Available online at www.sciencedirect.com
ScienceDirect
Signal Processing: *Image Communication* 22 (2007) 403–420
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

Error-resilient video streaming over wireless networks using combined scalable coding and multiple-description coding

Chih-Ming Chen[a,b], Chien-Min Chen[a], Chia-Wen Lin[a], Yung-Chang Chen[a,*]

[a]*Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan*
[b]*Chunghwa Telecom Co., Ltd., Telecommunication Laboratories, Taoyuan 326, Taiwan*

Received 8 November 2005; received in revised form 22 January 2007; accepted 23 January 2007

Abstract

In this paper, we propose a novel error-resilient coding technique, namely, multiple-description fine granularity scalability (MD–FGS) coding, to enhance error resiliency while retaining good temporal prediction efficiency for video transport over error-prone networks, especially for wireless channels. The proposed MD–FGS coding scheme is based on a combination of FGS and modified MD scalar quantization (MDSQ), named Base-MDSQ, in which partial prediction is utilized to control drifting error. We also propose an adaptive soft handoff algorithm to effectively mitigate video quality degradation caused by transient channel switching when a mobile client roams in wireless local area networks (WLANs). The handoff algorithm utilizes a channel estimator to decide which channels should be used for delivering video. Experimental results show that the proposed method effectively mitigates error propagation due to packet loss while maintaining good coding efficiency.

Keywords: Video streaming; Error-resilient coding; Transcoding; Scalable coding; Path diversity; Multiple-description coding; Channel handoff

1. Introduction

In video streaming, data packets may be lost or delayed in unreliable networks, such as congested networks and wireless networks. Delay constraint is one of the most important requirements in real-time applications. A video packet arriving later than the presentation time will become useless for the client. Besides delay constraint, video streaming over networks poses several other challenging problems, including bandwidth limitation and fluctuation, packet loss, etc. The first two problems can be partially solved by improving coding efficiency, deploying scalable codecs, and extending the receiver buffer. The problem of packet loss is, however, still difficult to deal with. MPEG-4 fine granularity scalability (FGS) [24] is a scalable coding technique which adopts bit-plane coding in the enhancement-layer to efficiently address the problem of channel bandwidth fluctuations. However, the coding efficiency of MPEG-4 FGS coder is typically significantly lower than a non-scalable

*Corresponding author. Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan.
Tel.: +886 3 5731153; fax: +886 3 5715971.
E-mail addresses: cwlin@ee.nthu.edu.tw (C.-W. Lin), ycchen@ee.nthu.edu.tw (Y.-C. Chen).

0923-5965/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2007.01.002

coder because its motion compensated prediction only utilizes the lowest quality base-layer. Using high-quality reference frames in the enhancement-layer can improve coding efficiency, but will result in prediction mismatch should part of video data be lost, which usually leads to serious visual quality degradation. The design of a scalable coder thus always has a trade-off between coding efficiency and drifting error. To improve the coding efficiency of FGS while controlling the prediction mismatch, leaky prediction methods that partially employs enhancement-layer information in the prediction loop were proposed in [12,13].

Fig. 1 illustrates an exemplary three-tier streaming system using a wireless local area network (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 with a transcoder inside, and a number of mobile client terminals. The data flow with such a streaming system 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.

Video transport over wireless networks may suffer from signal fading, noise interference, network congestion, and handoff, which will usually cause bit 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. Such packet loss not only affects the quality of current frame, but also leads to error propagation to subsequent frames because of the motion compensated prediction used in current video coding schemes [31].

Besides the above problems, video streaming with roaming across multiple base stations in wireless LANs further 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 will take place for completing a handoff. At first, a roaming mobile client decides whether or not to handoff between two APs based on factors such as signal strength and the number of missed beacons. The next action is to decide where to roam, and then initiate a handoff process. At last, the client must resume the existing application session. After the long handoff duration, it performs a nomadic roaming that may cause severe data loss that hurts

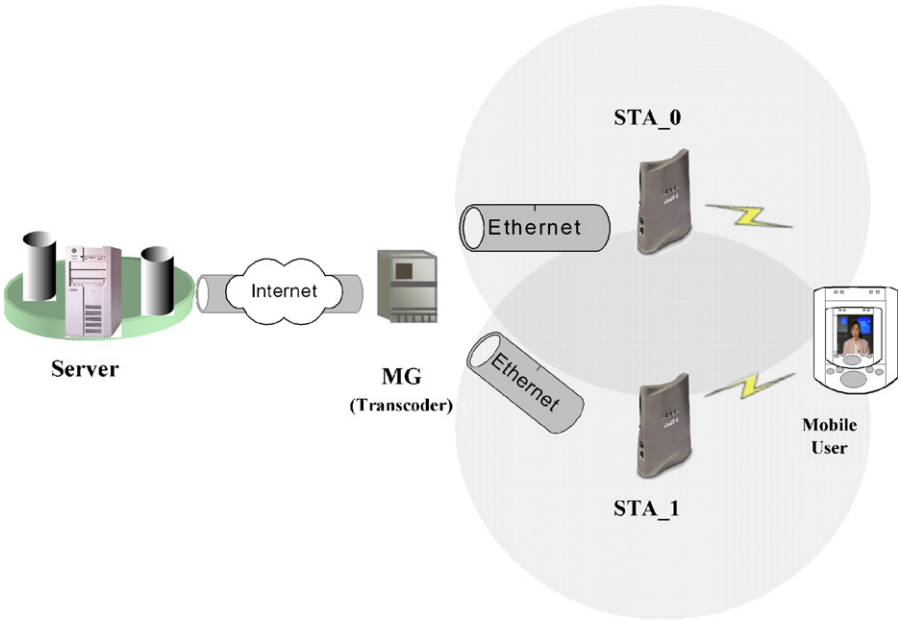

Fig. 1. Illustration of a three-tier video streaming with two diverse wireless LAN links between the media gateway and the mobile terminals.

the receiving video quality seriously. Fig. 3 illustrates the packet loss rate (PLR) and the visual quality degradation due to a channel handoff caused by a roam of the mobile client from STA_0 to STA_1 under the configuration shown in Fig. 1. The result shows that the channel handoff can lead to serious transient packet losses and visual quality degradation. The experimental details are provided in Section 4.

To handle the problem of packet loss while streaming a compressed video over error-prone networks, the most popular approach is to add a certain amount of redundancy to protect video data. Commonly used error-resilient source coding and transcoding tools [31,35] include data partitioning, synchronization marker, Reversible Variable Length Codes (RVLC) [31], Error Resilient Entropy Coding (EREC) [25], Adaptive Intra Refresh (AIR) [8], error tracking [23], Multiple-Description Coding (MDC) [31,10,33], etc. MDC is a kind of joint source and channel coder. The objective of MDC is to increase the reliability of data transmission under channel failures, by employing the diversity of channels. Fig. 2 shows a scenario of MDC with two descriptions sent over two channels. MDC intentionally inserts redundancy to make the bitstream more resilient to transmission error. Because MDC with channel diversity can usually significantly increase the possibility of receiving correct data from at least one channel, the effect of channel failures is mitigated. Several MDC schemes have been proposed recently to enhance error resilience of coded video. In [27], a video coder based on multiple-description scalar quantization (MDSQ) with two independent prediction loops was proposed. MD video coders based on pairwise correlating transform were presented in [11,32,22]. MDC is usually associated with drift when the signal used for prediction in an encoder is unavailable to a decoder due to loss of descriptions, which can lead to serious error propagation. The drift can be effectively mitigated using multiple predictors as proposed in [3], or using only a single predictor [22]. A drift-free MD video coder based on wavelet transform was proposed in [4]. The methods proposed in [29,16] illustrate how to select appropriate parameters in MDC to reach a reasonable trade-off between prediction efficiency and drifting error. In [28], method of combining MDSQ with motion compensated temporal filtering was proposed to enhance error resilience by using an optimal channel-aware rate allocation to minimize the expected distortion. Combining scalar coding with MDC to offer rate scalability and error resilience can be found in [1,2,7,30]. In [1], based on interframe wavelet scalable video coding, a scalable MDC scheme was proposed to decompose a video into a flexible number of descriptions with an adaptable bit-rate for each individual description and adaptable redundancy level of each description. The flexibility of the scalable MDC makes it very suited to peer-to-peer streaming applications [2]. In order to enhance error resilience, multiple-path transport (MPT) [5,6] is usually adopted for MDC [6], which can mitigate congestion in a network as well as increase overall network utilization. Several research works integrating MPT with MDC have been proposed for video communication over the Internet [3,28,2] and wireless networks [19,9,18]. A comprehensive survey about various MDC methods and systems integrating MPT and MDC can be found in [33].

In this paper, we propose an error-resilient MD–FGS coding/transcoding technique to enhance the error robustness against packet loss in WLANs. The main contribution of the proposed method is twofold. First, we present in the MD–FGS coder a novel scalable MDC method that combines MPEG-4 FGS with MDC. Compared to other prior ones on scalable MDC [1,2] or leaky prediction [12,13],

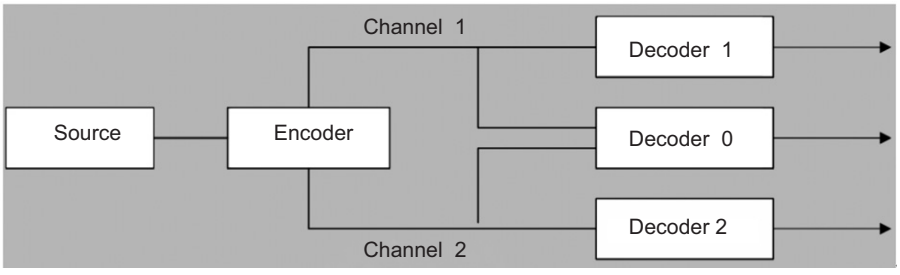

Fig. 2. A scenario of MDC with two descriptions sent on two channels.

the key novelty of our proposed MD–FGS method is a nice combination of scalable MDC with leaky prediction to achieve strong error robustness against packet loss while still keeping good coding efficiency. Second, we propose an adaptive soft handoff algorithm which makes use of path diversity to effectively mitigate video quality degradation caused by transient channel switching when a mobile client roams in WLANs (Fig. 3).

The rest of this paper is organized as follows. Section 2 describes the proposed Base-MDSQ coding scheme and its integration with two-layer partial prediction to reduce drift. In this section, a novel MD–FGS coder is also proposed to improve the error resilience while maintaining good coding efficiency. In Section 3, a robust roaming architecture with the proposed MD–FGS transcoder for video streaming across WLANs is presented. Section 4 reports experimental results to show the performance of our proposed schemes. Finally, the conclusion is given in Section 5.

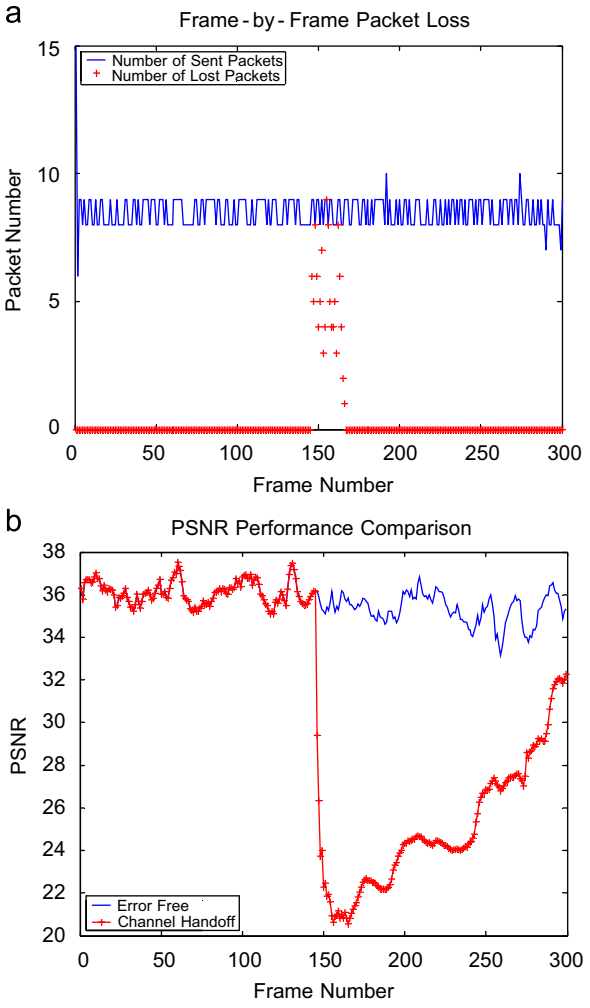

Fig. 3. Effect of handoff of video across two access points in a wireless LAN: (a) packet loss due to handoff; (b) PSNR quality degradation during handoff.

2. Base-MDSQ with partial prediction

In this section, the concept of MDSQ [27] is first briefly introduced. We then present the proposed Base-MDSQ coder which is an MDSQ coder with a newly designed quantizer index assignment to improve error resilience when parts of the video data are lost. Subsequently, Base-MDSQ with partial prediction is proposed to further mitigate drifting error without significantly reducing coding efficiency. Finally, the proposed MD–FGS codec is elaborated.

2.1. The MDSQ coding scheme

MDSQ is composed of two steps: normal scalar quantization $Q(\cdot)$ followed by an index mapping $a(\cdot)$. In the first step, it quantizes the source X and produces a "central" quantization index i_0. In the second step, the central quantization index i_0 is mapped into two "side" quantization indices i_1 and i_2. The side quantization indices are treated as two descriptions. Fig. 4 shows the architecture of MDSQ codec. By the conclusion of [27], the valid entries of the index assignment matrix are designed to concentrate around the principal 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 $2 \times k + 1$, in which larger k corresponds to smaller redundancy. That is, as k increases, the central distortion D_0 (when receiving both descriptions) will decrease, whereas the side distortion D_s (receiving only one description) will increase, given the same bit-rate. An example of a quantizer index assignment for MDSQ is depicted in Fig. 5, which maps a central quantization index in the set $\{1, 2, \ldots, N\}$ into a pair of side quantization indices in the set $\{1, 2, \ldots, M\}$.

2.2. The proposed Base-MDSQ coder

Base-MDSQ is a modified MDSQ coding scheme with a newly designed index assignment so that an encoder can extract the common part of all descriptions, which is referred to as the "base-part" of the descriptions, and then use it for predictive coding.

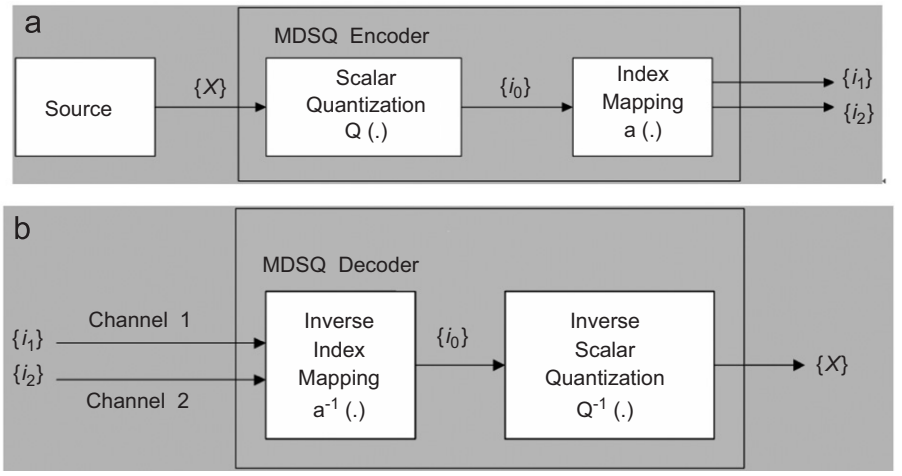

Fig. 4. Architecture of MDSQ codec: (a) encoder; (b) decoder.

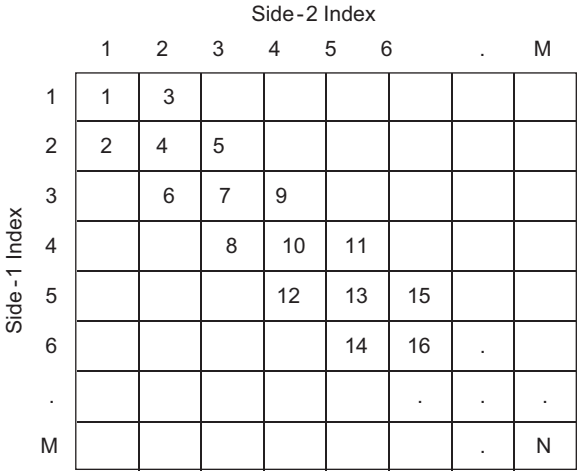

Fig. 5. An example of index assignment matrix for MDSQ.

In this way, if at least one description is received by a decoder, the decoder can retrieve exactly the same base-part from any of the received description(s), so that there is no mismatch between the predictions of the encoder and decoder. Using the term "base-part" is to avoid confusion with the "base-layer" of layered coding. Fig. 6 illustrates an example of index assignment for the proposed Base-MDSQ. In this example, we group all the central indices to form a number of "base-part" groups as bounded by a rounded rectangle, each corresponding to a specific base-part reconstruction value. At the decoder, when both descriptions are received, the central index can be decoded and the central reconstruction of the source can be retrieved accordingly. Should any of the two descriptions be received, the decoder can still reconstruct the base-part of the source. The difference between the central reconstruction and the "base-part" reconstruction is defined as the "enhancement-part" of the Base-MDSQ decoded data. Using only the base-part for prediction in the encoder and decoder can achieve better drift control since the possibility of receiving at least one description at the decoder would be relatively high.

In the following, we shall show how to obtain the optimal reconstructed value of a base-part given the probability distribution of the source values. Assume that the scalar quantizer at the encoder partitions the support of the source values into N central indices, and the partition thresholds $\bar{t} = (t_0, t_1, \ldots, t_N)$, where $t_0 \leqslant t_1 \leqslant \cdots \leqslant t_N$. Suppose the probability distribution function of the source is $p(x)$. The j-th base-part group B_j contains a set of central index values $\{i_0^{j,1}, i_0^{j,2}, \ldots, i_0^{j,bn_0}\}$ corresponding to a set of side-1 index values $\{i_1^{j,1}, i_1^{j,2}, \ldots, i_1^{j,bn_1}\}$ together with a set of side-2 index values $\{i_2^{j,1}, i_2^{j,2}, \ldots, i_2^{j,bn_2}\}$, where bn_0 is smaller than or equal to $bn_1 \times bn_2$. The reconstructed value of the base-part $\tilde{X}_{B_j}$ is calculated as the mean value of cell intervals of all the central index values $\{i_0^{j,1}, i_0^{j,2}, \ldots, i_0^{j,bn_0}\}$ as follows:

$$\tilde{X}_{B_j} = \sum_{i_0 \in \{i_0^{j,1}, \ldots, i_0^{j,bn_0}\}} \int_{t_{i_0-1}}^{t_{i_0}} xp(x)\,dx, \tag{1}$$

where a possible pair of side indices (i_1, i_2) is mapped from a central index i_0. We can divide the reconstruction procedure of base-part from the central and side indices into three mappings. The first mapping $b_0 : \{1, 2, \ldots, N\} \to \Re$ maps

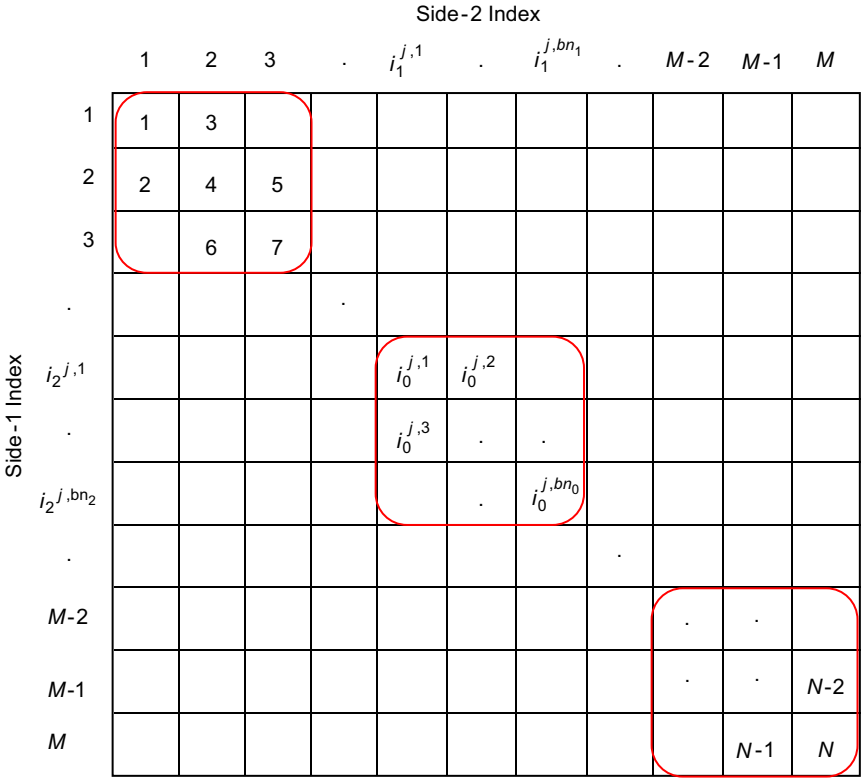

Fig. 6. An example of the modified index assignment for Base-MDSQ.

a central index to the reconstructed value of base-part, whereas the second $b_1 : \{1, 2, \ldots, M\} \to \Re$ and the last $b_2 : \{1, 2, \ldots, M\} \to \Re$ map the side-1 and side-2 indices to the reconstructed value of base-part, respectively. Now we can formally define the three mappings as follows:

$$b_k(i) = \tilde{X}_{B_j}, \quad \forall i \in \{i_k^{j,1}, i_k^{j,2}, \ldots, i_k^{j,bn_k}\}, \quad k = 0, 1, 2. \tag{2}$$

As long as the decoder receives at least one description, we can reconstruct the value of base-part exactly by the mappings $b_k(\cdot)$, $k = 0, 1, 2$, depending on which description(s) are received. Since the possibility that the base-part of the source can be successfully reconstructed at the decoder is typically high, we may use the base-part reconstruction as the prediction reference for video encoding and decoding with low possible drift.

2.3. Base-MDSQ with partial prediction

Partial prediction is a simple yet effective technique for layered coding to combat against data loss in predictive coding, at the expense of an increased bit-rate. In [13], a small number of most significant bit-planes (3–4 bit-planes) of the reconstructed enhancement-layer data are used in the prediction loop of the base-layer encoder to enhance the coding efficiency of MPEG-4 FGS. In this work, we propose to combine the Base-MDSQ codec with partial prediction as depicted in Fig. 7 to enhance error resilience without sacrificing coding efficiency significantly. In the encoder shown in Fig. 7(a), we first apply motion compensated prediction and DCT on the source frame, and then perform MDSQ encoding on the DCT coefficients of prediction residue to obtain two descriptions (i.e., two side indices). Run-length coding and variable-length coding are subsequently performed on the two side indices to obtain two output bitstreams. By integrating partial feedback into the proposed Base-MDSQ coder as shown in Fig. 7(a), the base-part and enhancement-part of Base-MDSQ reconstructed data are fed back into the prediction loop with two different partial factors, α_B and α_E, respectively. When the decoder shown in Fig. 7(b) receives the two descriptions, it can retrieve both

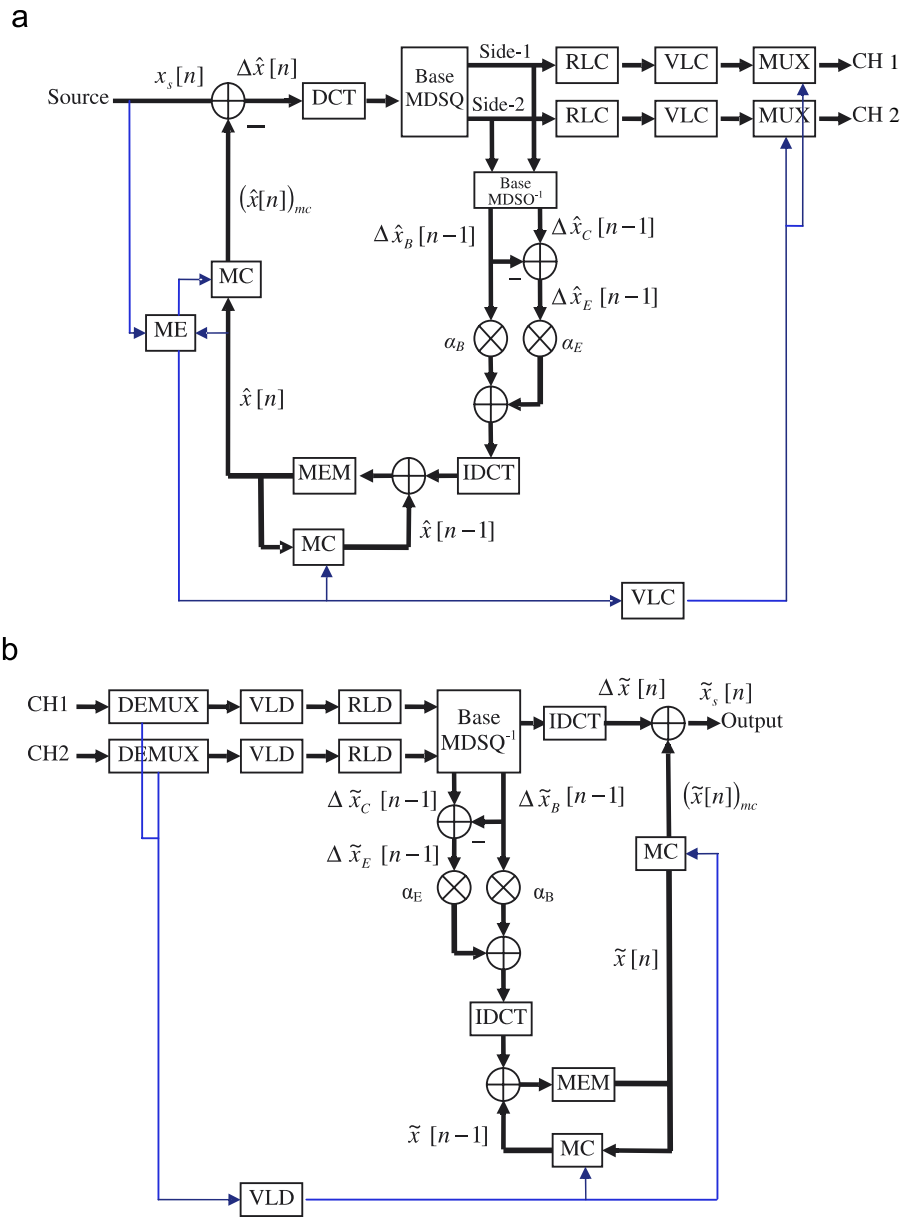

Fig. 7. Architectures of Base-MDSQ video codec with partial prediction: (a) encoder and (b) decoder.

the base-part and enhancement-part values of the source. In the case that only one description is received, the decoder can still reconstruct the base-part value of the source. Since the base-part can always be reconstructed at the decoder when at least one description is received, the use of the base-part in the prediction loop of the Base-MDSQ codec is much more reliable than the use of the central reconstruction (i.e., the sum of the base-part and enhancement-part). The value of α_B is therefore typically set relatively larger than the value of α_E.

Referring to Fig. 7(a), the prediction reference $\hat{x}[n]$ of the source sequence $x_s[n]$ is defined as follows:

$$\hat{x}[n] = \hat{x}[n-1] + \alpha_B \Delta\hat{x}_B[n-1] + \alpha_E \Delta\hat{x}_E[n-1], \tag{3}$$

where $\Delta\hat{x}_B[n]$ denotes the base-part reconstruction of the residue (see (4)) and $\Delta\hat{x}_E[n]$ the enhancement-part

which is the difference between the central reconstruction $\Delta\hat{x}_C[n]$ and the base-part reconstruction of residue $\Delta\hat{x}_B[n]$ (see (5)). $Q_{MD}(\cdot)$ and $IQ_{MD}(\cdot)$ represent the MDSQ (with index mapping) and inverse MDSQ for the DCT coefficients of prediction residue, respectively.

$$\Delta\hat{x}_C[n] \equiv IQ_{MD}(Q_{MD}(x_s[n] - \hat{x}[n])), \tag{4}$$

$$\Delta\hat{x}_E[n] \equiv \Delta\hat{x}_C[n] - \Delta\hat{x}_B[n]. \tag{5}$$

Note that, when being fed back, $\Delta\hat{x}_B[n]$ and $\Delta\hat{x}_E[n]$ are multiplied by the partial factors α_B and α_E, respectively, where $0 \leqslant \alpha_E \leqslant \alpha_B \leqslant 1$ as shown in (3). As a result, the reconstructed value $\tilde{x}_s[n]$, as shown in Fig. 7(b), of the source sample in the decoder can be recursively derived as follows:

$$\begin{aligned}
\tilde{x}_s[n] &= (\tilde{x}[n])_{mc} + \Delta\tilde{x}[n] \\
&= (\tilde{x}[n-1] + \alpha_B\Delta\tilde{x}_B[n-1] + \alpha_E\Delta\tilde{x}_E[n-1])_{mc} \\
&\quad + \Delta\tilde{x}[n] = (\tilde{x}[n-1])_{mc} + \alpha_B(\Delta\tilde{x}_B[n-1])_{mc} \\
&\quad + \alpha_E(\Delta\tilde{x}_E[n-1])_{mc} + \Delta\tilde{x}[n] \\
&= (\tilde{x}[n-2] + \alpha_B\Delta\tilde{x}_B[n-2] + \alpha_E\Delta\tilde{x}_E[n-2])_{mc} \\
&\quad + \alpha_B(\Delta\tilde{x}_B[n-1])_{mc} + \alpha_E(\tilde{x}_E[n-1])_{mc} \\
&\quad + \Delta\tilde{x}[n] = (\tilde{x}[n-2])_{mc} + \alpha_B(\Delta\tilde{x}_B[n-2])_{mc} \\
&\quad + \alpha_B(\Delta\tilde{x}_B[n-1])_{mc} \\
&\quad + \alpha_E(\Delta\tilde{x}_E[n-2])_{mc} \\
&\quad + \alpha_E(\Delta\tilde{x}_E[n-1])_{mc} + \Delta\tilde{x}[n] = \cdots \\
&= (\tilde{x}[1])_{mc} + \alpha_B\sum_{i=1}^{n-1}(\Delta\tilde{x}_B[i])_{mc} + \alpha_E\sum_{i=1}^{n-1}(\Delta\tilde{x}_E[i])_{mc} + \Delta\tilde{x}[n],
\end{aligned}$$

where the subscript mc indicates that $(y)_{mc}$ is the motion compensated version of y, and $\Delta\tilde{x}[n] - \Delta\tilde{x}_C[n] = \Delta Q[n]$ is the quantization error. The error propagation caused by the transmission error of $\Delta\tilde{x}_B[i]$ and $\Delta\tilde{x}_E[i]$ is effectively reduced by the partial factors α_B and α_E, respectively.

The partial factors can be used to trade coding efficiency for error resilience. When both partial factors are equal to unity, the partial prediction is the same as the standard coder. For factors smaller than unity, the actually used prediction reference is discounted, so the prediction error is larger and we need more bits to encode the residue. However, since the coding of subsequent samples is less dependent on previous residues, the loss of previous residue will not cause too much error propagation to the subsequent samples referencing to the lost residue. Hence partial prediction reduces the drifting error in the predictive coding, at the expense of prediction efficiency.

2.4. MD–FGS with partial prediction

As mentioned above, the coding efficiency of MPEG-4 FGS is significantly lower than that of MPEG-4 non-scalable coder because MPEG-4 FGS only utilizes the lowest quality base-layer for motion prediction. Using higher quality reference frames in the enhancement-layer can improve coding efficiency, but will lead to drifting error. Based on the proposed Base-MDSQ with partial prediction coding scheme, we also propose a novel MD–FGS coding technique to improve the error resilience of MPEG-4 FGS while retaining good coding performance. The encoder and decoder architectures of the proposed MD–FGS codec are shown in Fig. 8. In the base-layer MD–FGS encoder, the intra DC coefficients and motion vectors are simply duplicated in the two output descriptions, whereas the remaining coefficients are encapsulated into the two descriptions using an MDSQ encoder. The enhancement-layer data are generated by first calculating the difference between the DCT coefficients of the source video and the reconstructed base-layer video. The resulting residual signal is then encoded by the Base-MDSQ coder into two descriptions (i.e., two side indices). The side indices are subsequently encoded with run-length coding and variable-length coding, and packetized. The base-part of the enhancement-layer is extracted from the Base-MDSQ coder and added into the prediction loop of the base-layer video to obtain a higher quality reference for motion prediction. Since the possibility of receiving at least one description of the enhancement-layer data is high, using the base-part of the enhancement-layer achieves much stronger error robustness than directly using partial MSB planes of the enhancement-layer data as in [13]. The partial factor α is used to attenuate the drifting error in the extreme case that both enhancement-layer descriptions are lost. However, a smaller α will lead to lower coding efficiency when the base-part of Base-MDSQ coder can be reconstructed correctly.

3. Robust roaming using MD–FGS

Roaming across multiple basic service sets (BSS) in wireless LANs poses considerable challenges, as it may cause heavy data loss that will severely degrade the receiving video quality. We investigate the use of the proposed MD–FGS scheme in the three-tier streaming system to enhance the robustness to deal with packet loss in WLANs, especially while a mobile device is in the need of handoff.

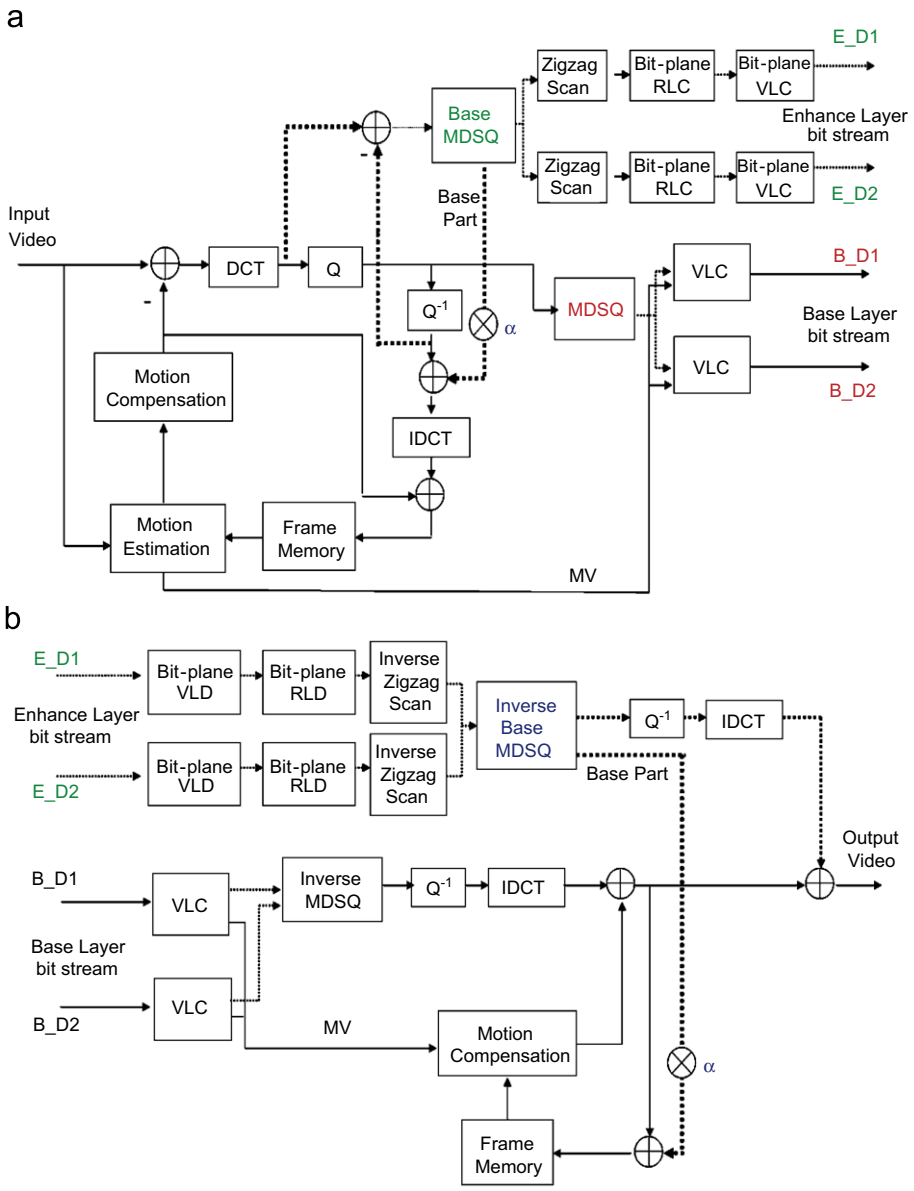

Fig. 8. Architectures of the proposed MD–FGS video codec: (a) encoder and (b) decoder.

3.1. Testbed for multipath streaming

Fig. 9 illustrates our proposed testbed for multipath streaming which is composed of four personal computers serving as an media gateway, two base stations (STA_0 and STA_1), and one client, to emulate possible roaming situations among the base stations and the wireless client. In the testbed, stations STA_0 and STA_1 are responsible for detecting the clients located in its communication zone, and then reporting to the media gateway the channel statistics about the clients they are serving. According to the feedback information, the media gateway maintains a path-list table that records which clients can be reached by the station. These paths are sorted according to their channel conditions. If the difference of the PLRs of two channels is less than a threshold, the channel conditions are sorted by the PLRs directly; otherwise, the round-trip times (RTTs) will be used instead. If both the PLRs and RTTs of two channels are not significantly different, the two channels are considered to have similar qualities. The status of a path to a client is updated when any new channel information about the client is received.

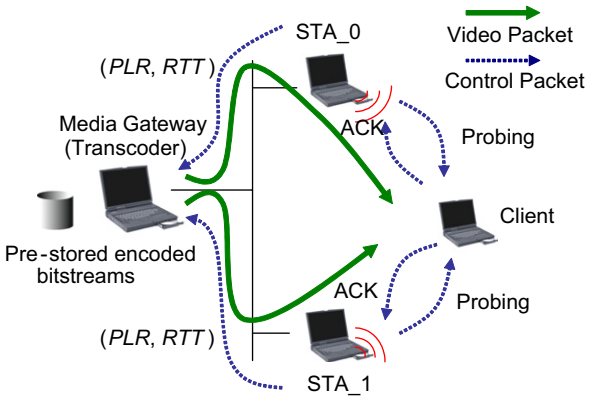

Fig. 9. Proposed testbed for real-time multipath streaming with one media gateway (video transcoder), two base stations, and one mobile client.

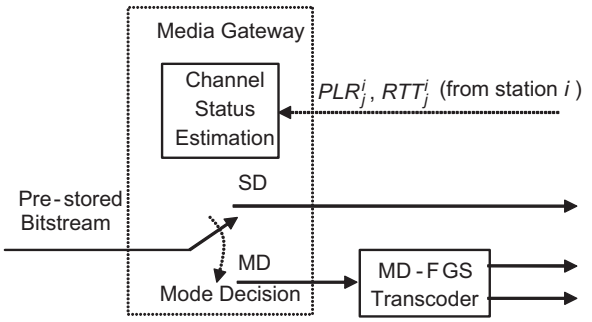

Fig. 10. Proposed adaptive error control in the media gateway.

If the media gateway 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 temporarily marked as non-connectable. According to the path-list table, the media gateway will either directly forward the video data to the client through only one channel with significantly better channel condition (SD mode), or transcode the video into two descriptions and then send the two descriptions respectively through two channels that have similar channel qualities (MD mode).

3.2. Adaptive transcoding based on channel estimation

3.2.1. Adaptive MD–FGS transcoding

For simplicity of implementation but without loss of generality, we assume there are two transmission paths between the media gateway and the client via STA_0 and STA_1. In the proposed error control mechanism shown in Fig. 10, when the media gateway receives a pre-encoded video bitstream from the streaming server, it will adaptively choose either of the two operational modes: the SD mode and the MD mode, to insert different error-resilient features according to the estimated conditions of both transmission channels. In the case that one channel's condition is good enough for sending the video, the media gateway will be operated in the SD mode by which the pre-stored bitstream is directly forwarded to the client without any modification via the channel with the best condition. Since this mode chooses a transmission path that is in a good condition and not to perform the MD–FGS transcoding, the video data can be delivered to the client reliably without sacrificing any coding efficiency. When the media gateway is operated in the MD mode, it transcodes the pre-stored bitstream into two descriptions using the MD–FGS coding scheme for diverse delivery via both channels simultaneously. This mode can provide the strongest protection 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 for error protection. As shown in Fig. 9, while a mobile user moves from STA_0 to STA_1, the media gateway should know when the packets will be streamed only by STA_0 (STA_1), or STA_0 and STA_1 simultaneously based on the channel estimator.

3.2.2. Channel estimation

With the medium access mechanism of IEEE 802.11 [14], if the condition of a channel is poor, the transmit station needs more retransmissions to successfully deliver the video data to the receiver, leading to an increase of delay time for the data to arrive at the receiver side. Furthermore, it will also lead to a higher PLR. These two attributes have been taken as congestion indicators in many existing congestion control schemes [26,5]. In this work, we also exploit the two attributes to estimate the channel conditions; no matter they are caused by network congestion or by the unreliable transmission media. Typically the PLR of a downstream channel directly reflects the condition of the channel, whereas RTT involves the delays of both downstream and upstream channels, which may not be directly related to the condition of downstream channel. We therefore choose PLR as the first indicator of path change and then check RTT, should it be difficult to differentiate the channel conditions using PLR solely. This concept was also used in the network detection scheme presented in [20].

In our streaming system, the base stations use the User Datagram Protocol (UDP) to send probe frames to 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 the 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 the probe frames and also in the ACK frames to keep low extra overhead cost. For the probing, it is difficult to measure the delay time accurately unless the clock times of the station and the client are well synchronized. Using RTT 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 experiments, the base stations use a sliding window to calculate $\overline{\mathrm{RTT}}_k^i$ and $\overline{\mathrm{PLR}}_k^i$ of each channel between the stations and the client, and report these channel statistics to the media gateway as references to guide the selection of channels, where $\overline{\mathrm{RTT}}_k^i$ and $\overline{\mathrm{PLR}}_k^i$ denote the average RTT and the average PLR in a sliding time interval C of a fixed number of packets, respectively, as follows:

$$\overline{\mathrm{RTT}}_k^i = \frac{1}{C}\sum_{j=k-C+1}^{k} \mathrm{RTT}_j^i, \tag{6}$$

$$\overline{\mathrm{PLR}}_k^i = \frac{1}{C}\sum_{j=k-C+1}^{k} \mathrm{PL}_j^i, \tag{7}$$

$$\max \mathrm{RTT}^i = \mathrm{Max}(\mathrm{RTT}_k^i, \max \mathrm{RTT}^{i-1}), \tag{8}$$

where RTT_j^i stands for the RTT of the jth packet through the ith channel, $\mathrm{PL}_j^i = 1$ indicates the jth packet through the ith channel gets lost or corrupted, and $\mathrm{RTT}_j^i = \max \mathrm{RTT}^{i-1}$ if $\mathrm{PL}_j^i = 1$.

As illustrated in Fig. 11, 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 channel. 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 the proposed method, it is not necessary to estimate the quality of each channel accurately, but instead, estimating the relative qualities of the two channels is sufficient for the mode decision. The proposed mode decision algorithm for error resilience transcoding based on channel status estimation is summarized below. The same decision algorithm is also applied to select the best path during the initial setup period.

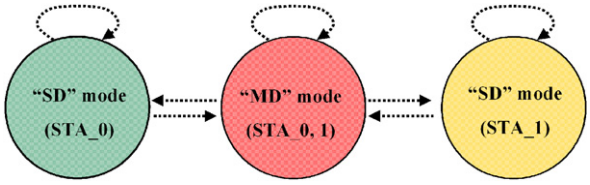

Fig. 11. State transition diagram of the proposed mode decision scheme.

Algorithm: Channel-Aware Mode Decision for Error Resilience Transcoding

At the SD mode:

if $(\overline{\mathrm{PLR}_k^i} - \overline{\mathrm{PLR}_k^j} > \mathrm{PLR}_{\mathrm{LowerBound}})$
Stay at the SD mode: forward the video data through channel j (via STA_ j) without transcoding
else
{
if
$(\overline{\mathrm{RTT}_k^i} - \overline{\mathrm{RTT}_k^j} > \mathrm{RTT}_{\mathrm{LowerBound}} \times \overline{\mathrm{RTT}_k^i})$
Stay at the SD mode: forward the video data through channel j (via STA_ j) without transcoding

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

At the MD mode:

if $(\overline{\mathrm{PLR}_k^i} - \overline{\mathrm{PLR}_k^j} > \mathrm{PLR}_{\mathrm{UpperBound}})$
Switch to the SD mode: forward the video data through channel j (via STA_ j) without transcoding
else
{
if
$(\overline{\mathrm{RTT}_k^i} - \overline{\mathrm{RTT}_k^j} > \mathrm{RTT}_{\mathrm{UpperBound}} \times \overline{\mathrm{RTT}_k^i})$
Switch to the SD mode: forward the video data through channel j (via STA_ j) without transcoding
else
Stay at the MD mode: transcode the video data into two descriptions, and send the two descriptions via both channels
}

4. Experimental results

We first evaluate the performance of the proposed Base-MDSQ and MD–FGS coding schemes with partial prediction by computer simulations. We then use the OPNET network simulator [21] to evaluate the performance of channel status detection mechanism. Finally, we evaluate the performance of the proposed roaming mechanism on our multi-path streaming testbed described in Section 3.1.

4.1. Performance evaluation for Base-MDSQ

In order to evaluate the performance of Base-MDSQ video coder with partial prediction, we assume that the video bitstreams are transmitted in a packet-erasure channel, such as a wireless LAN. We use an MPEG-4 encoder [15] to encode the *Foreman*, *Akiyo* and *Coastguard* test sequences at 144 Kbps and 7.5 fps with a GOP (Group of Pictures) size of 15 frames without B-pictures. The video package size is 512 bytes. We adopt the zero-motion replacement error concealment [17] for concealing the lost video data. In our experiments, intra DC coefficients and motion vectors are duplicated in both descriptions. The width of diagonal lines in the index assignment matrix of normal MDSQ is 3 (i.e., $k = 1$). To model the behavior of real-life networks, the statistics of packet losses obtained from the ITU document [34] is applied. Based on the error patterns of packet loss, we simulate all the algorithms at five PLRs: 0%, 3%, 5%, 10% and 20%, respectively.

Table 1
PSNR of central reconstruction and base-part only for sequences at 144 Kbps

Sequence	PSNR (dB) of central reconstruction	PSNR (dB) of base-part only	PSNR difference (dB)
Foreman	31.30	26.89	4.41
Coastguard	29.96	25.01	4.95

In the following we present the reconstruction quality of base-part reconstruction and central reconstruction of DCT coefficients in video coding. As shown in Table 1, the PSNR of base-part only is about 4–5 dB lower than the central reconstruction for two test sequences. Fig. 12 shows the reconstructed pictures of a P-frame with the central reconstruction and base-part reconstruction for *Foreman*, respectively.

In our proposed Base-MDSQ codec, since the base-part of residue can be reconstructed at the decoder much more reliably than the enhancement-part, we simply set $\alpha_B = 1$ and $\alpha_E = 0$. The parameters used in the simulations are listed in Table 2. Fig. 13 compares the average PSNR performance of the proposed Base-MDSQ coder and the standard (conventional) MPEG-4 coder, which shows that the Base-MDSQ coder significantly reduces the distortion at medium to high loss rates, since the base-part used for prediction can be retrieved with high possibility. Although the redundancy reduces the coding efficiency in a clean channel, the overall performance of Base-MDSQ in error-prone environments is promising.

4.2. Performance evaluation for MD–FGS

For performance evaluation, we implemented an "orgMD–FGS" coder that applies the original MDSQ described in Section 2.1 to quantize the DCT coefficients of both the base-layer and enhancement-layer. As shown in Fig. 8, the

Fig. 12. Reconstructed pictures of a P-frame with the central reconstruction (left) and base-part only (right) of *Foreman*.

Table 2
Parameters used in the simulations for Base-MDSQ

Test sequences	*Foreman, akiyo and coastguard*
Data rate	144 Kbps
Frame rate	7.5 fps
GOP size	15 frames
GOP structure	IPP…
k of MDSQ	3
α_B of Base-MDSQ	1
α_E of Base-MDSQ	0
Packet loss rates	0%, 3%, 5%, 10% and 20%

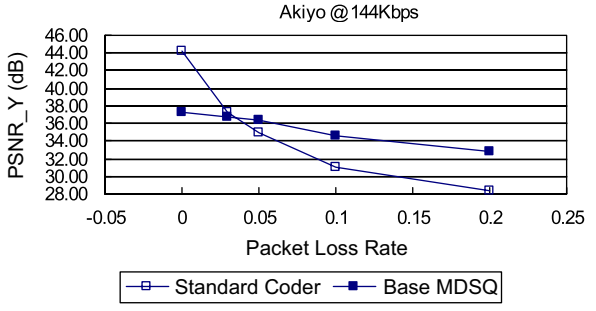

Fig. 13. Average PSNR performance comparison between Base-MDSQ and standard non-scalable MPEG-4 coder for *Akiyo* at 144 Kbps.

MD–FGS coder employs the proposed Base-MDSQ to quantize the DCT coefficients of the enhancement-layer data into two descriptions and then to extract the base-part of the two descriptions to obtain a higher quality reference for motion prediction in the base-layer coder. Oppositely, orgMD–FGS uses the original MDSQ to quantize the DCT coefficients of enhancement-layer into two descriptions without extracting any feedback from the enhancement-layer to base-layer for prediction. This is the main difference between the two coders.

The performance of MD–FGS is compared with the orgMD–FGS coder and the standard MPEG-4 FGS coder. We implement the algorithms on the reference MPEG-4 video coder with FGS-scalability. Only the first frames of each test sequence is encoded as an I-frame, and the others are coded as P-frames. The base-layer bitstreams of MD–FGS and orgMD–FGS coders are coded at a fixed bit-rate of 32 Kbps for each description (i.e., B_D1 and B_D2) by the TM5 rate control to carry the minimum fundamental data, including headers and motion vectors. The enhancement-layer bitstreams of MD–FGS and orgMD–FGS are encoded at 128 Kbps for each description (i.e., E_D1 and E_D2), respectively. For fair comparison, we apply a fixed bit-rate of 64 Kbps for the base-layer bitstream and 256 Kbps for the enhancement-layer bitstream of the standard MPEG-4 FGS coder. The width of diagonal lines in the index assignment matrix of MD–FGS and orgMD–FGS is three.

Fig. 14 shows that at error-free condition the coding efficiency of MD–FGS is better than orgMD–FGS by about 2.22 dB at the same bit-rate, but lower than MPEG-4 FGS by about 1.87 dB on average. Due to the higher redundancy of MDC, a coder with MDC typically has relatively lower coding efficiency compared to the coder without MDC at error-free condition. Table 2 compares the required bit-rates for the enhancement-layer bitstreams of MPEG-4 FGS, MD–FGS and orgMD–FGS coders to achieve reconstruction quality of about 37 dB at error-free condition, when the base-layers of the three coders are all encoded at 64 Kbps. Table 3 indicates that, to achieve similar quality, the bit-rates of MD–FGS and orgMD–FGS coders are higher than that of MPEG-4 FGS by 43% and 68%, respectively.

Fig. 15 compares the average PSNR performances of the three coders for the enhancement-layer PLR ranging from 0% to 15% for both E_D1 and E_D2, under two different base-layer PLRs, 0% and 1%, for both B_D1 and B_D2, respectively. It indicates that the reconstructed video quality of MPEG-4 FGS will be degraded rapidly while packet loss occurs in the base-layer bitstream even if the base-layer PLR is low. Fig. 15 shows that the PSNR

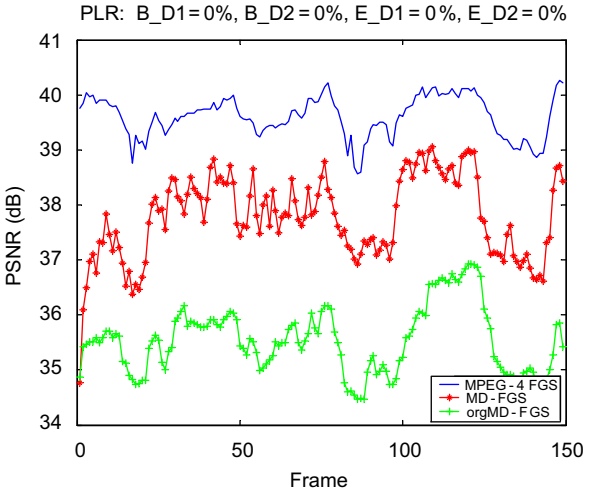

Fig. 14. Frame-by-frame PSNR comparison at PLR = 0% for all B_D1, B_D2, E_D1 and E_D2 bitstreams.

Table 3
Required bit-rates of enhancement-layer bitstreams for MPEG-4 FGS, MD–FGS and orgMD–FGS to achieve the same reconstruction quality in error-free condition

Coder	Bit-rate of enhancement-layer bitstream (Kbps)
MPEG-4 FGS	177.8
MD-FGS	254.6
orgMD-FGS	299.6

performance of MD–FGS is significantly better than that of orgMD–FGS when the enhancement-layer PLR = 0%–7% under the two base-layer PLRs (0% and 1%), and also than that of MPEG-4 FGS when the enhancement-layer PLR = 0%–15% under the base-layer PLR of 1%. The PSNR performance of MD–FGS becomes poorer than that of orgMD–FGS at high PLRs, since the drift caused by the loss of the base-part of enhancement-layer becomes severer than the coding gain obtained from using the base-part prediction.

We also compare the performance of MD–FGS to a non-layered MD coder, named orgMD, to quantify the advantage of using a combination of FGS and MDC over non-layered MDC. The orgMD coder adopts the original MDSQ to quantize the DCT coefficients into two descriptions which are both encoded at 160 Kbps, respectively. Fig. 16(a) shows the average PSNR performances using four coders, including two-layered MD coders (MD–FGS and orgMD–FGS), one layered non-MD coder (MPEG-4 FGS), one non-layered MD coder (orgMD), for a range of PLRs. In the experiment, we assume equal error protection (EEP) is adopted so that the PLRs for the base-layer and enhancement-layer bitstreams of layered coders and the bitstreams of non-layered coder are the same with a range of 0–15%. As shown in Fig. 16(a), orgMD has better PSNR performance at low to medium PLRs compared to the layered MDC coders because of its better coding efficiency; however, its performance deteriorates rapidly at high PLRs due to heavy drifting error. To enhance error resiliency, a layered coder can perform unequal error protection (UEP) [31] on the base-layer and enhancement-layer bitstreams so that the loss probability of base-layer data are relatively lower than that of enhancement-layer data. As an example, Fig. 16(b) shows the average PSNR performance comparison for a UEP scenario

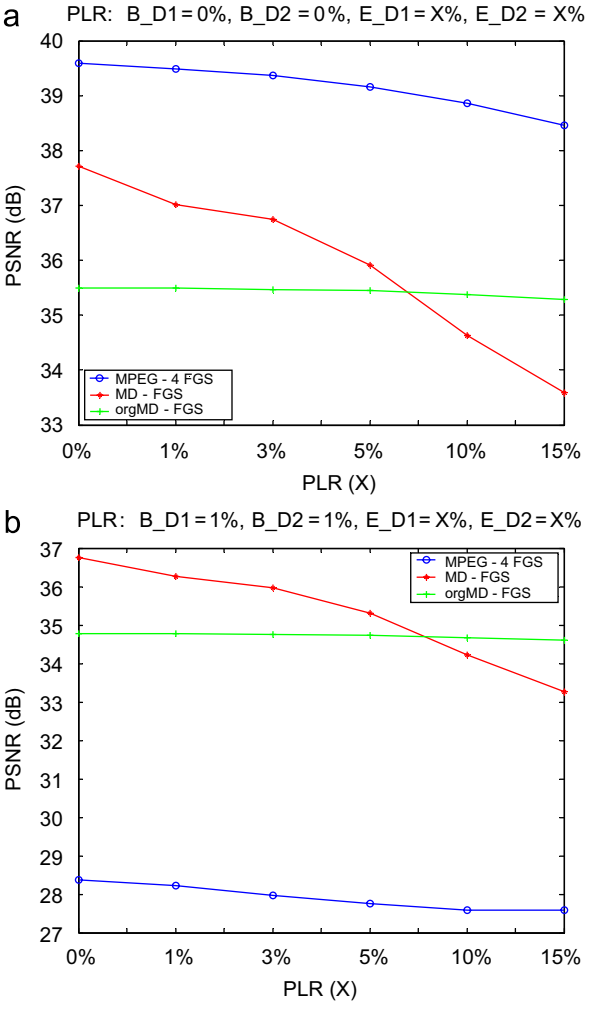

Fig. 15. Average PSNR performance comparison versus the enhancement-layer PLR ranging from 0% to 15% for both E_D1 and E_D2 with two different base-layer PLRs: (a) base-layer PLR = 0% and (b) base-layer PLR = 1% for both B_D1 and B_D2.

in which the PLR of enhancement-layer (PLR_E) is three times of the base-layer's (PLR_B), assuming that the average PLR of both layers is the same as that of the single-layered bitstreams (also ranging from 0% to 15%) for fair comparison. That is, $PLR_E = 3PLR_B$, $(PLR_B + PLR_E)/2 = PLR$ with PLR = 0%–15%. As shown in Fig. 16(b), with UEP on the base-layer and enhancement-layer bitstreams, the PSNR performances of non-layered MD coder (orgMD) and layered non-MD coder (MPEG-4 FGS) deteriorate relatively rapidly compared to the layered MD coders while PLR increases. The PSNR performance of MD–FGS is

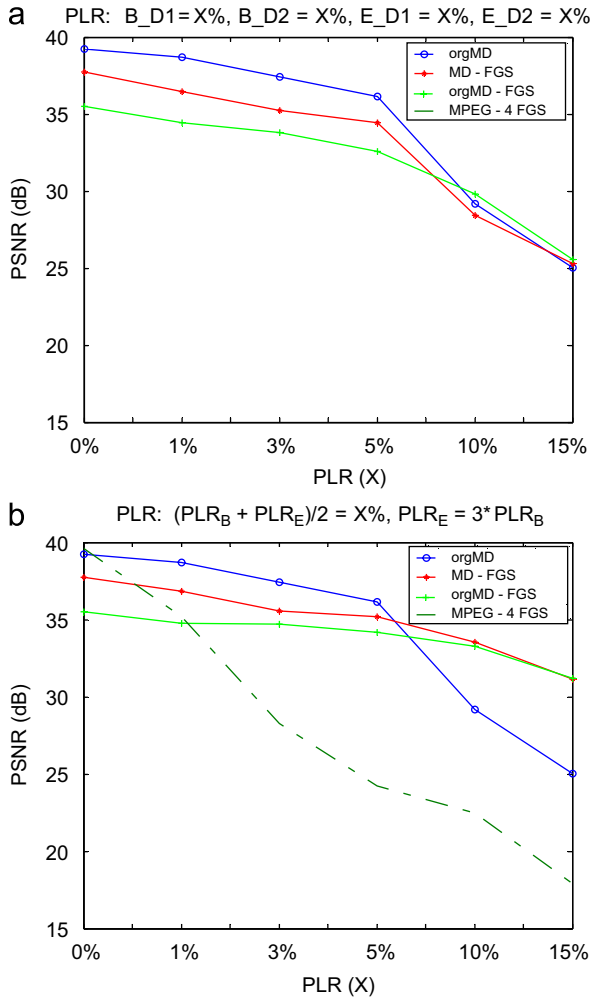

Fig. 16. Average PSNR performance comparisons with non-layered coder with (a) equal error protection (both the base-layer and enhancement-layer have the same PLR), and (b) unequal error protection (the enhancement-layer PLR is three times higher than the base-layer PLR).

better than that of orgMD–FGS at most PLRs under the EEP and UEP scenarios.

The selection of partial factor α for appropriately controlling drift is dependent on the channel condition. We focus on the selection of α for the range of enhancement-layer PLRs that will lead to significant drift at the base-layer due to the loss of base-part in Base-MDSQ. Fig. 17 illustrates the impact of the value of α on the average PSNR performance of MD–FGS at the error-free condition for the base-layer descriptions (B_D1 and B_D2) under different enhancement-layer PLRs for E_D1 and E_D2 (0%–15%). From Fig. 17, we can observe that the optimal value of partial factor

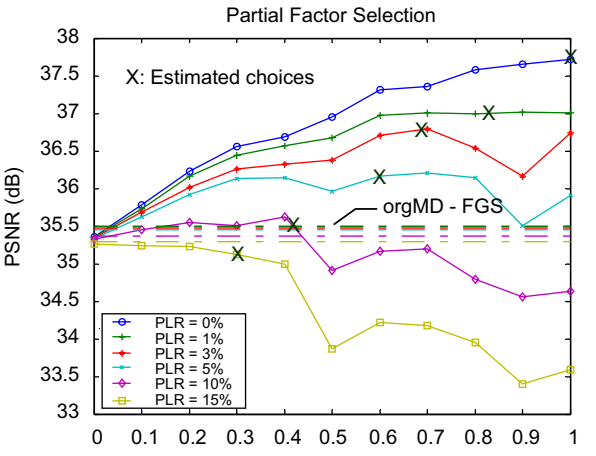

Fig. 17. Average PSNR comparison of MD–FGS and orgMD–FGS versus different partial factor α at PLR = 0% for both B_D1 and B_D2, and PLR = 0%–15% for E_D1 and E_D2.

α will decay along with the increase of PLR. The relation of PLR and the optimal values of partial factor α that achieve the best PSNR performances can be approximately modeled by

$$\alpha_{\text{est}} = 1 - k_{\alpha}\sqrt{\text{PLR}}, \tag{9}$$

where $k_{\alpha} = 1.8$ according to the experimental results. The marks "X" in Fig. 17 indicate the PSNR values for different PLRs using the values of α estimated by (9). We can observe that the PSNR values using PSNR α_{est} under the all PLRs are all close to the corresponding optimums. With the adaptive partial factor setting, MD–FGS outperforms orgMD–FGS in PSNR performance in the range of enhancement-layer PLR = 0%–10%, that is a reasonable working range for video transport in WLANs.

4.3. Performance evaluation of robust roaming

In the experiments, we first use the OPNET network simulator to generate channel patterns to evaluate the performance of the proposed channel status detection scheme. We then evaluate the performance of the proposed roaming scheme on our multipath streaming testbed. The source stream is generated using an MPEG-4 FGS reference coder. The wireless domain is configured at the "ad-hoc" mode with the AP functionality disabled. The parameters used in the experiments are listed in Table 4. Fig. 18 shows the traces of PLR and the traces of round-trip delay of the two

channels via STA_0 and STA_1, respectively, while client moves from STA_0 to STA_1. The channel selection result determined by the transcoder is also shown in Fig. 18, which indicates both paths via STA_0 and STA_1 are used to deliver the two descriptions sent from the MD–FGS transcoder between time instance 46 and 56. Finally, the channel is switched fully to the path via STA_1 after time instance 56.

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%. A factory default transmission retry limit of seven times is configured for the base stations. We simulate a roaming between stations STA_0 and STA_1 with a scenario that a client moves from station STA_0 toward station STA_1 at a pedestrian speed. In the experiment, we provide heavier protection to the base-layer packets by simply duplicating the base-layer packets.

Fig. 19 shows the channel utilization versus time. We can observe that, in the MD mode, the data rate is about 45% higher than that of the SD mode. This higher data rate only appears in very short periods during channel transition. The frame-by-frame PSNR performances of roaming with and without the proposed MD–FGS transcoder are compared in Fig. 20. The results indicate that the proposed roaming scheme with MD–FGS transcoding can greatly reduce the distortion caused by roaming with transitions, even though there still are some unavoidable packet losses at the MD mode.

Table 4
Parameters used in the simulation

Parameters for wireless LAN	
Data rate (bps)	11 Mbps
Physical characteristics	Frequency hopping
Access point functionality	Disabled
Parameters for environment setup	
Distance between STA_0 and STA_1	300 m
Moving speed of mobile users	4 km/h
Parameters for channel status estimation	
Probing period (T_{probe})	100 ms
$PLR_{LowerBound}$	0.2
$PLR_{UpperBound}$	0.4
$RTT_{LowerBound}$	0.6
$RTT_{UpperBound}$	1.2

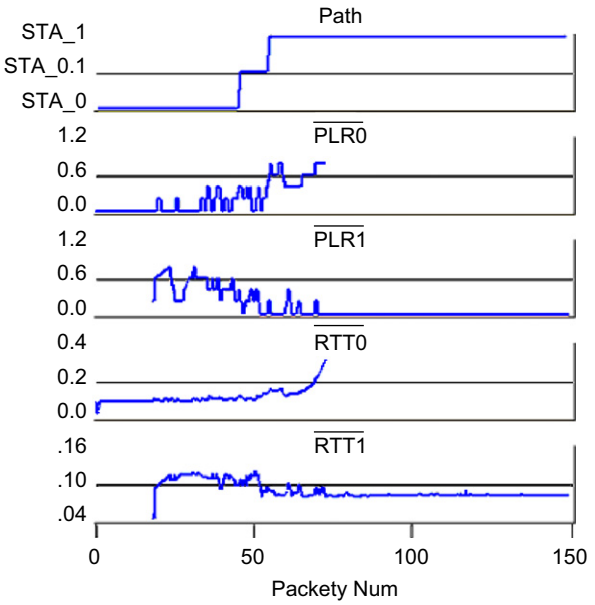

Fig. 18. Channel selection result and $\overline{PLR}$, $\overline{RTT}$ of the channels by STA_0 and STA_1.

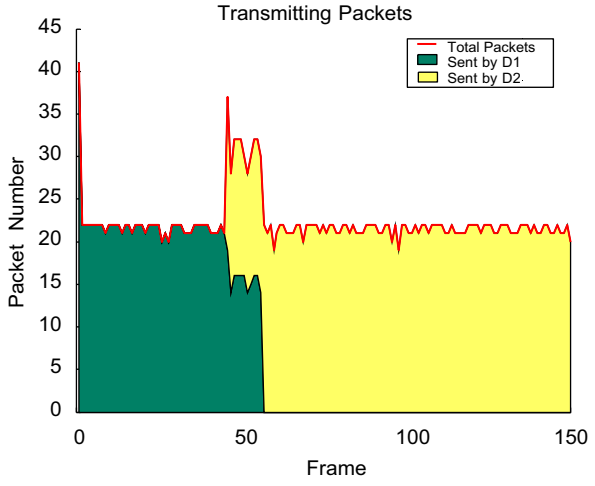

Fig. 19. Frame-by-frame channel utilization.

5. Conclusion

In this paper, we proposed a Base-MDSQ scheme by modifying the index assignment matrix of MDSQ. Based on Base-MDSQ, we have also proposed a novel MD–FGS coder to enhance error-resilience while retaining the temporal prediction efficiency for video streaming over lossy networks. Combining MD–FGS and multipath routing, we have proposed an adaptive MD–FGS

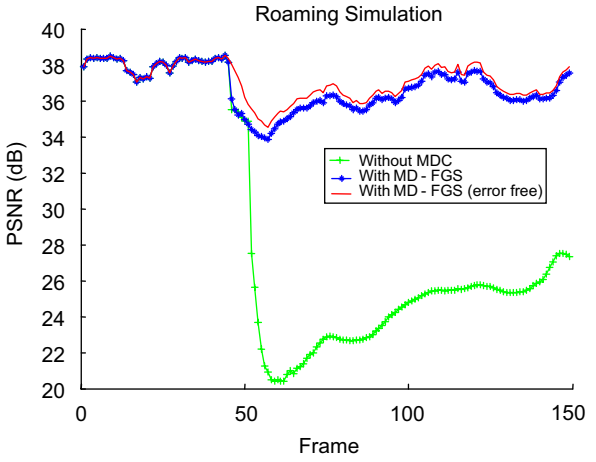

Fig. 20. Frame-by-frame PSNR performance comparison between roaming with and without the proposed MD–FGS transcoding.

transcoding method to tackle packet loss while a mobile client roams in WLANs. When a mobile client roams from one station to another, the proposed transcoder looks for whether there exists another path with an acceptable channel condition to be used simultaneously for path diversity. We proposed to use the feedback information, RTT and PLR, to estimate the channel condition to guide the channel switching. Finally, we have established a multipath streaming testbed to verify the performance of the proposed scheme. Experimental results show that the proposed Base-MDSQ and MD–FGS coders with partial prediction can increase the temporal prediction efficiency without introducing severe drifting error when packet loss occurs. Similarly, the use of path diversity as temporary step for roaming can provide significant benefits compared to the roaming among access points.

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] E. Akyol, A.M. Tekalp, M.R. Civanlar, 2005. Scalable multiple description video coding with flexible number of descriptions, In: Proceedings of the IEEE International Conference Image Processing, vol. 3, September 2005, pp. 712–715.

[2] E. Akyol, A.M. Tekalp, M.R. Civanlar, 2006. Adaptive peer to peer video streaming with flexible multiple description coding, In: Proceedings of the IEEE International Conference Image Processing, October 2006.

[3] J. Apostolopoulos, 2001. Reliable video communication over lossy packet networks using multiple state encoding and path diversity, In: Proceedings of the SPIE Conference Visual Communications and Image Processing, 2001, pp. 392–409.

[4] N.V. Boulgouris, K.E. Zachariadis, A.N. Leontaris, M.G. Strintzis, 2001. Drift-free multiple description coding of video, In: Proceedings of the IEEE Workshop on Multimedia Signal Processing, October 2001, pp. 105–110.

[5] S. Cen, P.C. Cosman, G.M. Voelker, End-to-end differentiation of congestion and wireless losses, IEEE/ACM Trans. Networking 11 (October 2003) 703–717.

[6] J. Chakareski, E. Setton, Y. Liang, B. Girod, 2003. Video streaming with diversity, In: Proceedings of the IEEE International Conference Multimedia and Expo, vol. 1, pp. 9–12.

[7] P.A. Chou, H.J. Wang, V. Padmanabhan, 2003. Layered multiple description coding, In: Proceedings of the Packet Video Workshop, April 2003.

[8] G. Cote, S. Shirani, F. Kossentini, Optimal mode selection and synchronization for robust video communications over error-prone networks, IEEE J. Select. Areas Commun. 18 (6) (June 2000) 952–965.

[9] 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 (9) (September 2002) 777–792.

[10] V.K. Goyal, Multiple description coding: compression meets the network, IEEE Signal Processing Magazine 18 (5) (September 2001) 74–93.

[11] V.K. Goyal, J. Kovacevic, 1998. Optimal multiple description transform coding of Gaussian vectors, In: Proceedings of the IEEE Data Compression Conference, March 1998, pp. 388–397.

[12] S. Han, B. Girod, 2002. Robust and efficient scalable video coding with leaky prediction, In: Proceedings of the IEEE International Conference Image Processing, vol. 2, September 2002, pp. 41–44.

[13] H.-C. Huang, C.-N. Wang, T. Chiang, A robust fine granularity scalability using trellis-based predictive leak, IEEE Trans. Circuits Syst. Video Technol. 12 (6) (June 2002) 372–385.

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

[15] ISO/IEC 14496 (MPEG-4). Video Reference Software, Version: Microsoft-FDAM1-2.3-001213.

[16] 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 (9) (September 2001) 1011–1020.

[17] W.M. Lam, A.R. Reibman, B. Liu, 1993. Recovery of lost or erroneously received motion vectors, In: Proceedings of the IEEE International Conference Acoustic, Speech, Signal Processing, Minneapolis, MN, April 27–30, 1993, pp. V417–V420.

[18] S. Mao, S. Lin, S.S. Panwar, Y. Wang, E. Celebi, Video transport over ad hoc networks: Multistream coding with multipath transport, IEEE J. Select. Areas Commun. 21 (December 2003) 1721–1737.
[19] A. Miu, J.G. Apostolopoulo, W.-T. Tan, M. Trott, 2003. Low-latency wireless video over 802.11 networks using path diversity, In: Proceedings of the IEEE International Conference Multimedia and Expo, August 2003, pp. 441–444.
[20] S. Na, J. Ahn, 2000. TCP-like flow control algorithm for real-time application, In: Proceedings of the IEEE International Conference Networks, 2000, pp. 99–104.
[21] OPNET, 2009. http://www.opnet.com/
[22] 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 (3) (March 2002) 193–204.
[23] E. Steinbach, N. Färber, B. Girod, Standard compatible extension of H.263 for robust video transmission in mobile environments,, IEEE Trans. Circuits Syst. Video Technol. 7 (6) (December 1997) 872–881.
[24] Streaming video profile–Final Draft Amendment (FDAM 4), 2001. ISO/IEC JTC1/SC29/WG11/N3904, January 2001.
[25] R. Swann, N. Kingsbury, 1996. Transcoding of MPEG-II for enhanced resilience to transmission errors, In: Proceedings of the IEEE International Conference Image Processing, vol. 2, 1996, pp. 813–816.
[26] K. Tan, Q. Zhang, W. Zhu, 2003. An end-to-end rate control protocol for multimedia streaming in wired-cum-wireless environments, In: Proceedings of the IEEE International Symposium on Circuits and Systems, May 2003, pp. 836–839.
[27] V.A. Vaishampayan, Design of multiple description scalar quantizers, IEEE Trans. Inf. Theory 39 (3) (May 1993) 821–834.
[28] F. Verdicchio, A. Munteanu, A. Gavrilescu, J. Cornelis, P. Schelkens, 2005. Error-resilient video coding using motion compensated temporal filtering and embedded multiple description scalar quantizers, In: Proceedings of the IEEE International Conference Image Processing, vol. 3, September 2005, pp. 932–935.
[29] Y. Wang, S. Lin, Error resilient video coding using multiple description motion compensation, IEEE Trans. Circuits Syst. Video Technol. 12 (6) (June 2002) 438–453.
[30] H. Wang, A. Ortega, 2003. Robust video communication by combining scalability and multiple description coding techniques, In: Proceedings of the SPIE Electronic Imaging: Image Video Communication Processing, vol. 5022, January 2003, pp. 111–124.
[31] Y. Wang, Q.-F. Zhu, Error control and concealment for video communication: a review, Proc. IEEE 86 (5) (May 1998) 974–997.
[32] Y. Wang, M.T. Orchard, V.A. Vaishampayan, A.R. Reibman, Multiple description coding using pairwise correlating transforms, IEEE Trans. Image Processing 10 (3) (March 2001) 351–366.
[33] Y. Wang, A. Reibman, S. Lin, Multiple description coding for video delivery, Proc. IEEE 93 (1) (2005) 57–70.
[34] S. Wenger, 1999. Error patterns for internet experiments, ITU Telecommunications Standardization Sector, Red Bank, New Jersey, Doc. Q15-I-16r1, October 1999.
[35] J. Xin, C.-W. Lin, M.-T. Sun, Digital video transcoding, Proc. IEEE 93 (1) (2005) 84–97.