



Robust video streaming over wireless LANs using multiple description transcoding and prioritized retransmission

Chih-Ming Chen ^{a,b}, Chia-Wen Lin ^{a,*}, Hsiao-Cheng Wei ^c, Yung-Chang Chen ^a

^a Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan

^b Telecommunication Laboratories, Chunghwa Telecom Co., Ltd, Taoyuan 326, Taiwan

^c Department of Computer Science and Information Engineering, National Chung Cheng University, Chiayi 621, Taiwan

Received 28 June 2005; accepted 2 February 2007

Abstract

Video transport over wireless Local Area Networks (LANs) usually suffers from signal fading, noise interference, and network congestion, leading to time-varying packet loss rate and fluctuating effective bandwidth. We propose in this paper an adaptive error control scheme to adaptively insert error-resilience features into a compressed video in the media gateway which serves roaming clients. Our scheme divides the error control operation into the single description (SD) and multiple-description (MD) modes according to channel condition estimation. In the SD mode, a content-based prioritized retransmission method is proposed to mitigate the error propagation due to packet loss in video transport over wireless LANs. The proposed prioritized retransmission scheme determines the retransmission schedule of a lost packet according to the packet's loss impact that is obtained in a front-end encoding process. In the MD mode, we propose a channel-aware MD transcoding method to take advantage of MD coding and path diversity so as to further mitigate the error propagation due to packet loss caused by transient channel switching while roaming in wireless LANs. Experimental results show that the proposed scheme effectively mitigates the error propagation due to packet loss, while maintaining low extra computational complexity and low overhead cost. The proposed method is suitable for realtime streaming of prestored videos.

© 2007 Elsevier Inc. All rights reserved.

Keywords: Wireless video; Video streaming; Error resilience transcoding; Automatic retransmission

1. Introduction

Due to the simplicity of configuration and the relatively low cost for network setup, accessing the Internet via wireless LANs is getting more and more popular. Using mobile devices, such as notebooks, Personal Digital Assistants (PDAs), and smart phones, wireless LANs are available for various applications anywhere in campuses, in offices, and even at home. Among all applications, multimedia applications, such as multimedia streaming, multimedia messaging, video telephony, and video-on-demand are the most interesting and popular applications, since multime-

dia applications are easier to be accepted, and are closer to human life. However, several characteristics of wireless LANs, such as limited bandwidth, high data error/loss rate, and unstable network condition, pose a great challenge on enabling multimedia applications. Video transport over wireless networks may suffer from signal fading, noise interference, network congestion, and handoff, which will usually cause data error or packet loss. Even a single bit error in a packet can cause the loss of a whole packet, if the number of corrupted bits goes beyond the error correction capacity of error correction codes. This packet-loss problem may lead to serious video quality degradation, which not only affects the quality of current frame, but also leads to error propagation to subsequent frames because of the motion compensated prediction used [1].

The benefits for wireless application are not only without wire connections but also mobile ability, especially

* Corresponding author. Tel.: +886 3 5731152; fax: +886 3 5715971.

E-mail addresses: cmchen2@cht.com.tw (C.-M. Chen), cwlin@ee.nthu.edu.tw (C.-W. Lin), scwei@msl.url.com.tw (H.-C. Wei), ycchen@ee.nthu.edu.tw (Y.-C. Chen).

for mobile computing devices, such as smart phones, handheld devices, and multimedia personal digital assistants (PDAs). However, the heterogeneity of client networks and devices makes it very difficult to adapt the video contents to a wide degree of different client channel conditions, especially for mobile users who usually roam in wireless networks. In order to achieve error robustness for transmitting video over wireless networks, a video transcoder can be placed in an intermediate network node (e.g., mobile switch/base-station, proxy server, and media gateway) connected to a high-loss network (e.g., wireless network or highly congested network) to adapt the non-error-resilient compressed video streams into error-resilience-capable streams [2,3]. Fig. 1 illustrates an exemplary three-tier streaming system using a wireless LAN as an extension to the existing wired infrastructure, offering local end-point devices the convenience of wireless connections. This three-tier streaming system involves a streaming server, a media gateway (MG) with a transcoder inside, and mobile client terminals. It can be divided into two parts: the first including the path from the streaming server to the media gateway and the second containing the path from the media gateway to wireless terminals via the wireless access points (APs). The media gateway serves as a proxy/adaptation-engine to adapt multimedia contents to heterogeneous devices. To achieve a high level of accessibility, it needs to provide a reliable and efficient transmission.

There exist several channel and source coding schemes which can be used in the media gateway to enhance the error robustness of outgoing video bitstream. Forward Error Correction (FEC) and Automatic Retransmission reQuest (ARQ) are the two most commonly used channel coding schemes for error protection. In the IEEE 802.11 standard [4], the concept of application-level framing (ALF) is often used for packetization, in which packets with a corrupted bit-number greater than the capacity of error correction will be dropped. In this way, a wireless LAN channel becomes a packet erasure channel. ARQ is particularly useful to combat against packet erasure errors

and has been adopted in several existing packet protection methods [5–12] for video streaming over wireless LANs. For example, a typical scheduling-oriented retransmission method which takes into account the play-out deadline based on the Early-Deadline-First (EDF) principle was presented in [5]. The EDF method gives priority to the lost packets with earliest play-out deadlines in transmission, which may lead to the overriding of regular packets by the retransmitted packets. In [6], the authors proposed a class of packet scheduling algorithms for wireless video streaming by applying different deadline thresholds to video packets of different importance. The importance of a packet is determined by its relative position within its group of pictures (GOP) and motion-texture context. In [7], an error model was proposed to accurately evaluate the influence of transmission errors on the decoded picture quality in a packet-loss network. The effects of intra-coding and spatial loop filtering are both taken into account in the model. In [8], five schemes based on uniform, frame level reference, slice-level reference, motion, and motion combined with slice-level reference, are respectively proposed for loss differentiation. The conditional retransmission scheme proposed in [9] uses the concealment error and the channel feedbacks to determine whether a packet is worthwhile to retransmit. It provides a rate-distortion analysis of the trade-off between the saved bits due to the reduced retransmission and the increased distortion resulting from the concealment error of not-retransmitted packets. A frame-based scheduling (FBS) scheme was proposed in [10,11] to improve video streaming over 802.11 WLANs. In FBS, the influence (i.e., the effect of error propagation) of each frame in a GOP is evaluated according to the number of frames that are inter-coded with respect to the frame. The FBS scheme then dynamically determines whether to send or discard a packet in one frame according to the influence and retransmission deadline of this frame.

Commonly used error resilient source coding tools include data partitioning, synchronization marker, reversible variable length codes (RVLC), Error resilience entropy coding (EREC), Multiple Description Coding (MDC), error tracking, and adaptive intra refresh (AIR), etc. [1]. MDC is kind of joint source and channel coder. The objective of MDC is to enhance the reliability (or error resilience) of data transmission under channel failures, by employing the diversity of channels. MDC [13,14] inserts additional rate to make the bitstream more resilient to transmission errors. If data are sent over multiple independent channels, the possibility of failures of all channels is greatly reduced, leading to a high possibility of receiving correct data from at least one channel, thereby making it easier to estimate the original data with acceptable quality. The redundancy intentionally added in MDC will, however, consume more bandwidth, leading to a trade-off between coding efficiency and error resilience. Several MDC schemes have been proposed recently to address the error resilience coding problem. In [15], a video coder based on MD scalar quantization (MDSQ) with two inde-

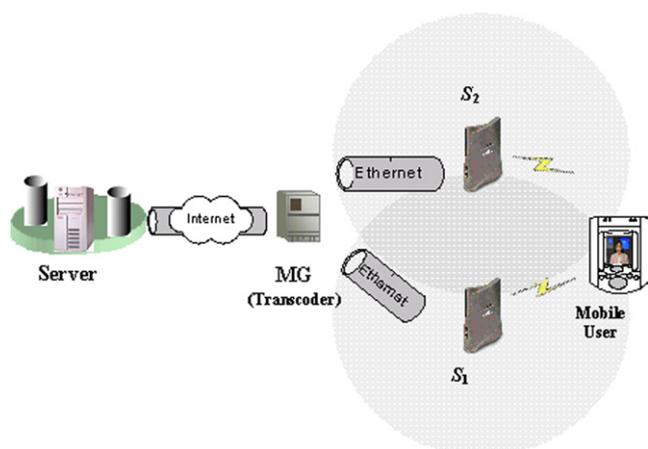


Fig. 1. Illustration of a three-tier video streaming with two diverse wireless LAN links between the media gateway and the mobile terminals.

pendent prediction loops was proposed. In addition to error resilience, multiple path transport (MPT) is always applied to MDC, which can mitigate congestion in network as well as increase overall network utilization. Several research works integrating MPT with MDC have been proposed for video communication over the Internet [16,17] or wireless networks [18–20]. A comprehensive survey about various MDC techniques and video transport systems integrating MPT and MDC can be found in [14].

Video streaming with roaming across multiple basic service sets (base stations) in wireless LANs always poses considerable challenges. In a wireless LAN, a handoff to find another AP for a mobile client will occur when the original channel serving the client cannot provide satisfactory connection quality. In general, a sequence of events must transpire for roaming. At first, the client has to decide when to roam based on the factors such as signal strength and missed beacons. The second action is to decide where to roam, and then initiate a roam. Finally, the client must resume the existing application session. After the long roaming duration, it performs a nomadic roaming which may cause very heavy transmission data loss, leading to severe transient video quality degradation. We shall show in this paper how a channel handoff caused by a roam in a wireless LAN will hurt the video quality.

We propose an adaptive error control scheme to adaptively insert error-resilience features into a compressed video in the media gateway which serves roaming clients. Our scheme divides the error control operation into the SD and MD modes based on channel condition estimation. In the SD mode, a content-based prioritized ARQ scheme is used to mitigate error propagation due to packet loss. We consider the application scenario that, in the channel, a fixed bandwidth is assigned to a mobile terminal and no dedicated bandwidth is reserved for resending lost packets for the terminal due to limited bandwidth. This implies that the retransmitted packets will compete with the regular video packets for the constrained bandwidth resources. In the proposed ARQ method, the retransmission of a lost packet is scheduled by the packet's importance level. The importance level of a packet is measured by estimating the relative amount of error propagation should the packet be lost. In our method, the encoder utilizes the motion information generated in the encoding process and the concealment distortion to estimate the error propagation effect of each packet within a GOP if the packet is lost during transmission. The measurements are performed only once and the results are stored in the streaming server for guiding the packet retransmission scheduling and decision.

To address the channel handoff problem in the MD mode, we also propose a channel-aware MD transcoding scheme to adaptively divide the incoming bitstream into multiple descriptions which are sent to the client terminal data via diverse channels. Compared with traditional MD coding (MDC) scheme, the proposed channel-aware MD transcoding framework can not only make best use of path diversity to effectively mitigate packet loss during channel

switching, but also avoids sacrificing the coding efficiency in normal channel conditions. We establish a testbed for real-time multipath streaming to verify the performance of the proposed schemes.

The proposed scheme is especially useful for applications involving real-time streaming of pre-stored video (e.g., video/movie on demand, non-live TV broadcasting, and multimedia messaging). In such applications, video clips are encoded offline and pre-stored in a server, making it feasible to deploy, in the off-line encoding process, those useful feature extraction methods which are computationally too extensive to be used in on-line processing. These compressed video streams will then be transcoded and streamed to client terminals for decoding and playback in real-time per clients' requests. The pre-extracted video content features can be exploited to achieve better performance during the real-time streaming/transcoding process.

The rest of this paper is organized as follows. Section 2 gives an overview of the proposed system. Our content-based prioritized ARQ scheme is described in Section 3. Section 4 presents the proposed channel-aware MD transcoding scheme. In Section 5, experiments are conducted to verify the performance of the proposed method. Finally, our conclusion is drawn in Section 6.

2. Overview of the proposed system

2.1. Multipath streaming testbed

To evaluate the impact of packet loss due to channel handoff on video quality degradation, we established in a laboratory environment a testbed which is composed of four personal computers as illustrated in Fig. 2. These four computers emulate a media gateway which implements an MDC transcoder, two base-stations (S_i and S_j), and one mobile client, respectively. In this test scenario, each base-station reports to the transcoder about the packet loss rates (PLRs) and round-trip times (RTTs) of the channels between the base-station and the mobile clients it serves.

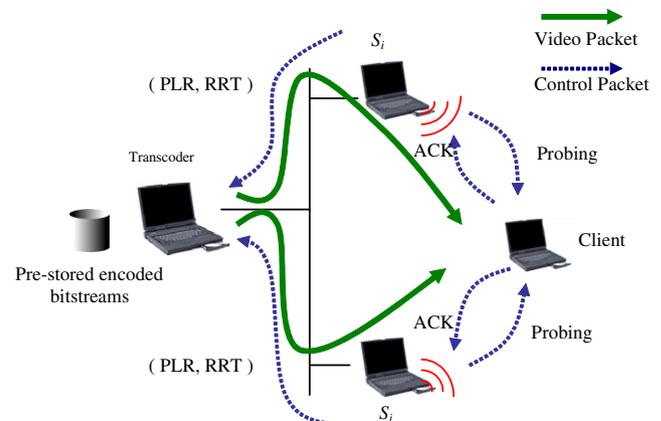


Fig. 2. Proposed testbed for real-time multipath streaming with one media gateway (video transcoder), two base-stations, and one mobile client.

On top of this testbed, we realize possible channel handoffs due to roaming among the base-stations and mobile clients. For example, Fig. 3 illustrates the packet loss rate and the video quality degradation due to a channel handoff caused by a roam of the mobile client from S_i to S_j in the testbed. In this experiment, the QCIF (176×144) *Foreman* sequence is encoded at 384 Kbps and 30 fps by a public-domain MPEG-4 software encoder [33]. The first frame is coded as an I-frame, and all subsequent frames are coded as P-frames. Each slice which contains one row of macro-blocks is encapsulated into one packet. In wireless networks, mobile clients always attempt to connect to the best available base-station. The result shows that the channel handoff will result in serious transient packet losses. The PSNR performance degradation due to the packet loss in channel handoff can be up to 15 dB. The detailed setting of the testbed is described in Section 5.2.

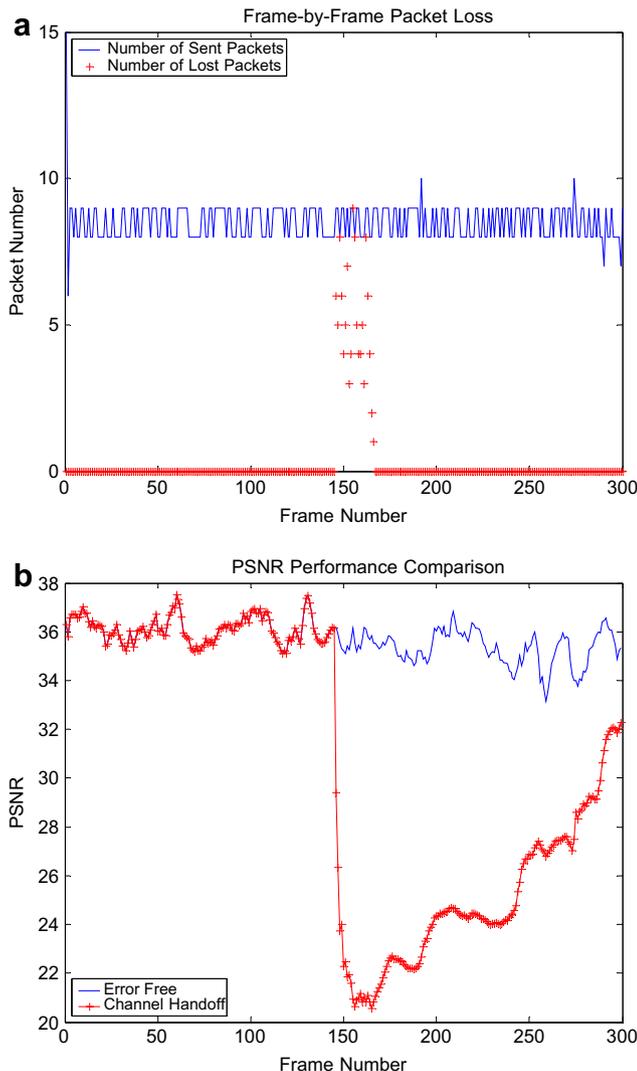


Fig. 3. Effect of handoff of video across two access points in a wireless LAN: (a) packet loss due to handoff; (b) PSNR quality degradation during handoff.

In the testbed, base-stations S_i and S_j are responsible for detecting the clients located in their communication zones, and then reporting to the media gateway the channel statistics about the clients they are serving. According to the information, the media gateway maintains a path-list table that records which clients can be reached by each station. These paths are ranked according to their channel conditions. The status of a path from a station to a client is updated when any new channel information about the client is received. When the server does not receive any information about the client from any station within a pre-specified timeout interval (namely, the lifetime of the client), the client will be marked as temporarily non-connectable. According to the path-list table, the media gateway chooses to deliver the video data to the client via either only one channel with good quality, or two channels with acceptable qualities but may not be very reliable.

2.2. Proposed error control framework

The proposed system architecture with channel-aware MD transcoding and content-based prioritized ARQ is shown in Fig. 4. In this system, the media gateway (enclosed by the shaded block) on-the-fly collects the channel statistics fed back from the corresponding base-stations. Two channel statistics: round-trip time (RTT) and the packet loss rate (PLR) are collected for estimating the channel conditions, no matter the packet loss and the round-trip delay are caused by network congestion or the unreliable transmission media. According to the channel estimation, the media gateway separates the error control operation for a mobile terminal into two modes: the SD and MD modes. In the case that one channel's condition is good enough for delivering the video, the media gateway will choose the SD mode. On the other hand, when the mobile terminal moves to the boundary of two coverage areas served by two distinct base-stations, the channel condition may become relatively poor then channel switching is necessary. When detecting such channel switching, the media gateway will seek whether there exists another channel with acceptable channel quality to be used simultaneously for path diversity. If the alternative path is available, the server will choose the MD mode to diversify the transmission of video data. The media gateway will subsequently switch back to the SD mode when detecting that the client completes the handoff. As a result, the "Adaptive Error Control" (AEC) block adaptively deploys two error control schemes: content-based prioritized ARQ and MD transcoding for the SD and MD modes, respectively.

Fig. 5 depicts the proposed adaptive error control scheme for the three-tier streaming application scenario illustrated in Fig. 1. For simplicity of implementation but without loss of generality, we assume there are two transmission paths between the media gateway and a mobile terminal. The SD mode indicates that the mobile terminal stays in a coverage area served by a dominate base-station.

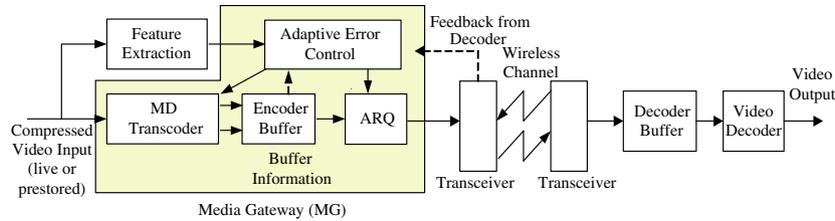


Fig. 4. Proposed system architecture consisting of adaptive MD transcoding with channel estimation based on feedback information and content-based prioritized ARQ.

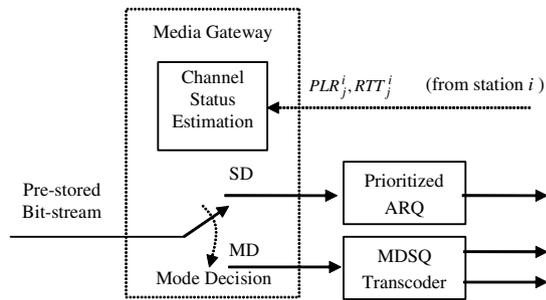


Fig. 5. Proposed adaptive error control in the media gateway.

In this case, the channel bandwidth assigned to the mobile terminal is kept fixed since the terminal needs to share the wireless channel with other client terminals. In the SD mode, we propose a content-based prioritized ARQ method which adaptively determines the retransmission schedule of a lost packet based on its loss impact (e.g., the level of error propagation). The packet-level loss estimation is done in a front-end encoding process, which is usually performed off-line, and stored in the streaming server. The pre-analyzed content information is sent to the media gateway as auxiliary data (metadata) to guide the ARQ scheduling while streaming.

The ARQ scheme, however, may not be able to protect video packets well under highly bursty packet loss situations which often happen during the channel handoff as illustrated in Fig. 3. To address the channel handoff problem, we propose to use in the MD mode an MDSQ transcoder to split the incoming bitstream into two descriptions which are sent via two diverse channels served by two base-stations. This method effectively enhances error robustness against the worst situation in which both channels are not reliable, whereas the coding efficiency is reduced due to the redundancy intentionally added to the two descriptions. Although the MD transcoding will increase the overall bitrate, the data rate in a single channel will not be increased (actually it will be reduced) since the data are split into two separate channels. However, a dynamic bandwidth management mechanism may need to be used to handle the traffic variations in channels due to the channel handoff. As MD transcoding will change the loss impacts of packets, in our method, the prioritized ARQ is performed only in the SD mode, although the loss estimation obtained in the SD mode still somehow reflects the relative importance of lost packets in the MD mode.

2.3. Channel statistics used for error control

With the medium access mechanism of IEEE 802.11 [4], should the condition of a channel become poor, the transmit station may need to resend the lost packets several times, leading to increased delay time for the packets to arrive at the receiver. A prolonged round-trip time therefore reflects the loss probability that is due to impaired channel error caused by the signal fading or noise interference. Furthermore, a poor channel condition usually also leads to a high packet loss rate, since a packet will be dropped after its retry limit has been reached. The round-trip time and packet loss rate are therefore the two major attributes that can be utilized to estimate the channel status [28,29].

In our experiments, the base-stations use the User Datagram Protocol (UDP) to send the probe frames to the clients every 0.1 s and then receive the ACK frames from the clients. The higher the probing period, the heavier the extra channel bandwidth consumed by probe frames. On the other hand, the lower the probing period, the lower the refresh rate of channel status. Only a departure timestamp and a sequence number are carried in each probe frame and also in the corresponding ACK frame. For the probing, it is difficult to measure the delay time accurately unless the clock times of the stations and the client are well synchronized. Using the round-trip time to replace the one-way delay to estimate the channel condition is more reliable, since the receiver sends back the ACK frame to the station immediately via the same channel. In our method, each base-station itself calculates the average RTT of the i th channel, \overline{RTT}^i , and the corresponding average packet loss rate, \overline{PLR}^i , from a base-station to the client in a sliding time interval with C packets as follows:

$$\overline{RTT}^i = \frac{1}{C} \sum_{k=0}^C RTT_k^i \quad (1)$$

$$\overline{PLR}^i = \frac{1}{C} \sum_{k=0}^C PL_k^i \quad (2)$$

$$\max RTT^i = \max(RTT_k^i, \max RTT^{i-1}) \quad (3)$$

where RTT_k^i stands for the round-trip time of the k th packet (ahead from the current time instance) via the i th channel, and PL_k^i denotes its corresponding status of loss. If the

packet is lost or corrupted, $PL_k^i = 1$, and its RTT_j^i is assigned to be $\max RTT^i$.

3. Content-based prioritized ARQ

ARQ is a useful channel coding tool to mitigate the effect of error propagation due to packet loss, especially for video streaming over a packet erasure channel like wireless LANs. However, in a fixed-rate channel without extra bandwidth reserved for resending lost packets, the retransmitted packets will compete for the limited bandwidth with regular video packets. Because each video packet contributes unequal importance to the video content, thereby leading to a different level of error propagation should it be lost, the retransmission of a lost packet should be properly scheduled by its importance so as to maximize the visual quality under the bandwidth limitation. In this work, we propose a content-adaptive prioritized ARQ scheme, based on pre-analyzed packet importance information, as a channel coding tool to enhance the error robustness of streaming video. The level of packet importance is measured by estimating the loss-impact value (i.e., the error propagation effect) of each packet. In our method, this measurement is performed offline and the results are stored in the streaming server for guiding the packet retransmission scheduling and decision in the media gateway.

3.1. Packet-level loss estimation

To estimate the impact of each lost packet, we use our previous method proposed in [3] which characterizes the pixel-level error propagation effect using two parameters: pixel reference count (PRC) and pixel-wise concealment error (PCE). This method first calculates the pixel-level loss-impact (LI) metric as the product of PRC and PCE by

$$LI(x, y, n) = PCE(x, y, n) \times PRC(x, y, n) \quad (4)$$

where $PRC(x, y, n)$ represents the frequency of pixel (x, y) of frame n being referenced by pixels in the following frames within a GOP in the motion-compensated prediction (MCP) process as illustrated in Fig. 6. It can be calcu-

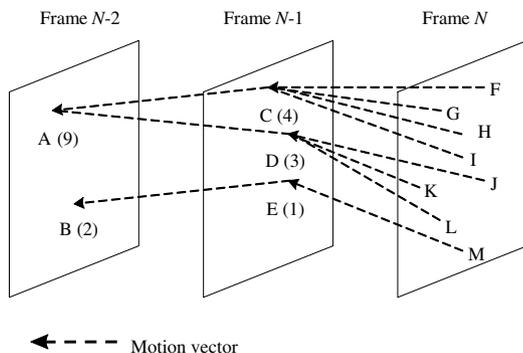


Fig. 6. An illustration of calculating the pixel reference count (PRC). Assuming frame N is the last frame of a GOP, the numbers in the braces indicates the PRCs.

lated recursively by summing up the individual reference counts of pixels in frame $n + 1$ which reference to pixel (x, y) of frame n in the reverse tracking order from the last frame to the first frame of a GOP as in (5), where N_{GOP} is the GOP size. In (6), $PCE(x, y, n)$ denotes the norm of concealment error of pixel (x, y) of frame n , where $f(x, y, n)$ is the pixel value of pixel (x, y) in frame n , assuming the zero-motion error concealment scheme [21] is adopted.

$$PRC(x, y, n) = \begin{cases} \sum_{(x', y', n+1) \text{ points to } (x, y, n)} PRC(x', y', n+1) & 1 \leq n < N_{GOP} \\ 1 & n = N_{GOP} \end{cases} \quad (5)$$

$$PCE(x, y, n) = |f(x, y, n) - f(x, y, n-1)|^2 \quad (6)$$

As depicted in Fig. 7, we then use the motion information to calculate the current frame's macroblock-level error-propagation by

$$EP_{MB}(m, n) = \sum_{(x, y) \in MB_m} LI(x + MV_{x, y} + MV_{y, n-1}) \quad (7)$$

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

$$EP_{pkt}^{(k)}(n) = \sum_{m=1}^{N_{MB}} EP_{MB}(m, n) \quad (8)$$

where k denotes the packet index of a frame, and N_{MB} denotes the number of macroblocks in the packet. The packet-level error propagation measure is taken as the estimation of the packet loss impact value of each packet.

Fig. 8 shows an example of the packet loss impact values for the QCIF Foreman sequence of 300 frames which is coded at 384 Kbps. In our experiments, a slice containing one row of macroblocks is encapsulated into one packet. Note, the loss estimation can be performed offline for pre-recorded video streaming applications, thereby resulting in no extra computational complexity while performing real-time transmission. Besides, the loss estimation is based on the assumption that there is only a single packet loss in a GOP. Since the prioritized ARQ determines the retransmission priority according to packets, relative importances, our experimental results show that the proposed loss estimation is reasonably accurate even when there are multiple lost packets in a GOP.

3.2. Content-adaptive prioritized retransmission

According to the packet-level loss estimation, we propose a content-adaptive prioritized retransmission scheme. Note that, the retransmitted packets will consume part of available bandwidth as well as cause extra transmission delay. In this work, we consider the application scenario that the retransmissions of lost packets from the media gateway may cause the network resource contention

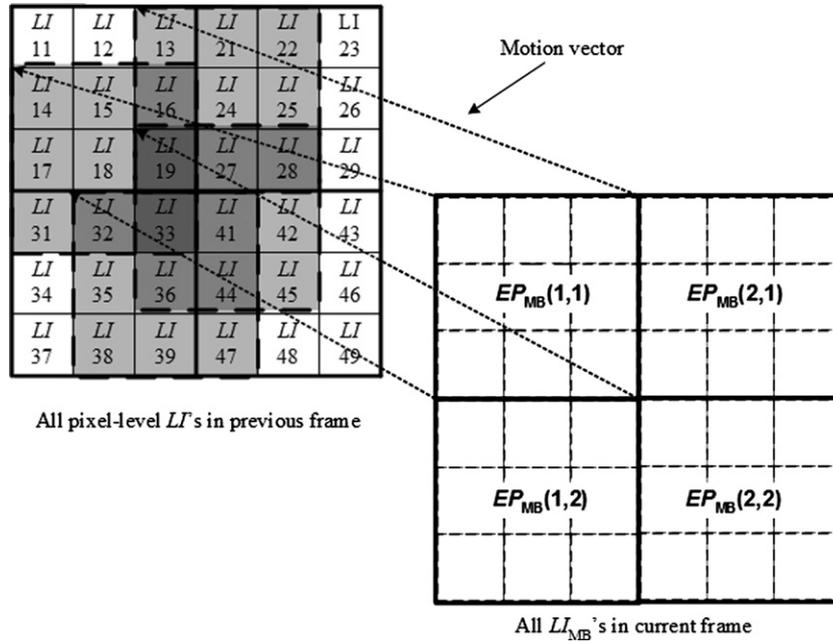


Fig. 7. Illustration of using motion vectors to map pixel-level loss impact values from the previous frame to obtain macroblock-level error propagation impact values in the current frame.

between the regular video packets and the protection information. Under this constrained scenario, if an important packet gets lost but cannot be resent due to the limited bandwidth, serious quality degradation may be introduced. One feasible solution to this problem is to drop out some “unimportant” regular packets so as to use the saved bandwidth to retransmit the “important” lost packets under a retransmission delay constraint. The level of importance of a packet is measured by its loss-impact value defined above.

Since the packet loss behavior of a communication channel is difficult to predict in advance, what “regular” packets the server is sending while receiving a retransmission request (i.e., a NAK packet) is also dependent on the arrival time of the request. In addition, the play-out deadline left for a retransmitted packet also varies dynamically, making it difficult to determine an optimal retransmission policy in real-time. We propose a low-complexity Greedy algorithm [30] for making the retransmission decision in real-time while still achieving satisfactory performance. In our method, the client will initially determine whether or not to request a retransmission for a lost packet according to its play-out deadline. While receiving a retransmission request for a lost packet, the media gateway decides whether or not to resend the lost packet by comparing the loss-impact of the lost packet with those of regular packets in the transmitter queue. In the proposed Greedy algorithm When a lost packet is resent, the queuing regular packet(s) with the least loss-impact rank(s) and a similar total size to that of the resent packet will be dropped out so as to regulate the output bit-rate to meet the bandwidth requirement.

The proposed method requires additional side information for recording the loss-impact rank of each video

packet within a GOP, which consumes extra bandwidth. For example, the maximum number of rank-orders of packet-level impact values is 270 for a QCIF video with a GOP size of 30, leading to 9 bits per packet to represent the full rank-orders of loss-impact values, respectively. The resultant overhead cost is about 0.63% for a QCIF video coded at 384 Kbps and 30 fps. The overhead cost can be further reduced by quantizing the rank-orders into few bits. The proposed content-based prioritized ARQ algorithm is summarized as follows:

Algorithm: Content-Based Prioritized ARQ

Client side:

if $(T_{cur} + \overline{RTT}_n + D_s) \geq T_d$ (pkt_n^{lost})/* the retransmitted packet cannot meet the play-out schedule */
do not request retransmission

else

request retransmission

Server side:

if the media gateway receives a retransmission request for pkt_n^{lost}

{

find pkt_m with the smallest loss-impact in the queuing regular packets with size(pkt_m) ≥ size(pkt_n^{lost}) under the T_d (pkt_n^{lost}) constraint

if pkt_m exists and the impact value of pkt_n^{lost} is greater than pkt_m
retransmit pkt_n^{lost} and drop out pkt_m

else

send the regular packets and ignore the retransmission request for pkt_n^{lost}

}

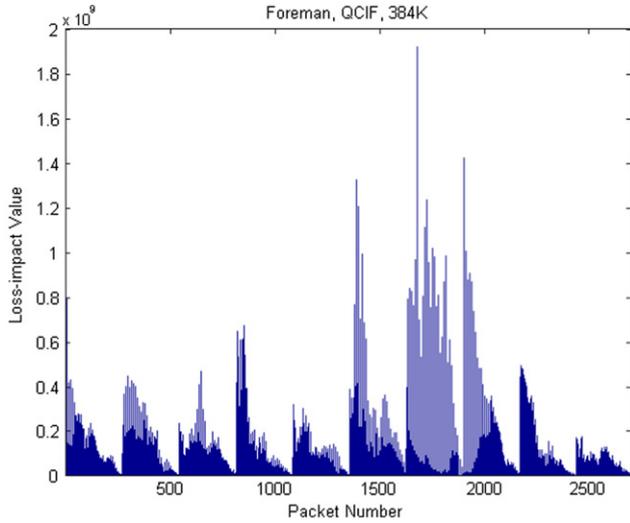


Fig. 8. Packet loss-impact values for the QCIF *Foreman* sequence coded at 384 Kbps.

where $\text{pkt}_n^{\text{lost}}$ stands for the n th lost packet and pkt_m represents the m th regular packet. T_{cur} represents the current time. D_s is a slack term. $T_d(\text{pkt}_n^{\text{lost}})$ denotes the play-out deadline for $\text{pkt}_n^{\text{lost}}$ and $\text{size}(\text{pkt}_n^{\text{lost}})$ is the packet's packet size. $\overline{\text{RTT}}_n$ denotes the average round-trip time estimated for $\text{pkt}_n^{\text{lost}}$ by (1) at the client terminal, which is updated every 0.1 s in our system.

4. Channel-aware MD transcoding

The proposed MDSQ transcoder is aimed at dealing with the handoff between two base-stations separated by a certain distance in wireless LANs. The transitions due to channel switching usually happen across the coverage boundary of the two base-stations, which may lead to poor channel conditions for both channels via the two base-stations. As a result, in the proposed method, estimating the relative qualities of the two channels is usually sufficient for the mode decision to choose either a better-condition channel or both channels with similar conditions.

4.1. Channel-aware mode switching

As illustrated in the state transition diagram shown in Fig. 9, the media gateway which is currently operated in the SD mode will, at the next stage, either switch to the MD mode which sends video data via multiple paths or still stay in the SD mode which sends the data via the same chan-

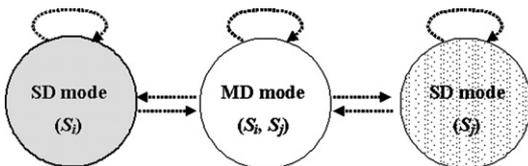


Fig. 9. State transition diagram of the proposed mode decision scheme.

nel. Assuming the media gateway is originally operated in the SD mode via S_i , in our method, when $\overline{\text{PLR}}^i$ exceeds $\overline{\text{PLR}}^j$ by more than a predetermined threshold PLR_{SD} , the media gateway will switch to the MD mode by transcoding the video data and send the two descriptions via S_i and S_j , respectively. If channel j is not significantly better than channel i in terms of PLR, our method will compare the RTTs of the two channels. If $\overline{\text{RTT}}^i$ is larger than $\overline{\text{RTT}}^j$ by $k_{\text{SD}} \times \overline{\text{RTT}}^i$, the media gateway will also switch to the MD mode; otherwise it will stay in the SD mode (via S_i).

If the media gateway is currently staying in the MD mode, it will either remain in the MD mode or switch to the SD mode by sending data through the channel in a relatively better channel condition at the next stage. In our method, $\overline{\text{PLR}}^i - \overline{\text{PLR}}^j > \text{PLR}_{\text{MD}}$ (where $\text{PLR}_{\text{MD}} > \text{PLR}_{\text{SD}}$) in the MD mode indicates the condition of channel i becomes much poorer such that the media gateway needs to switch to the SD mode via S_j . Otherwise, in the case that $\overline{\text{RTT}}^i - \overline{\text{RTT}}^j > k_{\text{MD}} \times \overline{\text{RTT}}^j$, the system will also choose to switch to the SD mode via S_j . Note that, the lower the values of PLR_{SD} , PLR_{MD} , k_{SD} , and k_{MD} , the higher the frequency of mode switches, thereby making it more sensitive to noise. Appropriate setting of these threshold values according to propagation environment and base-station layout can prevent excessive mode switches due to “ping-ponging” between two modes [32].

Algorithm: Channel-Aware Mode Decision for MD Transcoding

SD(S_i): the media gateway is operated in the SD mode via S_i

$\overline{\text{PLR}}^i$ and $\overline{\text{RTT}}^i$: Average packet loss rate and round-trip time of channel i

PLR_{SD} , PLR_{MD} , k_{SD} , and k_{MD} : Predetermined parameters for the round-trip time and packet loss rate in the SD and MD modes ($\text{PLR}_{\text{MD}} > \text{PLR}_{\text{SD}}$)

In the SD(S_i) mode:

if ($\overline{\text{PLR}}^i - \overline{\text{PLR}}^j > \text{PLR}_{\text{SD}}$)

Switch to the MD mode: transcode the video data into two descriptions, and send the two descriptions via two channels separately

else

{ if ($\overline{\text{RTT}}^i - \overline{\text{RTT}}^j > k_{\text{SD}} \times \overline{\text{RTT}}^i$)

Switch to the MD mode: transcode the video data into two descriptions, and send the two descriptions via two channels separately

else

Stay in the SD(S_i) mode: directly forward the video data through channel i (via S_i) without transcoding

}

In the MD mode:

if ($\overline{\text{PLR}}^i - \overline{\text{PLR}}^j > \text{PLR}_{\text{MD}}$)

Switch to the SD(S_j) mode: forward the video data through channel j (via S_j) without transcoding

```

else
{
  if ( $\overline{RTT}^i - \overline{RTT}^j > k_{MD} \times \overline{RTT}^j$ )
    Switch to the SD( $S_j$ ) mode: forward the video
    data through channel  $j$  (via  $S_j$ ) without transcoding
  else
    Stay in the MD mode: transcode the video data
    into two descriptions, and send the two descriptions via
    two channels separately
}

```

		Side-2 Index (i_2)							
		1	2	3	4	5	6	...	M
Side-1 Index (i_1)	1	1	3	5					
	2	2	6	8	10				
	3	4	7	11	12	14			
	4		9	13	16	17	19		
	5			15	18	21	23	25	
	6				20	22	26	.	
	⋮					24	.	(i_0)	.
M							.	N	

Fig. 11. Example of an index assignment matrix.

4.2. MDSQ transcoding

MDC schemes have been widely studied in the literature [13,14]. Because of its simplicity, we adopt in this work the MDSQ coding scheme [15] as an error resilience transcoding tool to demonstrate the power of channel-aware MDC-based transcoding. Other more sophisticated MDC schemes [22–27] can also be used in the proposed framework to further improve the performance, whereas the computational complexity of transcoding may also be increased.

Fig. 10 shows the MDSQ transcoder and decoder used in the media gateway and the client, respectively. We adopt a low-complexity compressed-domain MDSQ transcoder which is composed of three components: variable-length decoder (VLD), index mapper $a(\cdot)$, and variable-length coder (VLC). The transcoder can also be replaced with another higher performance transcoder, such as a close-loop pixel or DCT-domain transcoder [2] with a more sophisticated MDC scheme to improve the visual quality at the cost of significantly higher complexity [22–27]. In the index mapper of the transcoder, the central quantization index i_0 is mapped into two “side” quantization indices i_1 and i_2 . The two side quantization indices are treated as two descriptions and sent through two distinct channels. By the conclusion drawn in [15], the valid elements of index assignment matrices are designed to concentrate around the diagonal line of the matrix. Such an index assignment matrix is controlled by a redundancy parameter k , and

the width of diagonal lines of the matrix is set to $2 \times k + 1$. A larger k corresponds to smaller redundancy. That is, when k becomes larger, the central distortion D_0 (when receiving both descriptions) will be smaller, whereas the side distortion D_s (receiving only one description) will become larger, given the same bit rate. At the receiver side, the MDSQ decoder applies inverse index mapping $a^{-1}(\cdot)$ from two received side quantization indices i_1 and i_2 to find a central quantization index i_0 , and then performs inverse scalar quantization on i_0 to obtain the reconstruction value \tilde{X} of the source. When only one description (quantization index) is received by the decoder, there would be a set of possible central quantization indices. In this case, the mean value of cell intervals of all central indices is taken as the reconstructed value \tilde{X} to minimize the distortion from the original source X . Fig. 11 shows an example of an index assignment of MDSQ, which maps a central quantization index in the set $\{1, 2, \dots, N\}$ into a pair of side quantization indices in the set $\{1, 2, \dots, M\}$.

5. Experimental results

In our experiments, we first use the OPNET [31] network simulator to generate packet loss patterns to evaluate the performance of the proposed schemes, and then determine appropriate parameters, PLR_{SD} , PLR_{MD} , k_{SD} , and k_{MD} , for the proposed algorithms accordingly. We subsequently use the testbed system established in our laboratory to verify the proposed schemes again. In the experiments, three QCIF test sequences, *Foreman*, *Coastguard*, and *News* are encoded at 384 Kbps and 30 fps with a GOP size of 30 using a public-domain MPEG-4 coder [33]. A slice containing one row of macroblocks is encapsulated into one packet. The range of average packet length is 550–850 bytes for I-frame packets and 155–165 bytes for P-frame packets. We adopt the zero-motion replacement method [21] for concealing the lost video data.

5.1. Simulation of channel status detection mechanism

The parameters used for the OPNET network simulator are listed in Table 1. In the simulations, the wireless domain is configured at the “ad hoc” mode with the access point

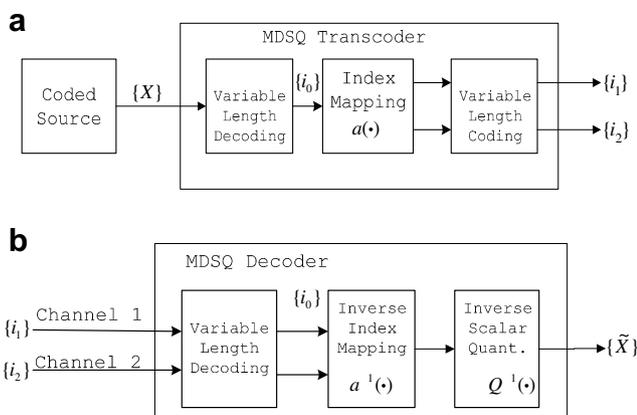


Fig. 10. Architecture of MDSQ codec: (a) the MDSQ transcoder adopted in the media gateway and (b) the MDSQ decoder used in the mobile client.

Table 1
Parameters used in the simulation

Parameters for wireless LAN	
Data rate	11 Mbps
Physical characteristics	Frequency hopping
Access point functionality	Disabled
Parameters for environment setup	
Distance between S_1 and S_2	300 m
Moving speed of mobile users	4 km/h
Parameters for channel status estimation	
Probing period (T_{probe})	100 ms
PLR_{SD}	0.2
PLR_{MD}	0.4
k_{SD}	0.6
k_{MD}	1.2

functionality disabled. The channel rate of wireless LAN is 11 Mbps with frequency-hopping physical links. The distance between the two base-stations, S_1 and S_2 , is assumed to be 100 m. The speed of the mobile user moving from S_1 to S_2 is set to be 4 km/h to simulate a pedestrian speed. Table 1 also lists the probing period and the four threshold values used in the experiments. The traces of packet loss rates and round-trip delays of the two channels via S_1 and S_2 , generated with the OPNET simulator are shown in Fig. 12, respectively. The channel selection result determined by the media gateway is also shown in Fig. 12. From the channel selection result, the path via S_1 is chosen first. During the period between time instances 46 and 56, both the two paths via S_1 and S_2 are respectively used to send the two descriptions of video obtained from the MDSQ transcoding. Finally, only the path via S_2 is used to send a single description after time instance 56.

5.2. Performance evaluation of MD transcoding in the proposed testbed

In order to construct a roaming environment in a laboratory environment, the two base-stations are physically separated by about 30 m and their transmitting power is reduced to 25%. We simulate a roam between stations S_1

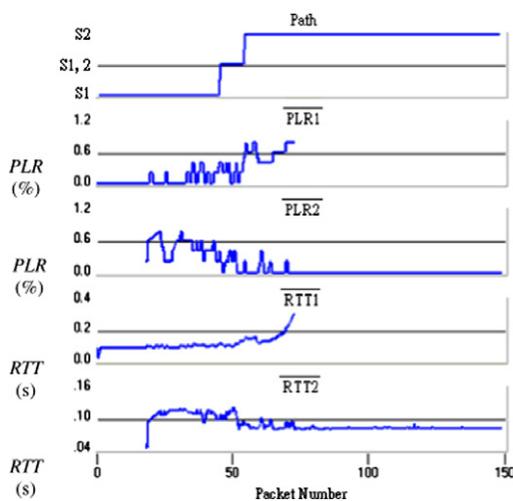


Fig. 12. Path selection result and \overline{PLR} , \overline{RTT} of the channels via S_1 and S_2 .

and S_2 with a scenario that a client moves from station S_1 toward station S_2 and then moves back to S_1 at a pedestrian speed. Fig. 13 shows the \overline{PLR} and \overline{RTT} traces of the two channels via S_1 and S_2 , respectively, and the channel selection result by the transcoder. At time instance 180 in Fig. 13, the mode decision mechanism detects that the \overline{PLR} and \overline{RTT} of the condition of the first path via S_1 are going worse and finds that the second path via S_2 is in good condition. It thus decides to choose the MD mode to deliver the data to the client through both paths. Then at time instance 210, because the first path's condition becomes even worse than the second path's, the video packets are sent through the second path (via S_2) entirely. The situation is similar during time instances 395–414 when the client happens to roam from S_2 to S_1 .

We can observe from the frame-by-frame channel utilization for the *Foreman* sequence shown in Fig. 14 that, in the MD mode during frames 98–110 and 225–232, the data rate is about 27% higher than that of the SD mode if no video quality loss (i.e., no drift) is introduced by the transcoder. Such excessive data rate will only appear in very short periods during channel transition. As shown in Table 2, the overall data rate of MD mode is higher than that of the SD mode by about 21–27% for the three test sequences. Since the data are transmitted via two separate channels, the excessive data rate will not introduce heavy burden on individual channels. Table 3 shows the run-time analysis of the transcoding of MD mode on an Intel Pentium-M 1.4-GHz PC with 1.24-GB RAM. The processing speed of the MDSQ transcoder is about 126.6–137.0 QCIF fps, which can meet the real-time processing requirement. Fig. 15 compares the frame-by-frame PSNR performances of the proposed method and the traditional roaming scheme in IEEE 802.11 WLANs for a situation with two handoffs happening in the fourth and eighth GOPs. Evidently, the proposed MD coding with multipath diversity can significantly mitigate the distortion caused by transient channel switching due to roaming. The average PSNR performance improvement on the corrupted frames using the proposed method is about 3.34–7.18 dB for the channel condition used in our experiments. Some reconstructed frames are illustrated in Fig. 16 for subjective quality comparison.

5.3. Performance evaluation of the content-based prioritized retransmission

To evaluate the performance of the proposed prioritized ARQ scheme, we use a two-state Markov model, which adopts a simplified Gilbert channel at the packet level [9] to generate test patterns for three packet loss rates (PLRs): 5%, 10%, and 15%. Ten different test patterns are generated for each packet loss rate. Such high packet loss rates usually lead to multiple lost packets within a GOP. Since the loss of picture header will destroy the whole frame, we set the loss impact of the first packet with the picture header of a frame to be the sum of all the impact values

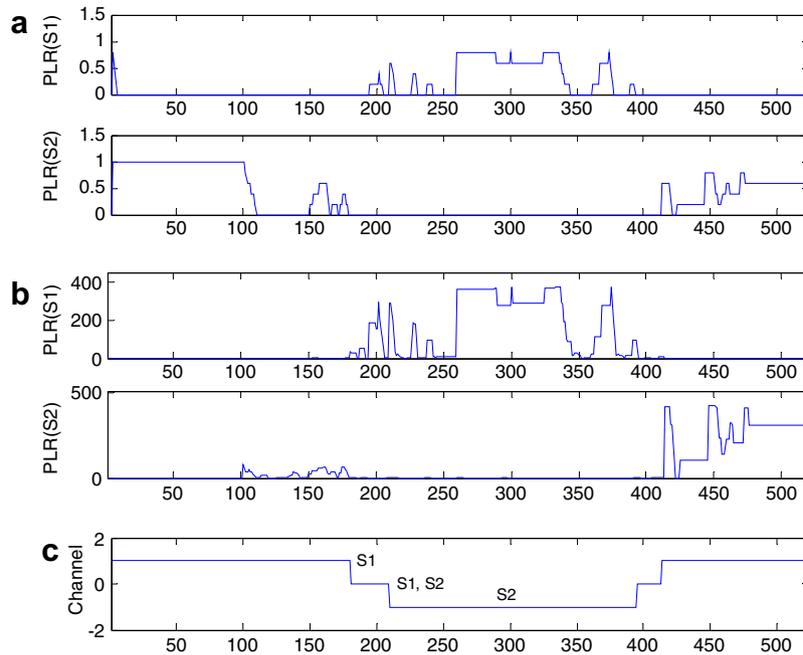


Fig. 13. Channel statistics collected from our streaming testbed: (a) the traces of PLR, (b) the traces of RTT for the two channels via the S_1 and S_2 stations, respectively, and (c) the channel selection result by the transcoder.

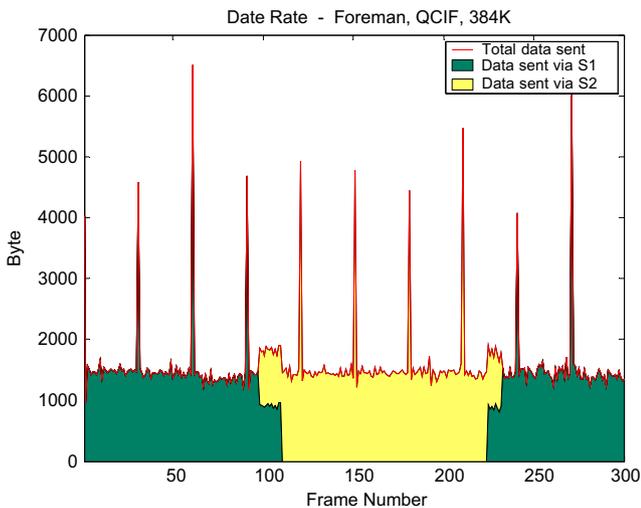


Fig. 14. Frame-by-frame channel utilization (*Foreman*, $N_{GOP} = 30$).

of the packets in the frame. Another alternative is to protect the picture header using the Header Extension Code (HEC) provided in MPEG-4, which inserts duplicate copies of important header information to other packets. We

compare the proposed prioritized ARQ scheme with the typical Early-Deadline-First (EDF) scheme [5], the FBS scheme [10,11], and that without retransmission protection. Note that, FBS is also kind of content adaptive error protection schemes since it protects a frame according to the frame's loss impact that is evaluated by counting the number of frames that are inter-coded with respect to the frame. Table 4 compares the average PSNR performances for the three test sequences, and Figs. 17–19 illustrate the corresponding frame-by-frame PSNR performances. The experimental results show that the proposed scheme achieves significant average PSNR improvement (up to 2.3 dB) over the FBS and EDF schemes, especially for packet losses that occur in high-motion frames.

Table 3

Run-time analysis of the MD transcoding

Sequence (300 frames)	Total time (s)	Average processing speed (fps)
<i>Foreman</i>	2.27	126.6
<i>Coastguard</i>	2.22	135.1
<i>News</i>	2.19	137.0

Table 2

Transcoding cost at the MD mode for the test sequences

Sequence QCIF 300 frames 384 Kbps	SD mode (single stream)		MD mode (two streams)			
	Average PSNR (dB)	Data-size (byte)	Average PSNR (dB)	PSNR cost (dB)	Data-size (byte)	Data-rate cost (ratio)
<i>Foreman</i>	35.83	467,174	35.83	-0.0	592,867	27%
<i>Coastguard</i>	34.07	466,286	34.07	-0.0	579,274	24%
<i>News</i>	40.37	479,057	40.37	-0.0	577,603	21%

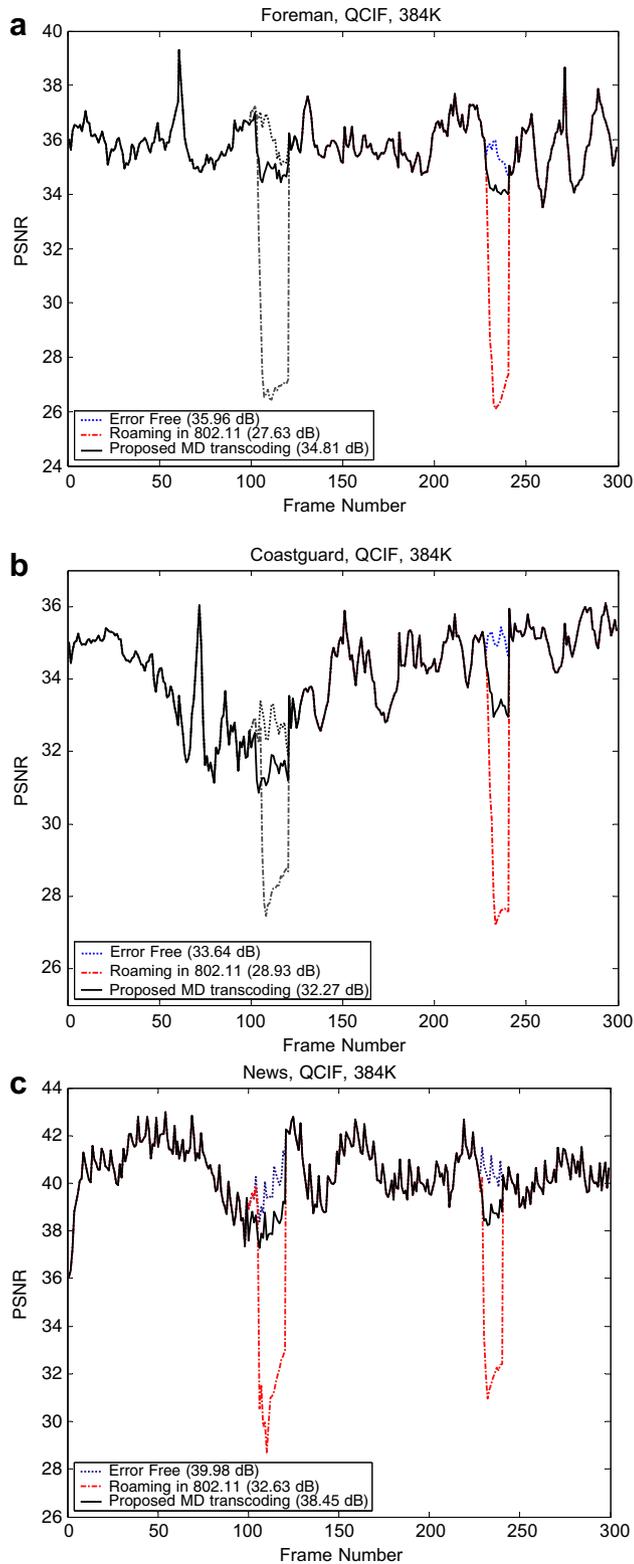


Fig. 15. Frame-by-frame PSNR performance comparison for roaming with and without the proposed MD transcoding: (a) *Foreman*, (b) *Coastguard*, and (c) *News*.

Since the proposed scheme tends to retransmit lost packets of higher loss-impact ranks by dropping out regular packets of less importance, it achieves better video quality

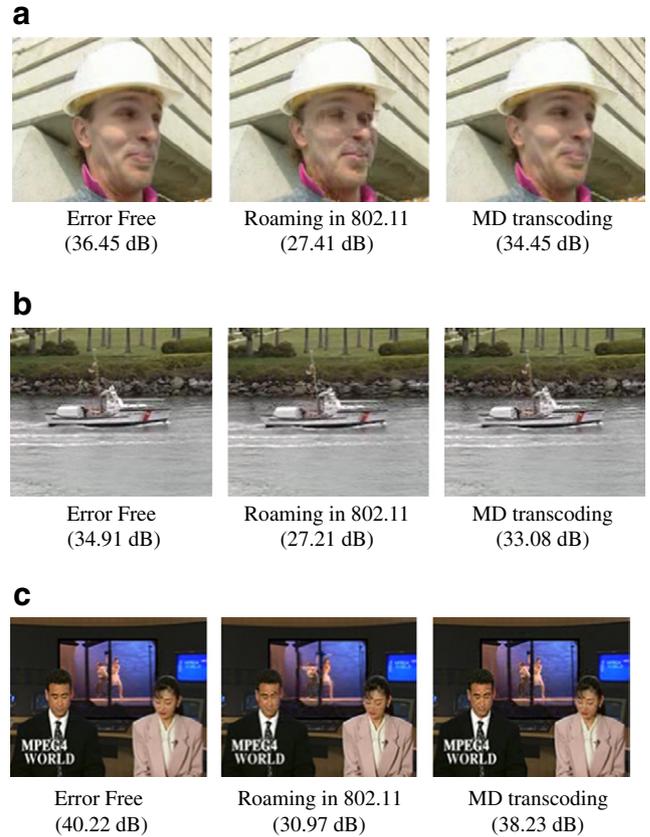


Fig. 16. Subjective quality comparison for roaming with and without the proposed MD transcoding: (a) *Foreman* (frame #106), (b) *Coastguard* (frame #233), and (c) *News* (frame #232).

most of the time. On the other hand, the EDF scheme only considers the play-out deadline without taking into account the priorities of video packets, packets of higher importance may be dropped out if their play-out deadlines are earlier than the dropped one's. The loss impact estimations scheme of FBS that simply counts the number of frames inter-coded with respect to the frame may not reflect the actual importance of the frame very well. Moreover, with the proposed method, if the client does not successfully receive a retransmitted video packet, the dropped packet may still have a chance to be retransmitted again as long as its play-out deadline can be met. In some rare cases (see Figs. 17–19), when the actual error propagation caused by the regular packets that are replaced during packet retransmission are higher than that of the resent packets, the proposed method, EDF, and FBS may degrade the visual quality. The reason for such quality degradation is that it is very difficult to find an optimal schedule of sending regular and retransmitted packets considering both their error propagations and play-out deadlines without performing a prohibitively high complexity exhaustively search. Fig. 20 shows some reconstructed frames for evaluating the subjective visual quality of the proposed prioritized ARQ scheme.

Table 4

Averaged PSNR comparison of different schemes for three test sequences under three different channel conditions (10 test patterns)

Sequence	Retransmission method	Average PSNR (dB) for different PLRs		
		5%	10%	15%
<i>Foreman</i>	Error free		36.27	
	Proposed	32.25	29.07	25.83
	FBS drop	31.32	27.68	24.49
	EDF drop	28.38	25.37	23.18
	No ARQ	28.09	25.19	23.00
<i>Coastguard</i>	Error free		34.04	
	Proposed	31.75	29.28	27.20
	FBS Drop	30.98	28.38	26.04
	EDF Drop	29.31	26.18	24.41
	No ARQ	28.66	25.78	24.35
<i>News</i>	Error free		40.42	
	Proposed	36.95	31.61	30.48
	FBS Drop	36.34	31.15	29.81
	EDF Drop	34.95	30.41	28.92
	No ARQ	34.12	29.68	28.66

6. Concluding remarks and future work

We proposed an adaptive error protection scheme in the media gateway to mitigate the packet loss caused by client roaming or other transient channel failures. Our scheme divides the error control operation into the SD and MD modes according to the RTTs and PLRs reported from the base-stations. As a result, our scheme deploys content-based prioritized RQ and channel-aware MD transcoding for error protection in the SD and MD modes, respectively. The proposed prioritized ARQ scheme determines the retransmission schedule of a lost packet based on the loss estimation of the lost packet which is obtained in an off-line coding process and prestored in the streaming server. The proposed channel-aware MD transcoding scheme will try to find an alternative channel with an acceptable channel condition to be used simultaneously for path diversity when detecting a mobile client roaming from one station to another. Experimental results demonstrated the proposed scheme can handle transient channel failures very well, while still keeping low extra computational complexity and overhead cost. We have established a testbed for real-time multipath streaming to verify the performance of the proposed schemes. The experimental results show that the proposed method achieves significant visual quality improvement over traditional methods. The proposed methods can also be extended to cellular applications, such as 3G systems, to achieve error robust handoffs among base-stations for mobile users in wireless Wide Area Networks (WANs).

In our current method, the prioritized ARQ is used only in the SD mode since MD transcoding will change the loss impact values of lost packets. While MD transcoding is activated, we can choose to either disable ARQ or resort to traditional ARQ. Although the loss estimation is done

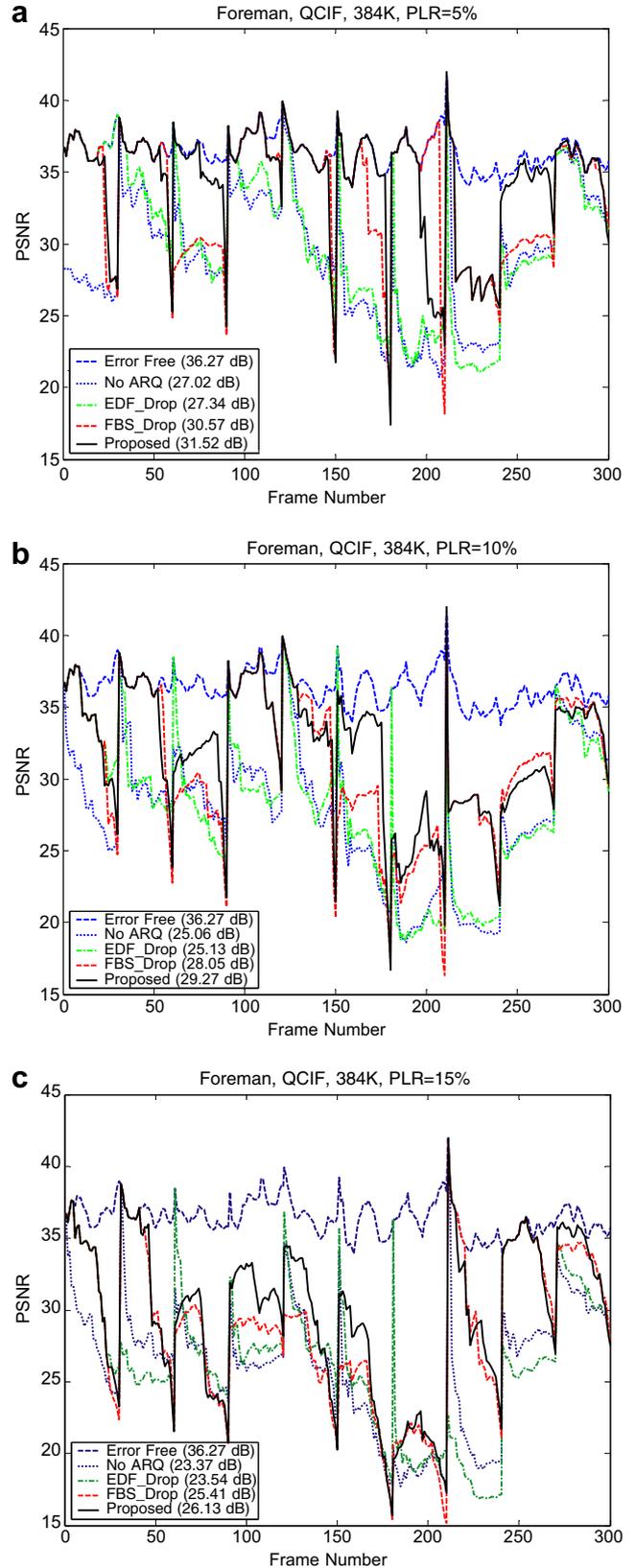


Fig. 17. Frame-by-frame PSNR performance comparison of the QCIF *Foreman* sequence coded at 384 Kbps under three different channel conditions.

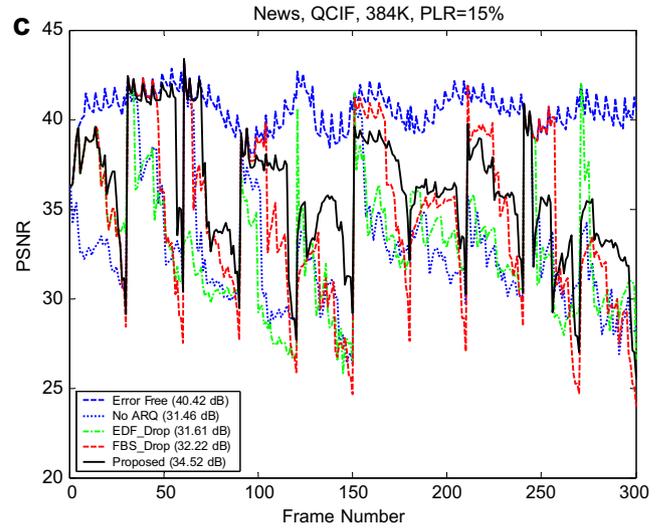
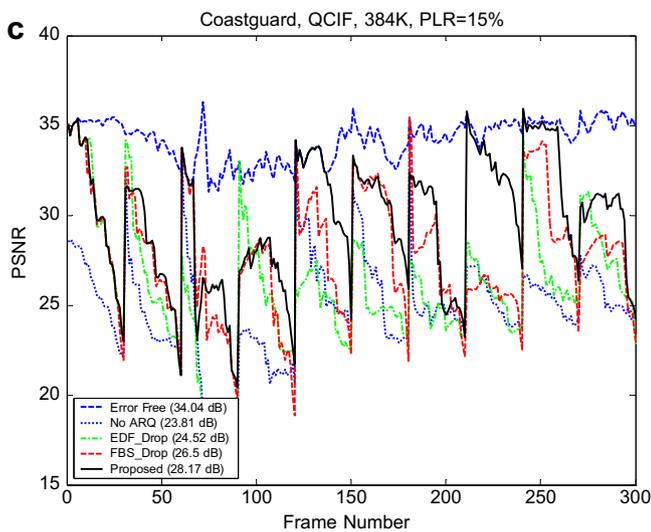
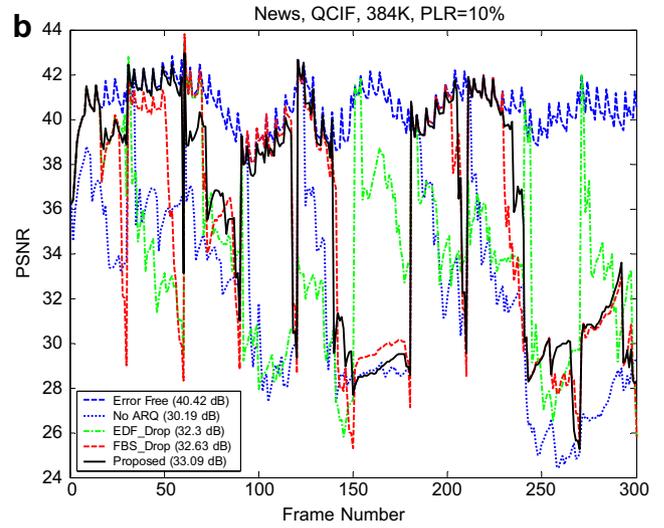
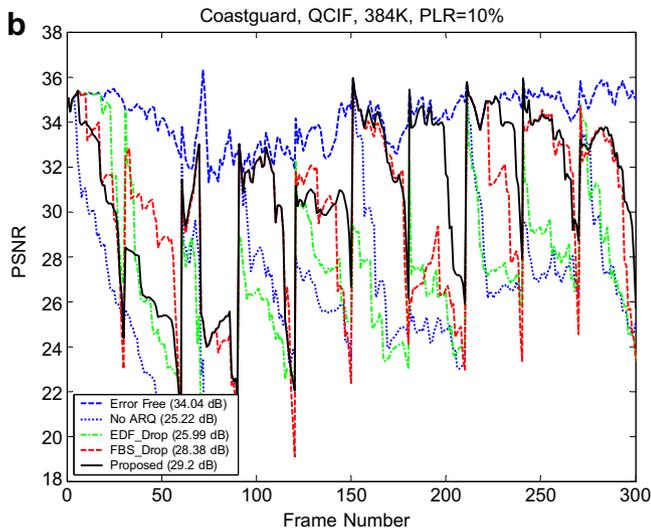
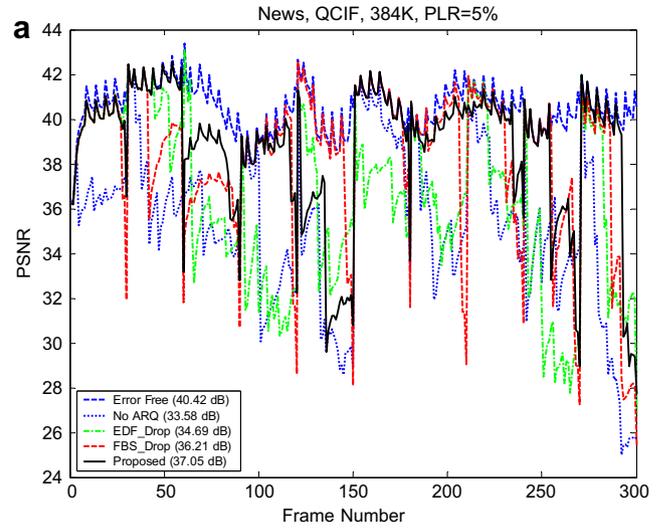
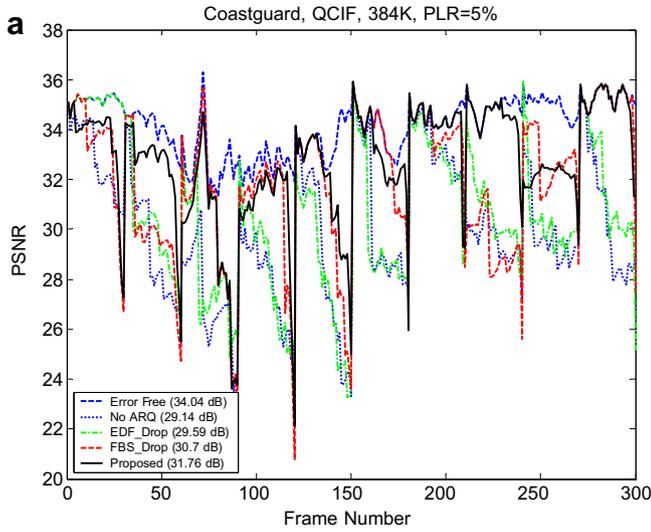


Fig. 18. Frame-by-frame PSNR performance comparison of the QCIF *Coastguard* sequence coded at 384 Kbps under three different channel conditions.

Fig. 19. Frame-by-frame PSNR performance comparison of the *News* sequence coded at 384 Kbps under three different channel conditions.

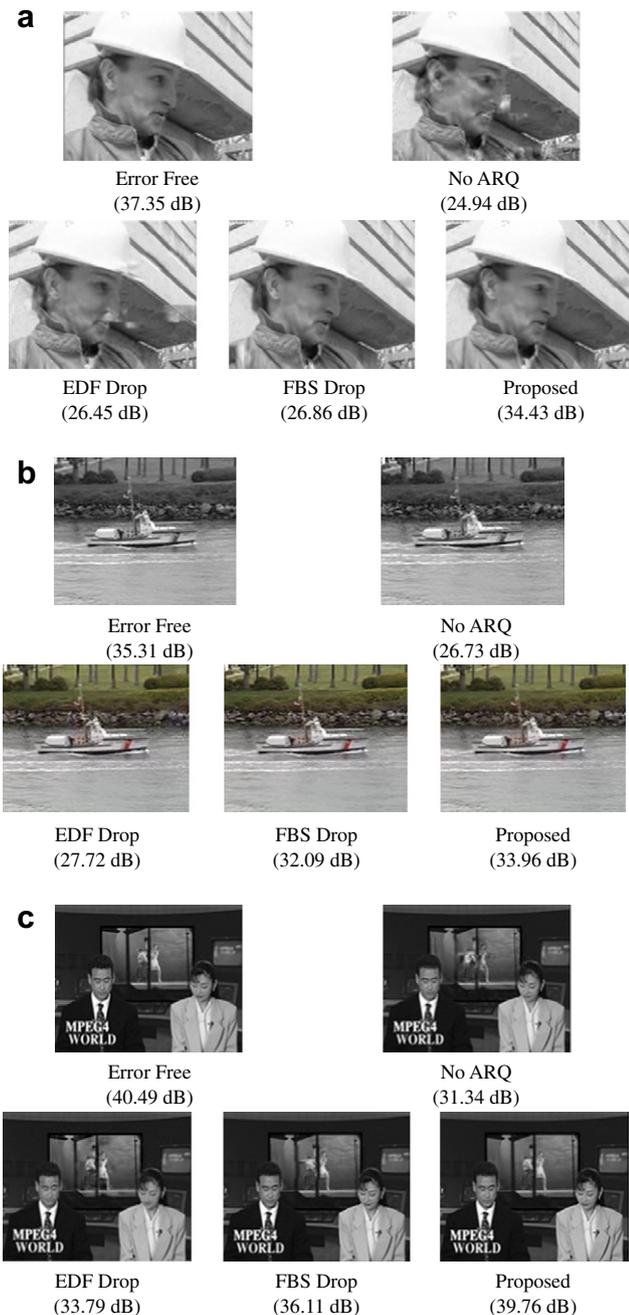


Fig. 20. Subjective quality comparison at 384 Kbps and PLR = 10%: (a) *Foreman* (frame #165), (b) *Coastguard* (frame #231), and (c) *News* (frame #232).

in the SD mode, the estimation still somehow reflects the relative importance of lost packets even in the MD mode. We are still investigating how to reasonably apply the prioritized ARQ in the MD mode to further enhance error robustness.

Acknowledgment

This work was supported in part by Ministry of Economic Affairs (MOEA), Taiwan, ROC, under project Grant 96-EC-17-A-02-S1-032.

References

- [1] Y. Wang, Q.-F. Zhu, Error control and concealment for video communication: a review, *Proc. IEEE* 86 (1998) 974–997.
- [2] J. Xin, C.-W. Lin, M.-T. Sun, Digital video transcoding, *Proc. IEEE* 93 (1) (2005) 84–97.
- [3] H.-J. Chiou, Y.-R. Lee, C.-W. Lin, Content-aware error resilient transcoding using prioritized intra-refresh for video streaming, *J. Visual Commun. Image Represent.* 16 (2005) 563–588.
- [4] IEEE P802.11, Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY), 1997.
- [5] T. Hasegawa, T. Kato, K. Suzuki, A video retrieval protocol with video data prefetch and packet retransmission considering play-out deadline, in: *Proc. IEEE Int. Conf. Network Protocols*, pp. 32–39, 1996.
- [6] S.H. Kang, A. Zakhor, Packet scheduling algorithm for wireless video streaming, in: *Proc. Workshop Packet Video*, 2002.
- [7] K. Stuhlmüller, N. Farber, M. Link, B. Girod, Analysis of video transmission over lossy channels, *IEEE J. Selected Areas Commun.* 18 (2000) 1012–1032.
- [8] W. Tan, A. Zakhor, Packet classification schemes for streaming MPEG video over delay and loss differentiated networks, in: *Proc. Packet Video Workshop*, 2001.
- [9] S. Aramvith, C.-W. Lin, S. Roy, M.-T. Sun, Wireless video transport using conditional retransmission and low-delay interleaving, *IEEE Trans. Circuits Syst. Video Technol.* 12 (2002) 558–565.
- [10] M.-H. Lu, P. Steenkiste, T. Chen, Video streaming over 802.11 WLAN with content-aware adaptive retry, in: *Proc. IEEE Int. Conf. Multimedia and Expo*, July 2005, Amsterdam, The Netherlands.
- [11] M.-H. Lu, P. Steenkiste, T. Chen, A time-based adaptive retry strategy for video streaming in 802.11 WLANs, *Wireless Commun. Mobile Computing, Special Issue on Video Communications for 4G Wireless Systems*, 7 (2007) 187–203.
- [12] M. Zorzi, R.R. Rao, L.B. Milstein, ARQ error control for fading mobile radio channels, *IEEE Trans. Vehicular Technol.* 46 (1997) 445–455.
- [13] V. K. Goyal, Multiple description coding: compression meets the network, *IEEE Signal Process. Mag.* 18 (2001) 74–93.
- [14] Y. Wang, A. Reibman, S. Lin, Multiple description coding for video delivery, *Proc. IEEE* 93 (2005) 57–70.
- [15] V.A. Vaishampayan, Design of multiple description scalar quantizers, *IEEE Trans. Inf. Theory* 39 (1993) 821–834.
- [16] J. Apostolopoulos, Reliable video communication over lossy packet networks using multiple state encoding and path diversity, in: *Proc. SPIE Conf. Visual Commun. Image Processing*, pp. 392–409, 2001.
- [17] J. Chakaresk, E. Setton, Y. Liang, B. Girod, Video streaming with diversity, *Proc. IEEE Int. Conf. Multimedia & Expo 1* (2003) 9–12.
- [18] N. Gogate, D.M. Chung, S.S. Panwar, Y. Wang, Supporting video/image applications in a mobile multihop radio environment using route diversity and multiple description coding, *IEEE Trans. Circuits Syst. Video Technol.* 12 (2002) 777–792.
- [19] A. Miu, J.G. Apostolopoulou, W.-T. Tan, M. Trott, Low-latency wireless video over 802.11 networks using path diversity, *Proc. IEEE Int. Conf. Multimedia & Expo* (2003) 441–444.
- [20] S. Mao, S. Lin, S.S. Panwar, Y. Wang, E. Celebi, Video transport over ad hoc networks: multistream coding with multipath transport, *IEEE J. Selected Areas Commun.* 21 (2003) 1721–1737.
- [21] W.M. Lam, A.R. Reibman, B. Liu, Recovery of lost or erroneously received motion vectors, *Proc. IEEE Int. Conf. Acoustic, Speech, Signal Process.* (1993) V417–V420.
- [22] V.K. Goyal, J. Kovacevic, Optimal multiple description transform coding of gaussian vectors, *Proc. IEEE Data Compression Conf.* (1998) 388–397.
- [23] Y. Wang, M.T. Orchard, V.A. Vaishampayan, A.R. Reibman, Multiple description coding using pairwise correlating transforms, *IEEE Trans. Image Process.* 10 (2001) 351–366.

- [24] A.R. Reibman, H. Jafarkhani, Y. Wang, M.T. Orchard, R. Puri, Multiple description coding for video using motion compensated prediction, *IEEE Trans. Circuits Syst. Video Technol.* 12 (2002) 193–204.
- [25] N.V. Boulgouris, K.E. Zachariadis, A.N. Leontaris, M.G. Strintzis, Drift-free multiple description coding of video, *Proc. IEEE Workshop Multimedia Signal Process.* (2001) 105–110.
- [26] Y. Wang, S. Lin, Error resilient video coding using multiple description motion compensation, *IEEE Trans. Circuits Syst. Video Technol.* 12 (2002) 438–453.
- [27] C.-S. Kim, R.-C. Kim, S.-U. Lee, Robust transmission of video sequence using double-vector motion compensation, *IEEE Trans. Circuits Syst. Video Technol.* 11 (2001) 1011–1020.
- [28] K. Tan, Q. Zhang, W. Zhu, An end-to-end rate control protocol for multimedia streaming in wired-cum-wireless environments, in: *Proc. IEEE Int. Symp. Circuits Syst.*, pp. 836–839, 2003.
- [29] S. Cen, P.C. Cosman, G.M. Voelker, End-to-end differentiation of congestion and wireless losses, *IEEE/ACM Trans. Networking* 11 (2003) 703–717.
- [30] T.H. Cormon, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms*, second ed., MIT Press, 2001.
- [31] OPNET, <http://www.opnet.com>.
- [32] G.L. Stuber, *Principles of Mobile Communication*, second ed., Kluwer Academic Publishers, 2001.
- [33] ISO/IEC 14496 (MPEG-4) Video Reference Software, Version: Microsoft-FDAM1-2.3-001213.