derive the confidence value $C(L_i(t), m(i, t, j))$, assuming that each arithmetic operation involves two instructions and each instruction consumes a CPU cycle. In order to optimize the computational time required in Eqs. (7) and Eq. (8), a hardware-based look-up-table approach can be developed so that all the possible confidence values are tabulated and stored in a hardware chip such as a field programmable gate array (FPGA). The size of the table could be as small as 160 (different sizes of L) by 35 (different values of $Diff_mL$), where $L_i(t)$ and $m(i, t, j)$ are taken as the parameters for the look-up-table process. The retrieved confidence value is then compared with a given confidence threshold for making a dropping decision. The look-up-table process can be constantly fast because of the small and finite searching space (i.e., 160×35). Note that the above table is for a specific pair of EBL and ε . Such table can be prepared and updated dynamically in the FPGA chip by the software approach for a particular wireless channel condition characterized by EBL and ε . In this case, an evaluation of dropping criteria can go through a proper table specific to the real geographical environment and channel conditions according to EBL and ε as shown in Fig. 6, where the computation overhead is kept as the minimal. A significant

number of such look-up-tables are generated for properly covering the range of the channel conditions in a deployment, which can be justified for the cost of a BS and the hardware technologies available today. In case, a look-up-table process does not succeed in matching the actual channel condition, interpolations is performed for making an immediate dropping decision. A new look-up-table for a new channel condition will be computed and inserted into the system.

L	1	..	50	..	160
1	1	..	0.45	..	0.1
..
1	1	..	1	..	0.9
..
3	1	..	1	..	1
..
35	1	..	1	..	1

Fig. 6. A hardware approach of look-up table for AD mechanism.

IV. SIMULATION RESULTS

The proposed AD mechanism is evaluated using Matlab with the video traces containing a various amount of frames (200, 400, 800, 1,600, 3,200 and 6,400). The video traces have a popular MPEG-4 Group-of-Pictures (GoP) pattern I-B-B-P-B-B-P-B-B-P-B-B. Each I-, P-, and B-frame is consisted of 4, 2, and 1 packets, respectively and, each packet is 2,100 bytes, such that all traces are simulating the variable bit rates. A MPDU has 52 bytes in size, and can be sent within a timeslot. In order to evaluate the efficiency of our proposed scheme fairly, we compare it with a counterpart in which the scheduling and buffer management policy in the BS will transmit L' most important MPDUs of a frame first, and then transmit more MPDUs if there is any time resources available within the corresponding delay bound. This abstracts many common video transmission techniques on wireless channels using prioritization-based cross-layer design approach previously reported, in which transmission/retransmission opportunities are normally prioritized for those MPDUs of the frames based on its importance to the minimal acceptable video quality through a utility function defined in an optimization process. We refer this counterpart as the “normal” prioritization-based cross-layer scheme. The recipient-side playback buffer is set with $\Delta = 1$. The retransmission limit is infinite for simplicity. Note that having the assumption of infinite retransmission does not lose the generality as discussed in Section III-A. The wireless channel is evaluated under a typical good condition ($EBL = 2$, $\varepsilon = 0.01$) and a typical bad condition ($EBL = 4$, $\varepsilon = 0.1$).

A. Rate of Frame Loss and Resources released

The frame loss rate is an important index for the video quality at the recipient. In our simulations, the confidence thresholds of different types of frames are not differentiated initially and set as 0.75 for all of them. The maximum losable

fraction, $x\%$, of a video frame is set as 0.2 (i.e., $x\% = 20\%$). The simulation results are shown in Fig. 7. Under the bad channel condition ($EBL = 4$, $\epsilon = 0.1$), the proposed AD scheme causes much less frame loss than that under the “normal” scheme even the number of frames in a video trace increases. On the contrary, the “normal” scheme suffers from a fast increasing frame loss rate when the number of frames in video traces increases due to the delay propagations to more subsequent frames from the preceding frames. With a good channel condition ($EBL = 2$, $\epsilon = 0.01$), the frame losses in both schemes are less severe due to the less fluctuating channel capacity and rate of errors, whereas our scheme still achieves a relatively lower frame loss than the “normal” scheme for any number of frames in the traces over 3200 frames. When the frame number in the video trace is getting smaller than 3200 frames, both schemes present the comparable frame loss rate. However, the proposed AD scheme yields the benefit of releasing resources for the other competing flows or subsequent frames by actively dropping MPDUs of any late arrival frame. Fig. 8 shows a detailed breakdown of the frame losses in terms of the distribution among the type of frames (I-/P-/B-) under the bad channel condition ($EBL = 4$, $\epsilon = 0.1$). The AD scheme incurs less frame loss in all frame types regardless of the number of frames in the video traces. It is because that our scheme never allows the delay of any frame to be propagated to others, and all associated MPDUs of a possibly late arrival frame are dropped immediately before consuming any transmission opportunity. Notice that the “normal” scheme here reflects the major shortcomings of many cross-layer approaches based on the prioritization-based optimization, where the frame losses caused by the delay propagations become worst when the number of frames of the video source increases. In this case, the resulting video quality is severely degrading across the streaming process beside the precious transmission resources are wasted in sending those late frames. In summary, the major benefit of adopting AD is the release of resources by dropping those MPDUs of a frame which is not likely to arrive at the recipient within its application-layer delay bound. Fig. 9 demonstrates the resources released by AD with different number of frames in the video traces. In general, more resources are released under a bad channel condition due to a larger chance of having the late arrival frames. The released resources can be allocated to the subsequent MPDUs of the frame or a competing video stream.

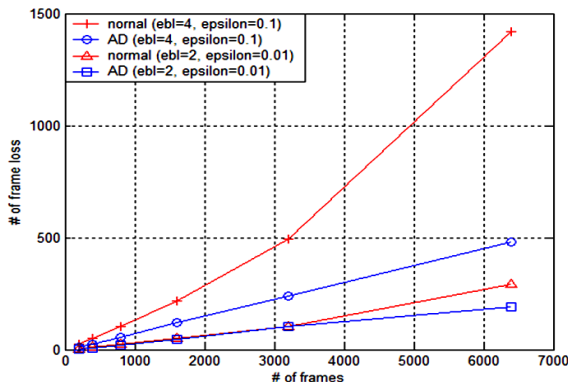


Fig. 7. Number of frame loss vs. number of frames in the traces.

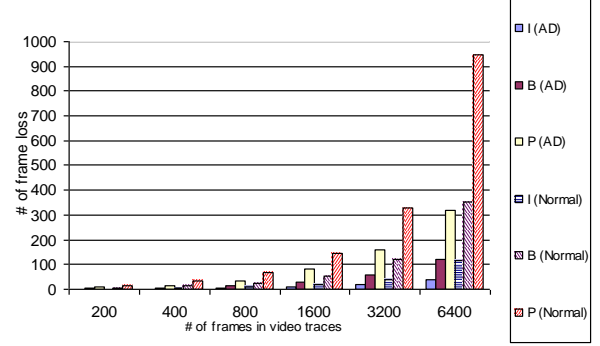


Fig. 8. Distribution of frame loss among I, P and B types vs. number of frames in video trace under a bad channel ($EBL = 4$, $\epsilon = 0.1$)

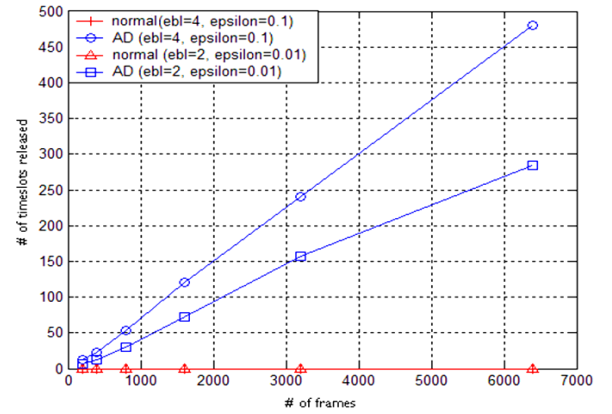


Fig. 9. Resource released vs. number of frames in the traces.

B. Impact of Maximum Losable Fraction, $x\%$

The impact of maximum losable fraction, $x\%$, in a frame towards the total number of frame loss is evaluated under a good channel condition in Fig. 10. With a smaller $x\%$ value, a higher frame loss rate is observed, where L' becomes larger such that more MPDUs are required to be received within the same amount of available transmission timeslots within the delay bound. The confidence of successfully sending required MPDUs of the frame will become smaller given the same amount extra resources ($Diff_mL$) as shown in Sub-Section III-B. Understanding the impact of $x\%$ in a video frame is critical for achieving a graceful tradeoff between the rate of frame loss and the constraints from the adopted video coding technique or the application-level QoS requirement in terms of video quality. Incorporating with the AD mechanism, a suite of adaptive and application-defined strategies can be developed to differentiate and compliment those reported cross-layer video streaming approach by exploiting the properties of scalable video coding with the parameters (i.e., $x\%$ and confidence threshold).

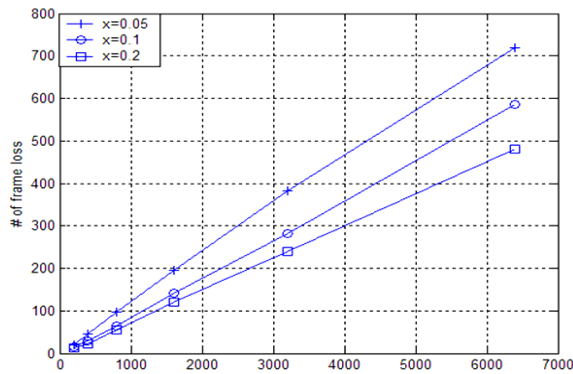


Fig. 10. Impact of losable fraction of data in a frame to frame loss vs. number of frames in the traces.

C. Differentiated Confidence Thresholds

The video trace with 6,400 frames is evaluated with 2 different confidence thresholds – $C_{th} = 0.75$ and $C_{th} = 0.95$. From Fig. 11, the higher confidence threshold ($C_{th} = 0.95$) causes more frame loss in all frame types as shown in the first two sets of bar charts. A different value of confidence threshold is applied to each individual frame type, such as $DC_{th} = 0.3$ for I-frame, $DC_{th} = 0.75$ for P-frame, and $DC_{th} = 0.95$ for B-frame, in order to reduce the loss of I-frames. The results are shown in the last set of bar charts of Fig. 11.

When the confidence threshold of the simulation is set as 0.75 in Section V-A, the AD scheme incurs the smaller frame loss rate than that of the “normal” cross-layer scheme in general, as shown in Fig. 7. This results a higher rate of decodable frames by the AD scheme and gives a better video quality according to Eq. (2) in the presence of the same transmission resources, as shown in Fig. 12(a). Due to the intrinsic ability of bounding delay propagation and resources released from the late video frames, the AD scheme tends to maintain a more graceful video quality degrade when the number of frames in the video source increases (so as the frame loss rate indicated in Fig. 7).

In summary, the confidence threshold applied for a recipient that determines the decodable frame rate of the video stream, and should be a parameter subject to dynamic configuration through a closed-loop control. If the video quality (Q) requirement has been already met at the recipient side, the confidence threshold at the BS can be increased to release more resources for the other competing flows when the inter-service time is large (which means the network resources are not sufficient). The adaptive adjustment of the confidence threshold can be directly tuned through the current video quality (Q) evaluated by Eq. (2) to achieve the optimal resource releases and allocation without losing the minimal expected video quality at the recipient-side.

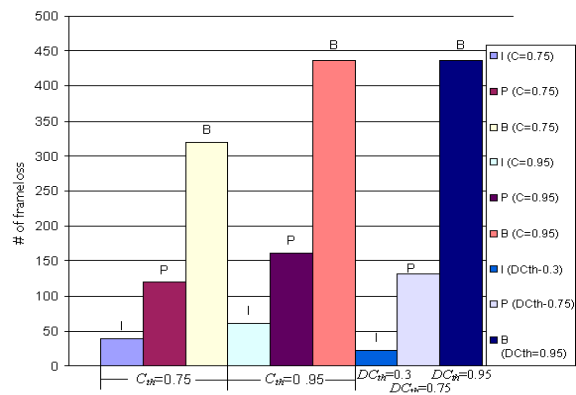
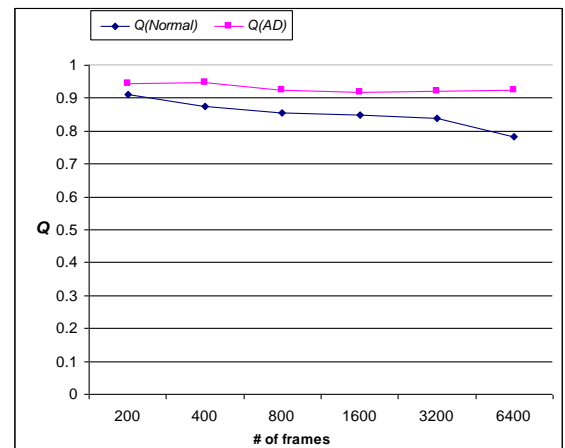
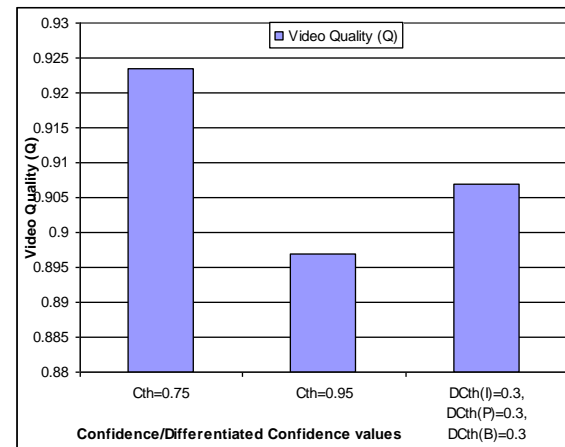


Fig. 11. Impacts of confidence thresholds to the frame loss.



(a)



(b)

Fig. 12. The relations between video quality and confidence thresholds

V. CONCLUSION

In this paper, an Active Dropping (AD) scheme has been introduced for achieving effective resource utilization and maintaining application-level QoS for real-time streaming video over emerging wireless broadband network using TDMA such as WiMAX. The proposed scheme is particularly effective when a stringent delay bound is required on each video frame, and the wireless channel condition is subject to serious fluctuations. A comprehensive analytical model has been formulated to quantify the probabilistic confidence of successfully sending a video frame within its application-layer

delay bound by jointly considering the effect of the wireless channel condition, MPDU retransmissions, playback buffer, and decodability requirement of the video coding. The scheme is simple yet generically interoperable with any MAC layer retransmission policy, and can be effectively implemented in a practical system through software- or/and hardware-based approaches. We have examined the effectiveness of the scheme by way of extensive simulations in the aspect of frame loss rate and resources released when compared to a generic cross-layer optimization based on prioritization. The impacts of maximum losable fraction in a video frame and the confidence threshold toward the frame losses are also evaluated. A video quality measure is also adopted to evaluate the scheme for the perceivable quality performance. We believe that the proposed scheme and the analytical framework can be also applicable to many future scalable video coding to achieve efficient wireless real-time video streaming for emerging multimedia services in IEEE 802.16 based BWA networks.

APPENDIX

N_{dec} is the expected total number of decodable frames in each type of frame. i.e., $N_{dec} = N_{dec-I} + N_{dec-P} + N_{dec-B}$, which is defined as the following according to [13].

A.1 The expected number of decodable I-frames (N_{dec-I})

In a GOP, an I-frame is decodable only if all the packets that belong to the I-frame are correctly received. Therefore, the probability that the I-frame is decodable is $(1-p)^{C_I}$, where p is the packet loss rate, and C_I is number of packets belonging to the I-frame. Consequently, the expected number of correctly decodable I-frames for the whole video is

$$N_{dec-I} = (1-p)^{C_I} \times N_{GOP}$$

where N_{GOP} is the total number of GoPs in the video.

A.2 The expected number of decodable P frames (N_{dec-P})

In a GOP, a P-frame is decodable only if the preceding I- or P-frames is decodable and all the packets that belong to the P-frame are decodable. The expected number of correctly decodable P-frames for the whole video is

$$\begin{aligned} N_{dec-P} &= (1-p)^{C_I} \times \sum_{j=1}^{N_P} (1-p)^{j \times C_P} \times N_{GOP} \\ &= N_{dec-I} \times \sum_{j=1}^{N_P} (1-p)^{j \times C_P}, \end{aligned}$$

where N_P is the number of P-frames in a GoP, and C_P is the number of packets in the P-frame.

A.3 The expected number of decodable B frames (N_{dec-B})

In a GOP, a B-frame is decodable only if the preceding and succeeding I- or P-frame are both decodable, and all the packets that belong to the B-frame are decodable. As consecutive B-frames have the same dependency throughout the GOP structure, we consider the consecutive B-frames as a

B-group. Especially, since the last B-frame in a GOP is encoded from the preceding P-frame and succeeding I-frame, the B-frame may be affected by the correctness of the two I-frames. Thus, the expected number of correctly decodable B-frames for the whole video is

$$\begin{aligned} N_{dec-B} &= \left[(1-p)^{C_I + N_P C_P} + \sum_{j=1}^{N_P} (1-p)^{j C_P} \times (1-p)^{C_B} \right] \\ &\quad \times (M-1) \times (1-p)^{C_I + C_B} \times N_{GOP} \\ &= N_{dec-I} \times \left[(1-p)^{C_I + N_P C_P} + \sum_{j=1}^{N_P} (1-p)^{j C_P} \times (1-p)^{C_B} \right] \\ &\quad \times (M-1) \times (1-p)^{C_B} \end{aligned}$$

where C_B is the number of packets in the B-frame, M is the distance between the I- and P-frame in term of the number of frames.

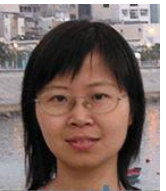
REFERENCES

- [1] IEEE Standard 802.16e-2005, "IEEE Standard for Local and Metropolitan Area Networks - Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems", Dec. 7, 2005.
- [2] J. Gross, J. Klaue, H. Karl and A. Wolisz, "Cross-layer Optimization of OFDM Transmission Systems for MPEG-4 Video Streaming," *Comp. Commun.*, vol. 27, no. 11, pp. 1044-55, Jul. 1, 2004.
- [3] R. S. Tupelly, J. Zhang, and E. K. P. Chong, "Opportunistic Scheduling for Streaming Video in Wireless Networks," in *Proc. of the 37th Annual Conference on Information Sciences and Systems*, Baltimore, Maryland, Mar. 12-14, 2003.
- [4] Y. Shan and A. Zakhor, "Cross Layer Techniques for Adaptive Video Streaming over Wireless Networks," *IEEE Int'l. Conf. Multimedia and Expo*, Lausanne, Switzerland, Aug. 26-29, 2002.
- [5] P. Buccioli, G. Davini, E. Masala, E. Filippi and J. C. De Martin, "Cross-layer Perceptual ARQ for H.264 Video Streaming over 802.11 Wireless Networks," in *Proc. of IEEE GLOBECOM*, vol. 5, pp. 3027-3031, Nov.-Dec. 2004.
- [6] J. Song and K. J. R. Liu, "An Integrated Source and Channel Rate Allocation Scheme for Robust Video Coding and Transmission over Wireless Channels," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 2, pp. 304-316, 2004.
- [7] 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 JSAC*, vol. 21, no. 10, pp. 1685-1698, Dec. 2003.
- [8] C. Krasic, J. Walpole and W.-C. Feng, "Quality-adaptive media streaming by priority drop", in *Proc. of the 13th Int'l. Workshop on Network and Operating Systems Support for Digital Audio and Video*, pp. 112-121, Jun. 2003.
- [9] Y. Huang, R. Guérin and P. Gupta, "Supporting Excess Real-Time Traffic With Active Drop Queue", *IEEE/ACM Tran. on Networking*, vol. 14, no. 5, pp.965-977, Oct. 2006
- [10] J. Huang, C. Krasic, J. Walpole, and W.-C. Feng, "Adaptive Live Video Streaming by Priority Drop", *IEEE Int'l. Conf. on Advanced Video and Signal Based Surveillance*, pp. 342, Jul. 2003.
- [11] J. She, F. Hou, and P.-H. Ho, "An Application-Driven MAC-layer Buffer Management with Active Dropping for Real-time Video Streaming in 802.16 Networks," 21st International Conference on Advanced Information Networking and Applications, 2007. AINA '07, vol., no., pp.451-458, 21-23 May 2007.
- [12] J. Mitchell and W. Pennebaker, "MPEG Video: Compression Standard", *Chapman and Hall*, 1996. ISBN 0412087715.
- [13] C.-H. Lin, C.-H. Ke, C.-K. Shieh and N. K. Chilamkurti, "The Packet Loss Effect on MPEG Video Transmission in Wireless Networks", in *Proc. of the 20th International Conference on Advanced Information Networking and Applications (AINA'06)*, vol. 1, pp. 565 - 572, 2006.

- [14] B. E. Wolfinger, "On The Potential of FEC Algorithms in Building Fault-tolerant Distributed Applications to Support High QoS Video Communications," in *Proc. of the ACM Symp. on Principles of Distributed Computing (PODC'97)*, Santa Barbara, CA, USA, 1997.
- [15] A. Ziviani, B. E. Wolfinger, J. F. Rezendé, O. C. M. B. Duarte, and S. Fdida, "Joint Adoption of QoS Schemes for MPEG Streams," *Multimedia Tools and Applications Journal*, Springer Netherlands, vol. 26, no.1, pp. 59-80(22), May 2005.
- [16] M. Kalman, S. T. Eckehard, and B. Girod, "Adaptive Playback for Real-time Media Streaming," *IEEE Int'l. Symposium on Circuits and Systems, 2002 (ISCAS 2002)*, vol. 1, pp. I-45 - I-48, May 26-29, 2002.
- [17] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error Statistics in Data Transmission over Fading Channels," *IEEE Trans. Comm.*, vol. 46, pp. 1468-1477, Nov. 1998.



James She is now a Research Fellow in Computer Laboratory at the University of Cambridge, Cambridge, United Kingdom. His research interests include wireless multimedia communications, social networks, content delivery networks, and wireless sensor devices/networks. He received the B.Eng. degree in Computer Engineering and the M.Phil. degree in Electrical and Computer Engineering from the Hong Kong University of Science and Technology. He received his Ph.D. degree in Electrical & Computer Engineering from University of Waterloo, Ontario, Canada in 2009, with an honor of "Outstanding Achievement of Graduate Studies". He is also the recipient of two national awards in 2009 – NSERC Postdoctoral Fellowship Award and NSERC Innovation Challenge Award. His Ph.D. thesis also secured cumulatively \$160,000 of prototype development and commercialization funding from various provincial and national funding agencies. During his Ph.D. study, he has generated 5 International, US, Taiwanese patents and authored/co-authored 20 publications. His masters' thesis on Internet-based video broadcasting generated U.S. and Chinese patents. From 2000–2002, James served as Chief Technology Officer for SinoCDN Limited, a streaming video delivery solution provider. In 2001, James received two Hong Kong Industry Awards for his patented technology. He was the youngest recipient, to-date, to be named one of "Ten Outstanding Young Persons" by the Innovation and Technology Association and the Information Technology Department of the Government of Hong Kong.



The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph. The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.



The second paragraph uses the pronoun of the person (he or she) and not the author's last name. It lists military and work experience, including summer and fellowship jobs. Job titles are capitalized. The current job must have a location; previous positions may be listed without one. Information concerning previous publications may be included. Try not to list more than three books or published articles. The format for listing publishers of a book within the biography is: title of book (city, state: publisher name, year) similar to a reference. Current and previous research interests end the paragraph.

The third paragraph begins with the author's title and last name (e.g., Dr. Smith, Prof. Jones, Mr. Kajor, Ms. Hunter). List any memberships in

professional societies other than the IEEE. Finally, list any awards and work for IEEE committees and publications. If a photograph is provided, the biography will be indented around it. The photograph is placed at the top left of the biography. Personal hobbies will be deleted from the biography.



Pin-Han Ho (M'02) received his B.Sc. and M.Sc. degrees from the Electrical Engineering department, National Taiwan University, in 1993, 1995, and 2002, respectively. He received Ph.D. degree from Queen's University at Kingston in 2002. Professor Ho's current research interests cover a wide range of topics in broadband wired and wireless communication networks, including survivable network design, wired-wireless metropolitan-area and access networks, and network security. He is the recipient of the Distinguished Research Excellence Award in the ECE department of the University of Waterloo, the Early Researcher Award in 2005, the Best Paper Award in SPECTS'02, ICC'05 Optical Networking Symposium, and ICC'07 Security and Wireless Communications symposium, and the the Outstanding Paper Award in HPSR'02.