MAC-Layer Active Dropping for Real-Time Video Streaming in 4G Access Networks

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

***Abstract*—This paper introduces a MAC-layer active dropping scheme to achieve effective resource utilization, which can satisfy the application-layer delay for real-time video streaming in time division multiple access based 4G broadband wireless access networks. When a video frame is not likely to be reconstructed within the application-layer delay bound at a receiver for the minimum decoding requirement, the MAC-layer protocol data units of such video frame will be proactively dropped before the transmission. An analytical model is developed to evaluate how confident a video frame can be delivered within its application-layer delay bound by jointly considering the effects of time-varying wireless channel, minimum decoding requirement of each video frame, data retransmission, and playback buffer. Extensive simulations with video traces are conducted to prove the effectiveness of the proposed scheme. When compared to conventional cross-layer schemes using prioritized-transmission/retransmission, the proposed scheme is practically implementable for more effective resource utilization, avoiding delay propagation, and achieving better video qualities under certain conditions.**

***Index Terms*—Application-layer delay, cross-layer design, dropping, IEEE 802.16, LTE, real-time video streaming, WiMAX.**

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 broadband wireless technologies such as IEEE 802.16 (also known as WiMAX) and TD-LTE (a variant of long-term evolution) to make broadband wireless access (BWA) services possible [1]. Both IEEE 802.16 and LTE technologies are envisioned to support high capacity broadband performance in wireless along with strong yet simple QoS supports using time-division multiple access (TDMA). With the early deployments, trials and researches around the world by both industries and academia, WiMAX and LTE are recognized as the promising and/or complementary access network technologies when compared to legacy cable wirelines, digital subscriber lines (DSL) and 802.11a/b/g (or Wi-Fi) for all-IP broadband wireless access services in the metropolitan area networks. Although delivering bandwidth-intensive data like video over the time-varying wireless channel is always challenging, such BWA platform can foster many existing IP-based streaming video applications to take advantages of wireless and/or mobility. Innovative and exciting next-generation wireless video-based services are therefore possibly cultivated to utilize the true potentials of BWA networks, such as wireless/mobile IPTV, wireless digital signage, mobile advertisement, etc.

Real-time video streaming is a widely adopted yet highly demanding component in the aforementioned 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 a streaming software at 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 (MPDUs) of consecutive video frames in a real-time video stream could be lost or delayed seriously. This is caused by the time-varying capacity and bit error rate of a wireless channel subject to the fading and mobility effects. Specific traffic classes of services (e.g., UGS and rtPS in WiMAX standards) for dedicated access control and radio resources allocation are defined in many BWA technologies. However, 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. This could eventually prolong disruptions on the perceived video quality over more than just one video frame and possibly all real-time video services in the system. To address 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] proposes 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 subcarriers to the user by jointly prioritizing with the size of the queues. In [3], an opportunistic scheduling algorithm for multiple video streams using a priority function is developed based on channel conditions, importance of frames, queue size, and

Manuscript received March 01, 2009; revised November 10, 2009.

J. She is with the Computer Laboratory, University of Cambridge, Cambridge, U.K.

F. Hou is with the Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong, China.

B. Shihada is with Mathematical and Computer Science and Engineering Division, King Abdullah University of Science and Technology (KAUST), Saudi Arabia (e-mail: basem.shihada@kaust.edu.sa).

P.-H. Ho is with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada.

Digital Object Identifier 10.1109/JSYST.2010.2086870

multiplexing gain. A packetization scheme is 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 is 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 data with efficient packetization and error concealment techniques altogether subject to the channel. It is 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-based transmission system [7]. However, the aforementioned cross-layer schemes 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 scheme 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 scheme. Some previous attempts of dropping packets for satisfying 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 to on-demand video streaming content. 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 are 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 maintaining 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 is 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 ms 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 this 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, this scheme is only designed to work with a fixed transmission window at the sender in a wired network without considering the dynamics of the playback-buffer at recipient software and/or any retransmitting strategy in place.

Based on our knowledge so far, it is the first attempt of a dropping approach that comprehensively and systematically addresses the aggregated dynamics on the application-layer delay towards the real-time video quality by jointly considering the impacts from: i) the wireless channel fluctuation; ii) the recipient-side playback-buffer, which is known to be critical to any video streaming system; iii) the minimum decoding requirement; and, iv) the retransmission of lost data. Note that the application-layer delay bound and the minimum decoding requirement (i.e., the maximum loss tolerance) of a frame are considered as two simple parameters that easily available from any adopted video coding and streaming software. Either/both of these parameters serve important factor(s) in designing any scalable video coding technique and effective cross-layer transmission strategies. A preliminary cross-layer dropping scheme is introduced in one of our previous work [11] that successfully avoid delay propagation. Promising results have showed the effectiveness of such initial design. In addition to the discussions about the practical implementation, we are interested to extend the work into a more comprehensive framework along with a generic analytical measure of achievable video quality for system evaluation based on the number of decodable frames and other system parameters discussed above. In this paper, we propose a MAC-layer active dropping (AD) scheme for real-time video streaming in BWA networks, which satisfies the stringent application-layer delay bound of each video frame. For the simplicity, the discussion is based on the WiMAX standards because of the maturity and availability of existing hardware/systems. However, the proposed scheme can also work in any emerging broadband wireless access technology that supports TDMA. The proposed AD scheme has the following contributions: 1) it guarantees an application-level delay bound on each video frame, while releasing transmission resources that were originally wasted for late video frames; 2) it 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) it preserves the MDPUs of a video frame delivered without changing their orders for seamless system integrations. To achieve such design goals above, a novel probabilistic model is developed and employed with the proposed scheme to determine the confidence of a video frame to satisfy its instantaneous application-layer delay bound. Based on such probabilistic evaluation, the proposed scheme proactively drops awaiting MPDUs of a video frame that is considered not confident enough to arrive at the recipient by its application-layer delay bound under the minimum decoding requirement. Instead of wasting the system resources, the precious transmission resources can be released for subsequent frames or other competing real-time video streams. The revenue potentials and system scalability are therefore maximized without losing the equivalent application-layer QoS conditions, such as the resultant rate of frame loss. Extensive simulations have demonstrated the effectiveness of the proposed

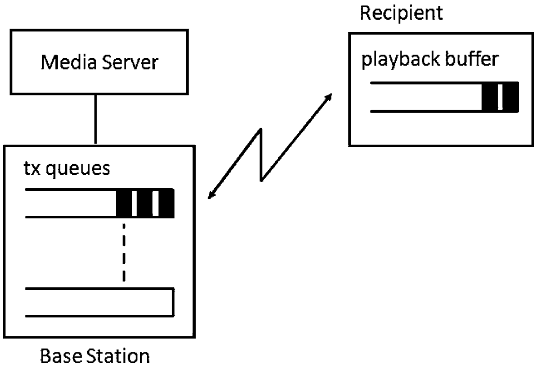


Fig. 1. System model.

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. The details of proposed AD scheme with a comprehensive analytical framework as well as the implementability are discussed in Section III. The effectiveness of the proposed scheme is evaluated with simulation results in Section IV, ended by the conclusion.

II. System Model

A. System Architecture

As discussed earlier, the proposed scheme is applicable to any emerging BWA technology supporting TDMA. Without loss of generality, a generic system architecture here is presented based on WiMAX standards (PMP mode), in which transmission resources, in terms of slotted times, are shared by way of time division multiple access (TDMA). All subcarriers can be possibly allocated in a timeslot. The traffic class of UGS in WiMAX is associated for a stream of video data, where a dedicated number of transmission timeslots may be assigned in each physical downlink subframe. Fig. 1 illustrates the system architecture, which includes: 1) a media server; 2) the 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.

On the other hand, there are some basic characteristics defined in this system architecture: 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 to be sent 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 that defines the maximum application-layer delay bound of a 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 at the BS for a lost MPDU, whereas, a retransmission strategy is optional for real-time video streaming but is included in the proposed scheme for the sake of completeness.

B. Recipient-Side Playback Buffer

Without loss of generality, the frame playback rate at the recipient is assumed to be the same as the frame-producing rate at the media server (i.e., every P timeslots per frame). A built-in playback buffer at the recipient's streaming software serves as a reservoir for mitigating the vicious impact due to fluctuations of the communication latency [16]. Hence, the head of line (HoL) frame will be started to playback, only if a number, Δ, of frames is already accumulated in such buffer. The value of Δ is usually designed small for real-time video applications, for example, 1–5 frame(s), which in turn defines a maximum application-layer delay bound, denoted as, D, of a video frame that the recipient's streaming software can tolerate to wait for the data of a video frame before its playback. In other words, a newly generated video frame from the media server could has an extended budget of time duration, D, to be transmitted from the BS to the recipient (note that the latency between a media server to the BS is assumed to be negligible in our discussion). In the case of no playback buffer (i.e., $\Delta = 0$, D will be equal to limited duration of P, which means a frame can only spend a duration no longer than P to reach the recipient for its playback for avoiding a frame loss event. In summary, the maximum application-layer delay bound of a video frame, D, can be formulated as

$$D = (\Delta + 1) \times P \tag{1}$$

where Δ is the number of frames 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 MPEG standards [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 then 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 timeslots inside a downlink subframe. The lossy wireless channel can introduce unpredictable bandwidth and loss of MPDUs due to the fluctuating channel capacity and bit error rates, which will impair the arrival time of a video frame and its video quality perceived by the recipient. The impairment on the perceptual video quality becomes more serious when the channel condition gets worse, and/or the application-layer delay propagates across consecutive video frames when the system transmission resources are heavily demanded. To practically and objectively evaluate the video quality, a reasonable evaluation model of video quality (Q) from [13] is adopted, which is defined by the number of decodable frames over the total number of frames originally from the video source

$$Q = \frac{N_{\text{dec}}}{N_{\text{total}-I} + N_{\text{total}-P} + N_{\text{total}-B}} \tag{2}$$

where $0 < Q < 1, N_{\text{dec}}$ is the expected total number of decodable frames including all the types of frames (i.e., $N_{\text{dec}} = N_{\text{dec}-I} + N_{\text{dec}-P} + N_{\text{dec}-B}$). Note 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., $N_{\text{dec}-I}, N_{\text{dec}-P}$, and $N_{\text{dec}-B}$), which is given in the Appendix. $N_{\text{total}-I}, N_{\text{total}-P}$ and $N_{\text{total}-B}$ are the total number of I-, P-, B-frames in the video source, respectively. Q is designed as an objective measure to evaluate the video quality. The larger Q means the better video quality can be 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 practically considered by most of the previous work in fields of communication and networking researches. According to [14], the objective quality measure in (2) has effectively provided a lower bound of the video quality, 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 even some error recovery mechanism is in place. Otherwise, the frame is not decodable at all by the recipient regardless if it is received within the delay bound, D. Let $x\%$ be the statistical maximum fraction of bitstreams in a video frame that can be tolerated to lose. This parameter can be available from the adopted video coding and the video quality required by the real-time streaming video services. Hence, the statistical minimal amount of MPDUs of a video frame, denoted as L_i', that must be received by the recipient within D can be formulated as

$$L_i' = L_i \times (100 - x)\% \tag{3}$$

where L_i is the total number of MPDUs of frame i. If the available transmission resources can send all the L_i MPDUs of frame i, the recipient will perceive the optimal video quality of frame i; otherwise, at least L_i' MPDUs must be delivered within D in order to make the frame decodable with the minimal video quality. Assume that each MPDU consumes a single timeslot to transmit, at least L_i' timeslots are therefore required for sending frame i. Within the corresponding delay bound of frame i, the number of timeslots that actually available for transmitting MPDUs of frame i is indeed affected by various factors to be described in the rest of this section.

D. Modeling of Available Transmission Timeslots

A BS has a scheduling cycle to allocate certain amount of transmission resources, in terms of timeslots, to different servicing queues. The duration of each scheduling cycle, is denoted as S. Each timeslot is assumed to be long enough to send the data of a complete MPDU. When MPDUs of a frame arrive at the BS, they will be buffered in an assigned queue first. If there is no any MPDU of a previous frame already in that queue, the MPDUs of a newly arrived frame will be served and transmitted immediately in the coming scheduling cycle with the allocated amount of timeslots. Otherwise, they will need to wait in the queue for available transmission timeslots remained within their corresponding application delay bound. Each MPDU of a frame therefore may experience some delays in the queue within each scheduling cycle due to: 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; and 3) the time for transmitting its own MPDUs of the same frame. Hence, the actual available transmission timeslots for sending MPDUs of frame i in a queue at time t in scheduling cycle j can be derived as

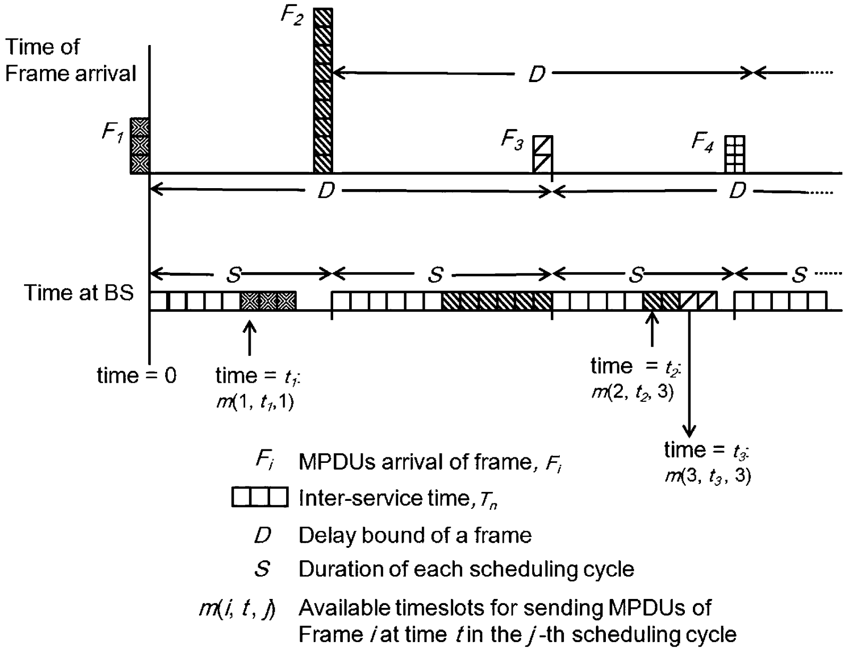
$$m(i,t,j) = D - T_w(i,t,j) \tag{4}$$

where t is the universal system 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 as defined in (1), $T_w(i,t,j)$ represents the total waiting time that the remaining MPDUs of frame i will experience in the queue estimated at time t in the scheduling cycle j, which can be derived by

$$T_w(i,t,j) = (j-i)S + T_{\text{in}} + T_{\text{leftover}}(i,t,j) + T_x(i,t,j) \tag{5}$$

where the first 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 is the duration of inter-service time that frame i has to wait in scheduling cycle j. $T_{\text{leftover}}(i,t,j)$ is the time consumed for transmitting MPDUs of previous frames in scheduling cycle j, and $T_x(i,t,j)$ is the total of 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 easily tracked by the BS system after each transmission, whereas, T_{in} and S are known from the selected scheduling policy.

Fig. 2 presents an example with all possible scenarios and their dynamics at time instants t_1, t_2 and t_3, in which all MPDUs of a frame can be sent successfully through the available transmission timeslots, $m(i,t,j)$, within D. In case of $\Delta = 1$, the application-layer delay bound, D, of the current



F_2
Time of Frame arrival
D
F_1
F_3
F_4
D
D
Time at BS
S
S
S
S
time = 0
time = t_1: $m(1, t_1, 1)$
time = t_2: $m(2, t_2, 3)$
time = t_3: $m(3, t_3, 3)$
F_i MPDUs arrival of frame, F_i
Inter-service time, T_n
D Delay bound of a frame
S Duration of each scheduling cycle
$m(i, t, j)$ Available timeslots for sending MPDUs of Frame i at time t in the j-th scheduling cycle

Fig. 2. Scenarios that MPDUs of a frame are sent completely.

frame is equal to $2P$ according to (1). Since the rate of a real-time video streaming traffic is varying, it is not trivial to choose an optimal value for the duration S of a scheduling cycle, whereas, it can also be adjusted regularly subject to the scheduling, call-admission and resources provisioning policies for the QoS requirements. For the simplicity but without loss of generality, S is assumed to be equal to the duration of frames producing rate at the media server (i.e., $S = P$) here.

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 D of F_1 is simply due to the inter-service time T_{in} in the first and second scheduling cycles. The total transmission timeslots available for sending the MPDUs of F_1 at time t_1 in the first scheduling cycle is

$$m(1, t_1, 1) = D - T_w(1, t_1, 1) = D - T_{\text{in}}. \quad (6)$$

At time t_2: All MPDUs of video frame F_2 could not be sent within the second 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 third scheduling cycle with the available transmission timeslots derived below

$$m(2, t_2, 3) = D - T_w(2, t_2, 3) = D - (S + T_{\text{in}}). \quad (7)$$

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 third scheduling cycle. The total available transmission timeslots for sending MPDUs of F_3 at time t_3 in the third scheduling cycle is derived below with the consideration of the leftover MPDUs of F_2

$$m(3, t_3, 3) = D - T_w(3, t_3, 3) = D - (T_{\text{in}} + 2). \quad (8)$$

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. Active Dropping Scheme

By (3), L_i' is the minimal number of MPDUs of frame i required by the recipient within D for the minimum decoding requirements. Although the reliable arrival of L_i' MPDUs of frame i can be guaranteed by a link-layer retransmission policy such as ARQ, retransmitting a lost MPDU certainly takes over some 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. Although these L_i' MPDUs can be normally prioritized by some cross-layer scheme for transmissions/retransmissions, it is reasonable to drop any MPDU of a frame immediately in the queue if it is believed that those L_i' MPDUs arriving at the recipient within D is not likely to happen. This is the fundamental motivation of the proposed AD scheme, which is further developed with an analytical approach. By jointly considering various factors in a TDMA-based BWA network that have not been comprehensively addressed in any previous literature, a probabilistic confidence value is analytically evaluated to decide when such dropping mechanism should be proactively exercised.

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 parameters: a) the current values of $m(i, t, j)$; and b) $L_i(t)$. 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 L_i' for frame i by (3). Transmitting MPDUs at time t will continue only if the confidence value, $C(L_i(t), m(i, t, j))$,

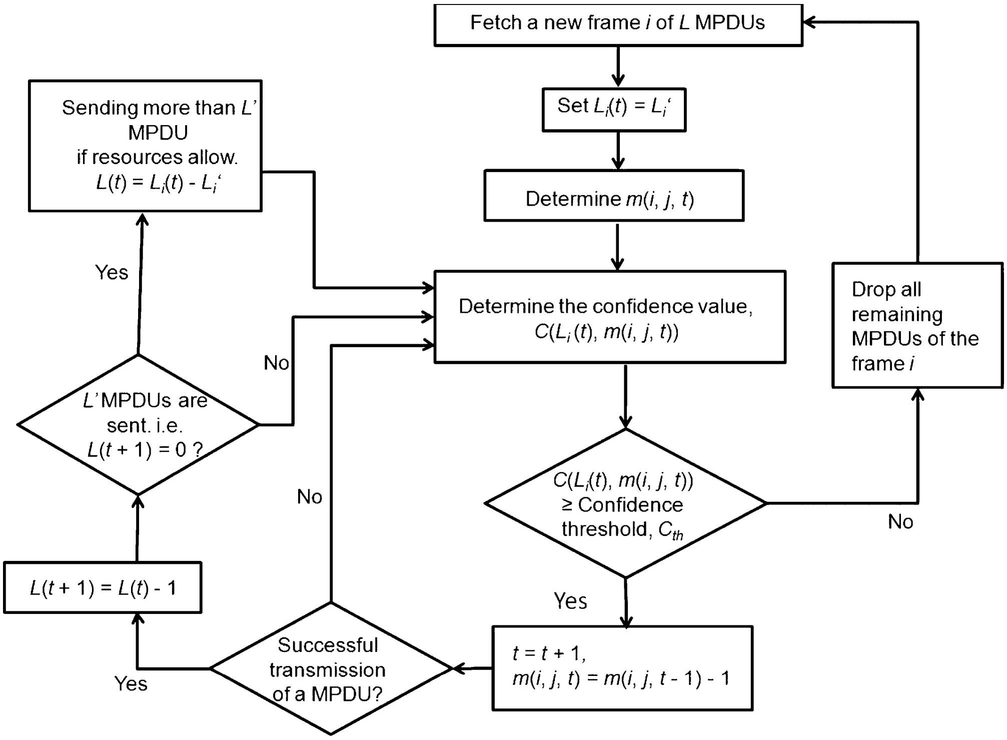

Fetch a new frame i of L MPDUs
Set Li(t) = Li'
Determine m(i, j, t)
Determine the confidence value, C(Li(t), m(i, j, t))
C(Li(t), m(i, j, t)) ≥ Confidence threshold, Cth
No
Drop all remaining MPDUs of the frame i
Yes
t = t + 1, m(i, j, t) = m(i, j, t - 1) - 1
Successful transmission of a MPDU?
No
Yes
L(t + 1) = L(t) - 1
L'MPDUs are sent. i.e. L(t + 1) = 0 ?
No
Yes
Sending more than L' MPDU if resources allow. L(t) = Li(t) - Li'

Fig. 3. Active dropping mechanism.

is not below a confidence threshold, C_{th}. The number of retransmissions is assumed infinite for simplicity, but it can also be a finite limit in our analytic model 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., see (9), as shown at the bottom of the page.

When $L_i(t)$ reaches zero, it will be intentionally set as the value of $L_i - L'_i$ for the purpose of sending more than L'_i MPDUs of frame i if more transmission timeslots are available. However, regardless of the value of $L_i(t)$, when the conference value $C(L_i(t), m(i,t,j))$ is less than C_{th} at any time instant, it is a strong statistical indication that frame i will not be delivered and decodable for the minimal video quality by the available $m(i,t,j)$ transmission timeslots within the associated D. The BS therefore immediately drops all MPDUs of frame i in the queue, and release the current remaining $m(i,t,j)$ transmission timeslots to subsequent frames or any other servicing queues. In summary, there is no difference leading to a frame loss event between the case of the required MPDUs of a frame is delayed due to the lack of transmission resources and that of those MPDU of the same frame are dropped proactively due to a lower-than-threshold probabilistic confidence for sending them within the application-layer delay bound. Fig. 3 illustrates a flowchart of the proposed AD scheme.

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 its 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. Instead, a more complicated wireless channel model could be employed here in the proposed analytical framework at the expense of higher complexity. However, it is not the focus of our intended contribution in this paper.

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 or 1 when the channel is "Good" or "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 packet-error-rate (PER), ϵ, and error burst length (EBL) as follows:

$$L_i(t+1) = \begin{cases} L_i(t) - 1, & \text{successful delivery of a MPDU at the timeslot } t \\ L_i(t), & \text{otherwise} \end{cases} \tag{9}$$

$$\begin{aligned} P_{BG} &= \frac{1}{\mathrm{EBL}}, P_{GB} = \frac{\epsilon}{\mathrm{EBL}(1-\epsilon)}, \\ P_{BB} &= 1 - P_{BG}, P_{GG} = 1 - P_{GB}. \end{aligned} \tag{10}$$

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 first MPDU of the frame to be transmitted when the channel is in "Good" state or "Bad" state, respectively. Similarly, states G_2 or B_2 represent the second MPDU of the frame to be transmitted when the channel is in "Good" or "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 ith 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 a maximum number of retransmission attempts may be defined by the retransmission policy as discussed before. Our analytical model on the probabilistic confidence, nonetheless, is independent of the retransmission limit, which is therefore flexible to incorporate with any retransmission policy in a cross-layer scheme. Transitions between states are described by the two-state wireless channel model through a one-step transition probability matrix, $\mathrm{Tr}(L')$ with the size of $2L \times 2L$, which can be derived by the current value of L' as illustrated in (11), shown at the bottom of the page. Assuming that all $m(i,t,j)$ transmission timeslots will be completely consumed to send the $L_i(t)$ most essential MPDUs of frame i at time t in the scheduling cycle j. Say, if there is any number of MPDUs of frame i still remained in the queue after $m(i,t,j)$ timeslots, the system must fall into one of the states in Fig. 4. This implies that those $L_i(t)$ MPDUs of frame i should be proactively dropped from the queue 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. Hence, the confidence of the BS to successfully send $L_i(t)$ MPDUs with only $m(i,t,j)$ timeslots at time t in scheduling cycle j can be defined as

$$C(L_i(t), m(i,t,j)) = 1 - \Pi_0 \mathrm{Tr}(L_i(t))^{m(i,t,j)} e \tag{12}$$

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, $\mathrm{Tr}(L_i(t))$ is the one step transition probability matrix with the size of $2L_i(t) \times 2L_i(t)$, and $e = [1 \cdots 1]^T$ is a column vector with a size of $1 \times 2L_i(t)$. Fig. 5 shows the relationship between the confidence value, and the difference between $m(i,t,j)$ and $L_i(t)$ which is denoted as Diff_{mL} (i.e., $m(i,t,j) - L_i(t)$). 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

$$\mathrm{Tr}(L')|_{L'=1} = \begin{vmatrix} 0 & 0 \\ P_{BG} & P_{BB} \end{vmatrix}$$

$$\mathrm{Tr}(L')|_{L'=2} = \begin{vmatrix} 0 & 0 & P_{GG} & P_{GB} \\ P_{BG} & P_{BB} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & P_{BG} & P_{BB} \end{vmatrix}$$

$$\cdots$$

$$\mathrm{Tr}(L') = \begin{bmatrix} 0 & 0 & P_{GG} & P_{GB} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ P_{BG} & P_{BB} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{GG} & P_{GB} & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & P_{BG} & P_{BB} & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & P_{BG} & P_{BB} & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & & & & & & \cdots & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & P_{GG} & P_{GB} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & P_{BG} & P_{BB} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & P_{GG} & P_{GB} \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & P_{BG} & P_{BB} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 & 0 & P_{BG} & P_{BB} \end{bmatrix} \tag{11}$$

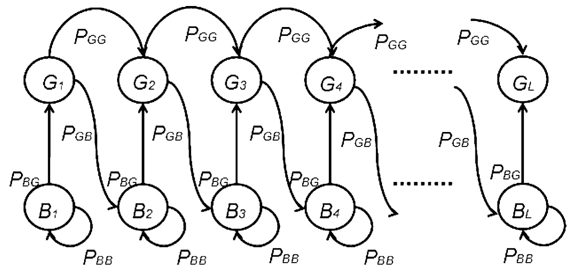


Fig. 4. Markov model for successfully transmitted MPDUs of a frame.

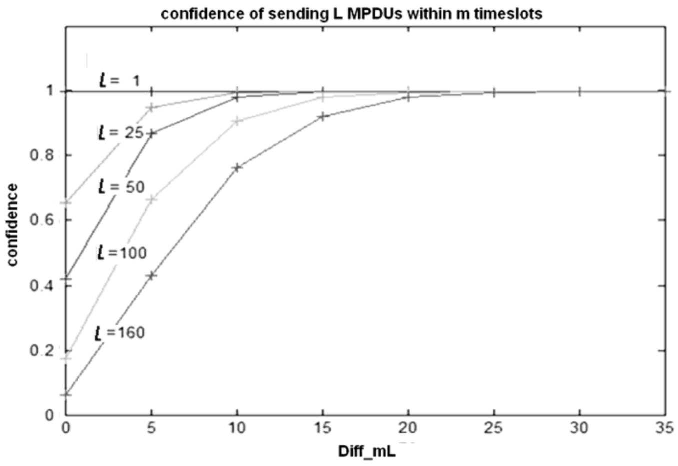


Fig. 5. Relationship between the confidence value and the difference of $m(i,t,j) - L(t)$.

gracefully balances between the tradeoff of video quality and the aggressiveness of proactive dropping decisions.

The relationship shown in Fig. 5 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}, it is more suitable to assign an I-frame with a smaller confidence threshold than that for a P-/ B-frame because: i) a successful MPDU transmission belonging to an I-frame is much favorable to the overall video quality than that from a B- or P-frame and ii) an I-frame usually has a much larger value of L, which starts from a lower confidence tendency curve (e.g., $L = 160$ in Fig. 5) to escalate to a higher one (e.g., $L = 50$) for the same Diff_{mL}. In other words, the dropping mechanism on different video frames should be differentiated by using different confidence thresholds, where I-frames are assigned a smaller confidence threshold in order to factor its importance to the video quality.

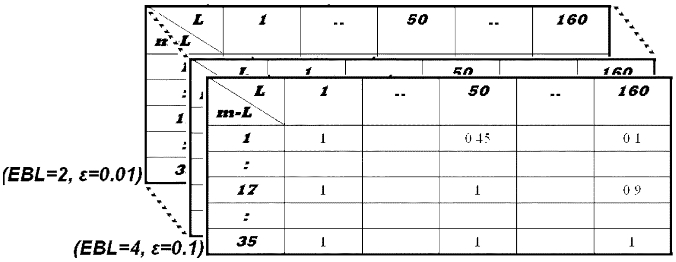
C. Implementation Issues

One of our major contributions of the proposed AD scheme is its simplicity to practically implement and integrate into a real world system. The associated complexity and implementability are to be discussed in the followings. As illustrated in Fig. 3, only the knowledge about $m(i,t,j)$, 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 parameters involved in (4)–(5) are either constants such as D, Q, Δ, and T_{in}, or tracked system parameters such as $T_{\text{leftover}}(i,t,j)$ and $T_x(i,t,j)$. To determine $C(L_i(t), m(i,t,j))$, on the other hand, requires the calculation of the one-step transition probability, $\text{Tr}(L_i(t))$ from (9) and (11). The complexity of the state space of $\text{Tr}(L_i(t))$ is $O((2L)^2)$, which is solely limited by the maximum video frame size. The current value of $L_i(t)$ is easily obtained and updated after every successful transmission in BS, which reflects the number of remaining MPDUs of frame i waiting in the queue to be transmitted. For an I-frame with a size of 8400 bytes plus the overhead of RTP/IP headers and a MPDU size of 52 bytes, about 160 MPDUs on average are generated to form a 320-by-320 matrix of $\text{Tr}(L_i(t))$ initially. After the first successful MPDU transmission, $\text{Tr}(L_i(t))$ is reduced to a 318-by-318 matrix, which is in turn used to evaluate the second MPDU transmission of the same frame, and so on. Assuming that each arithmetic operation involves two instructions and each instruction consumes a CPU cycle, a single 2-GHz processor in our experimental system only takes about 102.4 ms to calculate a 320-by-320 matrix of $\text{Tr}(L_i(t))$ (i.e., costs around 102 400 operations) by Matlab to obtain the confidence value $C(L_i(t), m(i,t,j))$.

In order to speed up the computational time required, a hardware-based lookup-table approach is introduced such that all possible confidence values are tabulated and stored in a hardware chip using 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}). $L_i(t)$ and $m(i,t,j)$ are the searching keys used in the lookup-table process. Similarly, a retrieved confidence value from the hardware-based lookup table is used to compared with a given confidence threshold for the dropping decision, instead of using a computed confidence value. The lookup-table process is constantly fast (i.e., $<$ 1 ms) due to the hardware capability along with a relatively small and finite searching space (i.e., 160 × 35). Note that the table above is for a specific pair of EBL and ϵ. Combining the channel feedbacks, multiple of such tables can be prepared and updated on-the-fly dynamically in the FPGA chip for various wireless channel condition characterized by EBL and ϵ. The hardware cost of maintaining a large number of these lookup tables to accommodate a wide range of diverse channel conditions is not significant at all when compared to a BS system. In this case, any evaluation of dropping decision 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 implementation cost and computational complexity are minimal. In case, a lookup-table process does not succeed in matching the actual channel condition, interpolations is performed for making an immediate dropping decision. A new lookup table for a new channel condition will be computed and inserted into the FGPA dynamically.

IV. Simulation Results

The proposed AD mechanism is evaluated using Matlab with the video traces containing a various amount of frames (200, 400, 800, 1600, 3200, and 6400). 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 IP packets respectively, and each packet is 2100 bytes such that all traces are simulating the variable bit rates. A MPDU has 52 bytes in size, which is possible to



L / m-L	1	..	50	..	160
1	1		0.45		0.1
:					
17	1		1		0.9
:					
35	1		1		1

(EBL=2, ε=0.01)
(EBL=4, ε=0.1)
L
n-L
1
:
1
:
3

Fig. 6. Hardware-based lookup table to implement AD.

be sent within a transmission timeslot. In order to evaluate the efficiency of our proposed scheme fairly, the simulation only compares it with a counterpart in which transmission/retransmission opportunities are "normally" prioritized for the first L' MPDUs of a video frame based on its importance to the minimal acceptable video quality. If there is any time resource available within the corresponding delay bound, the system transmit the rest of MPDUs of the same frame. This counterpart here is referred to as the "normal" prioritization-based cross-layer scheme, which indeed abstracted many previously reported cross-layer schemes (e.g., [4], [5], [8]–[10]) regardless of their actual utilization functions. 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 $(\text{EBL} = 2, \epsilon = 0.01)$ and a typical bad condition $(\text{EBL} = 4, \epsilon = 0.1)$.

A. *Rate of Frame Loss and Resources Released*

According to (2), 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 statistical maximum fraction, $x\%$, of a video frame that tolerable to data lose is set as 0.2 (i.e., $x = 20$) according to the MPEG standard. Fig. 7 compares the rates of frame loss by both schemes under two diverse channel conditions. With the bad channel condition (i.e., $\text{EBL} = 4, \epsilon = 0.1$), the proposed scheme not just has a much lower rate of frame loss even dropping video data proactively, but also the rate of frame loss is slowly growing when the size of video traces increases. On the contrary, the "normal" scheme suffers from an exponential increase of 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 $(\text{EBL} = 2, \epsilon = 0.01)$, 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 still yields the benefit of releasing resources for the other competing flows or subsequent frames by actively dropping MPDUs of any late arrival frame.

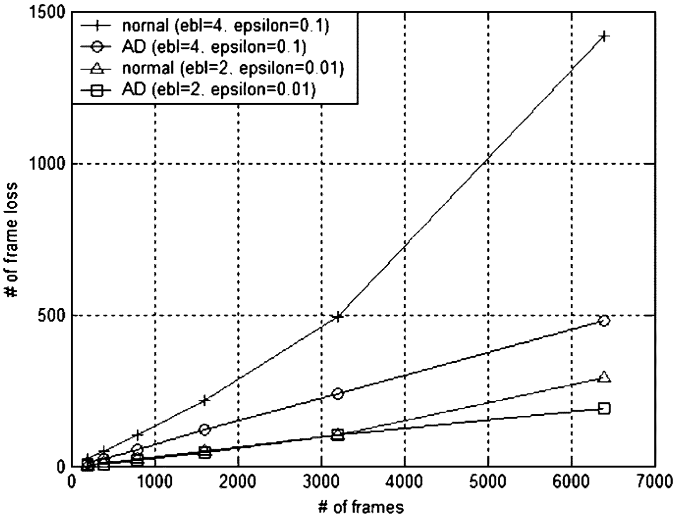


Fig. 7. Number of frame loss versus size of frames in the traces.

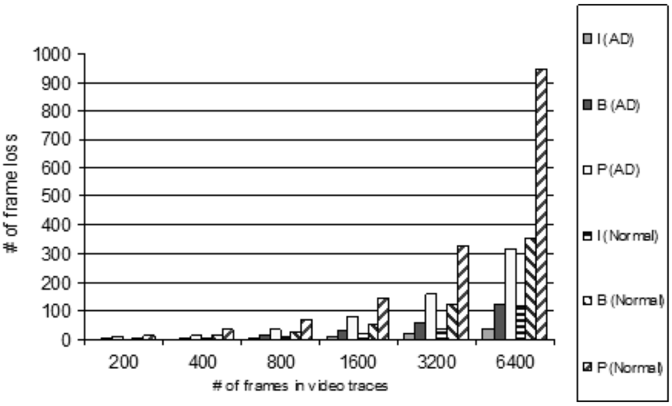


Fig. 8. Distribution of frame loss among frame types.

Fig. 8 shows a detailed breakdown of the frame losses in terms of the distribution among the type of frames (i.e., I-/ P-/ B-) under the bad channel condition $(\text{EBL} = 4, \epsilon = 0.1)$. Regardless of the number of frames in the video traces, the AD scheme generally incurs less frame losses in all types of frame. 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. Note that the results of the "normal" scheme in Fig. 8 indeed reflects the major shortcomings of many prioritization-based cross-layer schemes, in where the frame losses caused by the delay propagations become more serious when the size of video source increases.

Fig. 9 demonstrates the amount of resources released by AD under different sizes of the video traces. Obviously, the 'normal' scheme did not release any resource at all. In general, more resources are released by AD under the bad channel condition due to a larger chance of having more late arrival frames to be dropped ahead of time. The released resources can be allocated to the subsequent MPDUs of the frame to avoid delay propagation or to a competing video stream, which will never be possible by the 'normal' prioritization-based cross layer scheme.

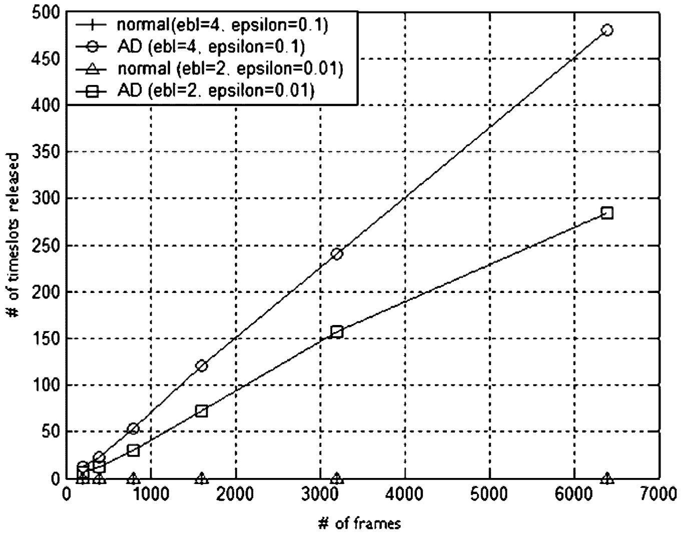


Fig. 9. Resource released versus number of frames in the traces.

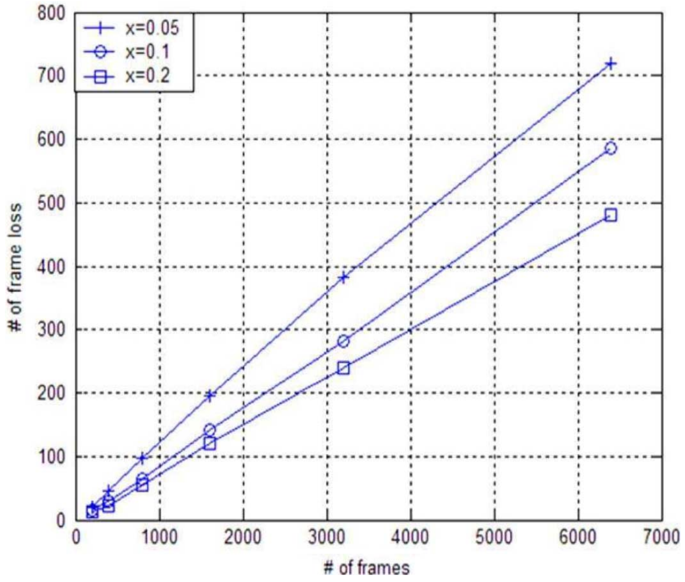


Fig. 10. Impact of $x\%$ in a frame to the number of frame loss.

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 and requires more MPDUs to be received through the same amount of available transmission timeslots within the application-layer delay bound. Hence, this naturally aligns to the conclusion in the Subsection III-B that the confidence value decreases given the same amount extra resources (Diff_{mL}) when L' increases (or $x\%$ decreases). The impact of $x\%$ of a video frame is insightful for achieving a graceful tradeoff between the rate of frame loss and adopted video coding technique for video quality requirements in a real-time video streaming application.

C. Differentiated Confidence Thresholds

A video trace with 6400 frames is evaluated with two different confidence thresholds: 1) $C_{\mathrm{th}} = 0.75$; and 2) $C_{\mathrm{th}} = 0.95$. For a higher confidence threshold $(C_{\mathrm{th}} = 0.95)$, more frame losses are caused in all frame types as shown in the first two groups of bar charts in Fig. 11. Different values of confidence threshold are applied to individual frame types, such as $DC_{\mathrm{th}} = 0.3$ for I-frame, $DC_{\mathrm{th}} = 0.75$ for P-frame, and $DC_{\mathrm{th}} = 0.95$ for B-frame, in order to reduce the loss of I-frames. The results are shown in the last group of bar charts using differentiated confidence thresholds in Fig. 11.

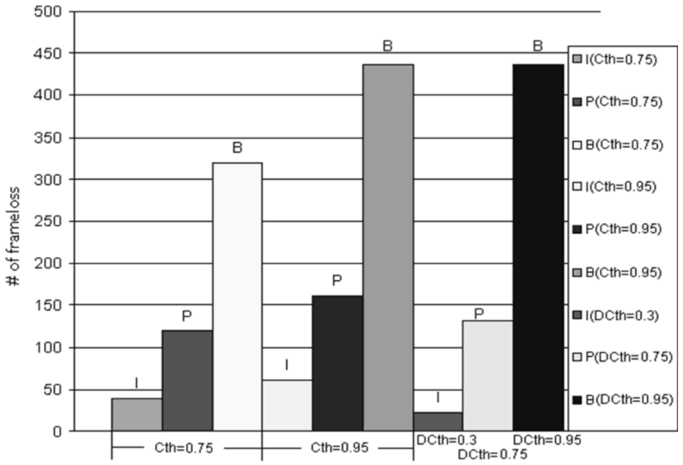


Fig. 11. Impacts of confidence thresholds to the frame loss.

As shown in Fig. 7, the AD scheme indeed incurs a smaller frame loss rate than the “normal” scheme, in which the confidence threshold, C_{th}, is set as 0.75. This results with more decodable frames by the AD scheme for a better video quality, in term of Q, according to (2). The same observation can be concluded when the size of the video source increases as shown in Fig. 12(a). A poorer video quality is seen expectedly using a higher confidence value (e.g., $C_{\mathrm{th}} = 0.95$), which could be mitigated using differentiated confidence thresholds for the improved video quality as shown in Fig. 12(b).

In summary, the confidence threshold will determine the resultant decodable frame rate perceived by a recipient, which should be a parameter subject to dynamic configuration through a closed-loop control. If the video quality, Q, requirement has been already satisfied at the recipient software, the confidence threshold at the BS should be increased to release more resources for the other competing flows when the inter-service time is large (which implies the network resources are not abundant). The adaptive adjustment of the confidence threshold can be directly adjusted through the current video quality, Q, evaluated by (2) to achieve the adaptive resource releases without failing to satisfy video quality requirement.

V. Conclusion

In this paper, a MAC-layer AD scheme to achieve effective resource utilization, which can satisfy the application-layer delay for real-time video streaming in TDMA-based 4G broadband wireless access networks, such as WiMAX and TD-LTE. 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

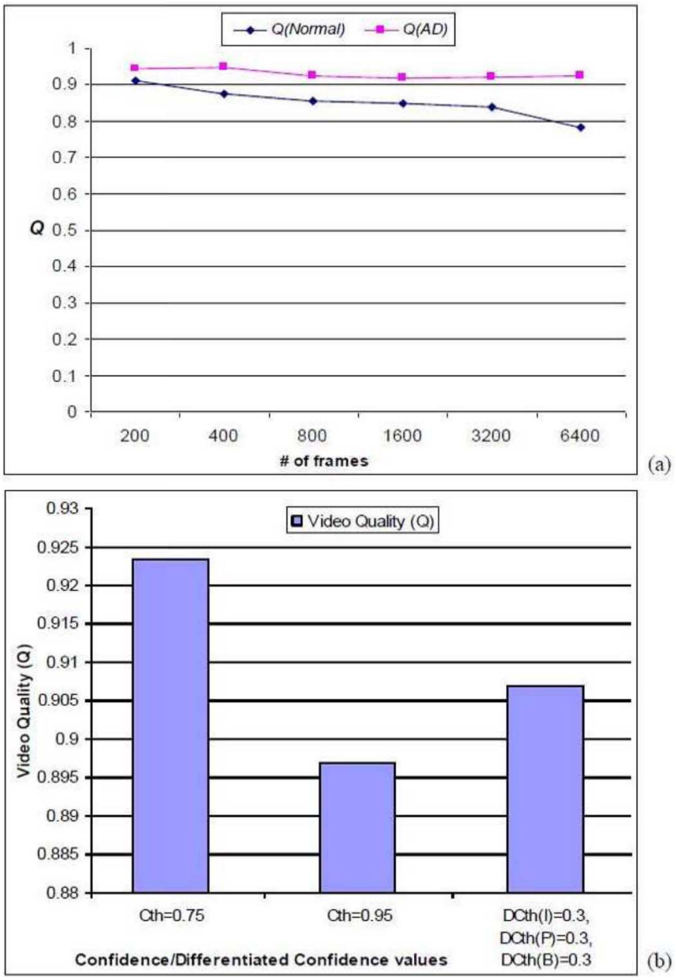


Fig. 12. Relations between video quality and confidence thresholds.

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 practically implemented in a real-world BS system through software- or/and hardware-based approaches. The effectiveness of the scheme is proved to be better than that of a prioritization-based cross-layer scheme by way of extensive simulations in the aspects of frame loss rate and released resources. The impacts of maximum losable fraction of a video frame and the confidence threshold toward the rate of frame loss are also evaluated. A video quality measure is also adopted to evaluate the scheme for the perceived video quality, which concludes that differentiated and adaptive confident thresholds can gracefully achieve better resource releases while satisfying the video quality requirement. The proposed scheme and the analytical framework for the dropping decision are believed to be applicable to many future real-time wireless video streaming systems in emerging 4 G BWA networks.

Appendix

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

The Expected Number of Decodable I-Frames $(N_{\text{dec}-I})$*:* 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

$$N_{\text{dec}-I} = (1-p)^{C_I} \times N_{\text{GOP}} \tag{13}$$

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

Expected Number of Decodable P-Frames $(N_{\text{dec}-P})$*:* 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

$$N_{\text{dec}-P} = (1-p)^{C_I} \times \sum_{j=1}^{N_p} (1-p)^{j \times C_p} \times N_{\text{GOP}} \tag{14}$$

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

Expected Number of Decodable B-Frames $(N_{\text{dec}-B})$*:* 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, the consecutive B-frames are considered 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. Hence, the expected number of correctly decodable B-frames for the whole video is

$$\begin{aligned} M_{\text{dec}-B} &= \left[(1-p)^{C_I + N_p C_p} + \sum_{j=1}^{N_p} (1-p)^{j \times C_p} \times (1-p)^{C_B} \right] \\ &\quad \times (M-1) \times (1-p)^{C_I + C_B} \times N_{\text{GOP}} \end{aligned} \tag{15}$$

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 for Local and Metropolitan Area Networks—Part 16: Air Interface for Fixed and Mobile Broadband Wireless Access Systems*, IEEE Standard 802.16e-2005, 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," *Comput. 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. 37th Annu. Conf. Information Sciences and Systems*, Baltimore, MD, Mar. 12–14, 2003.

[4] Y. Shan and A. Zakhor, "Cross layer techniques for adaptive video streaming over wireless networks," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Lausanne, Switzerland, Aug. 26–29, 2002.

[5] P. Bucciol, 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. IEEE GLOBECOM*, Nov.–Dec. 2004, vol. 5, pp. 3027–3031.

[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 J. Appl. Signal Process.*, 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 J. Select. Areas Commun.*, vol. 21, no. 12, pp. 1685–1698, Dec. 2003.

[8] C. Krasic, J. Walpole, and W.-C. Feng, "Quality-adaptive media streaming by priority drop," in *Proc. 13th Int. Workshop on Network and Operating Systems Support for Digital Audio and Video*, Jun. 2003, pp. 112–121.

[9] Y. Huang, R. Guerin, and P. Gupta, "Supporting excess real-time traffic with active drop queue," *IEEE/ACM Trans. Netw.*, 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," in *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance*, Jul. 2003, p. 342.

[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," in *Proc. 21st Int. Conf. Advanced Information Networking and Applications (AINA)*, May 21–23, 2007, pp. 451–458.

[12] J. Mitchell and W. Pennebaker*, MPEG Video: Compression Standard*. London, U.K.: Chapman and Hall, 1996.

[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. 20th Int. Conf. Advanced Information Networking and Applications (AINA'06)*, 2006, vol. 1, pp. 565–572.

[14] B. E. Wolfinger, "On the potential of FEC algorithms in building fault-tolerant distributed applications to support high QoS video communications," in *Proc. ACM Symp. Principles of Distributed Computing (PODC'97)*, Santa Barbara, CA, 1997.

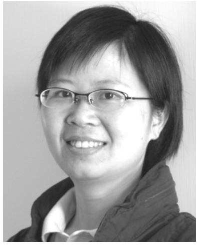
[15] A. Ziviani, B. E. Wolfinger, J. F. Rezende, O. C. M. B. Duarte, and S. Fdida, "Joint adoption of QoS schemes for MPEG streams," *Multimedia Tools and Applic. J.*, vol. 26, no. 1, pp. 59–80(22), May 2005.

[16] M. Kalman, S. T. Eckehard, and B. Girod, "Adaptive playback for real-time media streaming," in *Proc. IEEE Int. Symp. Circuits and Systems*, May 26–29, 2002, vol. 1, pp. I-45–I-48, 2002 (ISCAS 2002).

[17] M. Zorzi, R. R. Rao, and L. B. Milstein, "Error statistics in data transmission over fading channels," *IEEE Trans. Commun.*, vol. 46, no. 11, pp. 1468–1477, Nov. 1998.

James She (M'09) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2009.

He is a Research Fellow in the Computer Laboratory, University of Cambridge, Cambridge, U.K. His current research interests include wireless multimedia communications, wireless/mobile social networks, content delivery networks, and wireless sensor devices/networks.



Fen Hou (M'09) received the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2008.

She is currently a Postdoctoral Research Fellow in the Department of Information Engineering, Chinese University of Hong Kong. Her research areas include QoS provisioning, resource allocation and scheduling for multimedia service in wireless networks, broadband wireless networks, delay-tolerant networks (DTNs), and cognitive radio networks.

Basem Shihada (M'09) received the B.Sc. degree from the United Arab Emirates (UAE) University, Al Ain, UAE, in 1997, the M.Sc. degree from Dalhousie University, Halifax, Nova Scotia, in 2001, and the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2007, in computer science

He is an Assistant Professor at King Abdullah University of Science and Technology, Saudi Arabia. His research interests include high-speed optical networks, wireless networks, multimedia over wireless networks, wireless mesh networks, transport-layer protocols, resource management, and system security.

Pin-Han Ho (M'02) received the B.Sc. and M.Sc. degrees from the Electrical Engineering Department, National Taiwan University, Taipei, Taiwan, R.O.C., in 1993, 1995, and 2002, respectively, and the Ph.D. degree from Queen's University, Kingston, ON, Canada, in 2002.

He is an Associate Professor in the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON, Canada. His 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.

Dr. Ho is the recipient of the Distinguished Research Excellence Award in the ECE Department, University of Waterloo, the Early Researcher Award in 2005, the Best Paper Award at SPECTS'02, ICC'05 Optical Networking Symposium, and ICC'07 Security and Wireless Communications Symposium, and the Outstanding Paper Award at HPSR'02.