Video Streaming over an Ad Hoc Network using Data Partitioning and Path Diversity

Ismail Ali, Sandro Moiron, Martin Fleury, and Mohammed Ghanbari
University of Essex, United Kingdom

Abstract — *Multiple description video coding can exploit path diversity to improve error resiliency by generating several independently decodable video descriptions and sending them over different paths. Such paths are readily available in ad hoc wireless networks. To approach the original quality, all descriptions need to be successfully received. However, in a wireless channel, error bursts or link disruption may still affect one or more descriptions, resulting in significant drops in video quality. This article describes a lower complexity yet effective scheme for video streaming over such wireless networks. In order to improve error resiliency, the scheme combines unequal packet importance with the advantages of path diversity. The scheme can be implemented with a minimal increase in computational complexity, as spatial decomposition of video frames is employed to generate the descriptions. Experimental results for an ad hoc wireless network show that a significant objective quality gain can be achieved (up to 3 dB) in the presence of signal shadow fading.*

Index Terms — ad hoc wireless networks, error resilience, H.264/AVC, multiple-description coding, path diversity, video data partitioning.

I. INTRODUCTION

Video communication relies on data compression by video codecs, as otherwise the raw data rates would overwhelm most network links. Consequently, sending video over unreliable wireless channels is a challenging task due to the temporal data dependencies that exist within a compressed bitstream. We have considered video over wireless ad hoc networks, which can both extend the coverage of cellular networks and provide a means to offload traffic from busy cells. Though there is an ever-present risk of damaged packets, by exploiting the built-in error resilience features of the H.264/AVC (Advanced Video Coding) standard codec [1], combined with path diversity, it is possible to build upon embedded physical-layer forward error correction (FEC), when it is present. In turn, higher transmission rates become possible. This article proposes a scheme that exploits path diversity to provide better protection for the most important video packets. H.264/AVC Flexible Macroblock Ordering (FMO) [1] is adopted for the path diversity scheme. Furthermore, FMO together with data partitioning results in a relatively low complexity form of error resiliency. Because retransmission is not required, the scheme is most applicable to conversational video applications such as video phone, which use two-way video streaming.

In standard hybrid codecs, such as H.264/AVC, the basic unit of compression is the macroblock (MB), with a video frame being divided into rows of MBs. By default, processing of the MBs takes place in a raster scan order. However, FMO enables arbitrary MB grouping into individually decodable slices and is present in the Baseline Profile of H.264/AVC, with consequently a number of hardware designs being available. In other words, MBs under FMO need not be processed in raster scan order but can be selected according to any desirable pattern. This tool, FMO, can generate spatial descriptions of the video, described in Section II.A. Data partitioning, on the other hand, allows the packetization of the bitstream into three different partitions: A, B, and C of decreasing order of importance for decoder reconstruction purposes. Data partition-A (DP-A) contains the most important information for error concealment such as motion information. Partitions B and C mainly carry the transform coefficients of the intra- and inter-coded blocks respectively. If DP-A is lost, motion vectors can be estimated from the scaled motion vectors of equivalent MBs in the last reference video frame or anchor point. When data partitioning is combined with FMO, as used in this paper, further error concealment possibilities exist (refer to Section III), though DP-A remains just as important for error concealment purposes. Indeed, if partition B or partition C is lost, irrespective of whether FMO is used, the pixels can be concealed by replacing with pixels identified from the previous frame, using the motion vectors of DP-A. Thus, DP-A packets require higher protection for delivery of higher-quality video.

A combination of hierarchical modulation and forward error protection (application-layer Turbo coding) was employed in [2] to give greater protection to DP-A packets, while in [1] DP-A packets were simply duplicated (or triplicated). However, the method of [2] requires modification of the wireless system (terrestrial Digital Video Broadcasting) and neither of the techniques in [1] [2] can cope with link disruption or long error bursts. To address that problem, we have applied Multiple Description Coding (MDC) [3] to take advantage of path diversity. In this article, MDC is achieved by spatial decomposition of video pictures by means of dispersed mode FMO, as further explained in Section III. This article also describes how path diversity can be exploited to provide better protection to redundant DP-A packets. Apart from path diversity in an ad-hoc wireless network, the approach is also suitable for video streaming from multi-homed devices [4], in which packets are routed over different access networks. This allows bandwidth aggregation, with the added advantage from the scheme of immunity to error bursts occurring on one of the paths.

Layered video streaming is the main alternative to MDC. A specialized codec outputs a coarsely quantized version of the video, the base layer, which must be received for decoding to take place. Optional enhancement layers, formed by finer quantization of the coding residue, allow the received video quality to be scaled up. In [5], transfer of a base layer and one or more enhancement layers over multi-paths was combined with Automatic Repeat reQuest (ARQ). Unlike MDC, if the base layer is not received correctly, in layered video, the decoder cannot reconstruct the original video. By assuming that the display deadline is twice the round-trip time, it is possible to send one ARQ to protect the base layer. Therefore, that scheme assumes sufficient play-out time and bandwidth to allow ARQs. Sending ARQs will also cause more control packet overhead, which can be high in an ad hoc network. In a more general context, the research in [6] concluded that layered video is competitive with MDC if the video sending rate is optimized according to its impact on video quality at the display. However, such rate-distortion analysis is computationally intensive [1] and unlikely to be employed for video streaming to mobile devices. In mobile devices with a limitation in battery power and/or processor computational power, simplicity is required. Therefore, we have applied a simpler FMO-based MDC scheme in combination with duplication of DP-A packets, which are sent along different paths. As opposed to the normal procedure, duplicate packets are sent over a different path to the matching description. As dispersed mode FMO can provide up to eight alternative descriptions, eight alternative paths could be chosen. However, with an increase in the number of paths, non-adjacent MBs need to be used to conceal corrupted or missing video data at the decoder.

The advantage of the proposed approach is mainly in its implementation simplicity and compatibility with an existing standard codec. Most previous work on MDC combined with multipath delivery, or layered coding with unequal error protection, has employed more complex and possibly more refined methods of MDC or layered coding. In contrast, the proposed work employs standard compatible FMO and data partitioning, making it more feasible for mobile applications. For example, the work in [7] on MDC was chiefly concerned with finding paths that result in a minimum of packet-drops. The paths selected were node-disjoint, i.e. did not pass through the same node. This involved creating a conflict graph of interference between wireless links to provide a heuristic approximation to an NP-hard problem. Our approach exploits existing tools specified in the H.264/AVC codec and does not involve complex optimizations. Research-orientated approaches aim to present a choice of options for future requirements. Our approach more directly meets the needs of commerce, which must develop solutions that reduce the time to market. In general, our work may lead to recognition that there is a place for both types of approach. A further aspect to this approach is that we also determine the computational complexity, as that is important when targeting constrained mobile platforms.

The remainder of this article is organized as follows. Section II briefly introduces readers to FMO and data-partitioning. Section III presents the scheme described by this article, while Section IV evaluates the performance for an ad-hoc network and evaluates the scheme’s computational and coding overheads. Finally, Section V summarizes this article.

II. H.264/AVC CONFIGURATION

Recent video codecs target high compression efficiency in order to alleviate the network load and allow video delivery over bandwidth constrained networks. However, as the compression efficiency increases, error sensitivity increases as well. As a result, a single transmission error can propagate over time to multiple frames until there is an intra-refresh reset. To achieve such a reset, the entire frame can be coded without reference to previous video frames. However, to avoid loss of coding efficiency, intra-refresh must be used sparingly. We now consider the error resiliency methods we have selected from H.264/AVC's set of tools.

A. Slicing and FMO

The simplest form of error resiliency is through slicing. When errors occur, the affected slices are discarded and all contained data is lost. The bigger the slice, the bigger is the potential loss of data. Therefore, the compressed picture data is often split into a number of slices each consisting of a set of MBs. Each slice is independently decodable, thus acting as a resynchronization unit that prevents spatial error propagation to the entire picture.

Additionally, the H.264/AVC introduced FMO error resiliency to create slices. FMO organizes the way MBs are assigned to a slice group with a maximum of eight slice groups per frame. The H.264/AVC JM reference software implements, amongst others, an FMO dispersed mode (otherwise known as checkerboard) that generates two or more slice groups, whereby MBs are packed into each group based on their even or odd positions (when just two groups are used). This pattern is particularly useful to improve the performance of error concealment, as when one slice is lost, there are up to four spatially adjacent MBs available from the other slice group to aid the reconstruction of the lost MBs. Each of the FMO slices can form an MDC description. Fig. 1 illustrates the spatial decomposition applied to the *Paris* test sequence (cf. results in [1]) employed in our tests. *Paris* exhibits some spatial coding complexity resulting from the bookcase background, while small motions of the objects held by the presenters need to be temporally coded.

Fig. 1. Multiple Descriptions using FMO with a dispersed mode pattern.

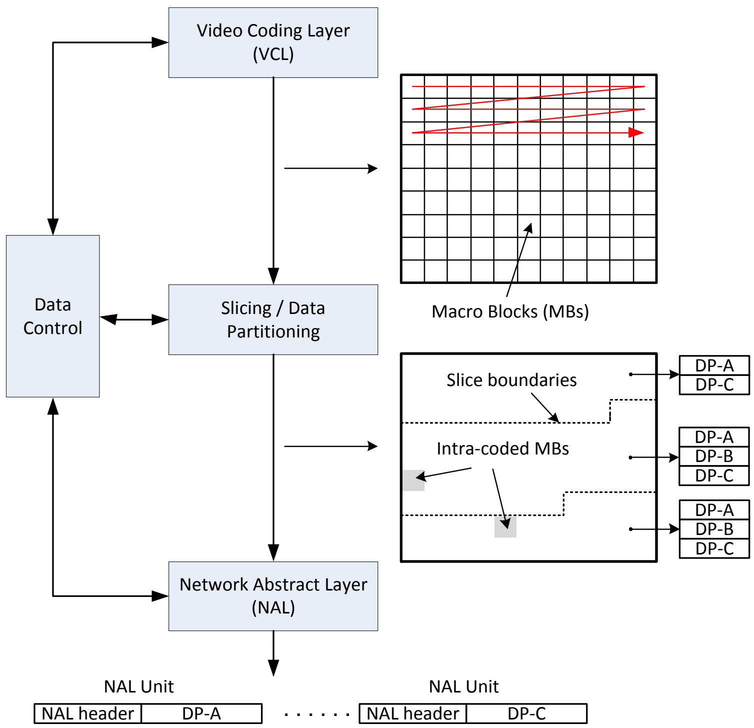

Fig. 2. H.264/AVC data flow for combined slicing with data-partitioning.

B. Data partitioning

An H.264/AVC codec conceptually separates the Video Coding Layer (VCL) from the Network Abstraction Layer (NAL). The compressed content arriving from the VCL layer is encapsulated into NAL units (NALUs). In a codec implementation, this dataflow can be managed by a data control unit, as Fig. 2 illustrates. (Fig. 2 also shows the processing of a frame's MBs in raster scan order.) A NALU consists of a one-byte header followed by the corresponding payload information. In H.264/AVC Real-time Transport Protocol (RTP) output mode, each NALU is encapsulated in an RTP packet prior to output (not shown in

Fig. 2). An RTP header provides a sequence number, and a timestamp for functions such as jitter compensation and media synchronization, as well as the data format but no network functionality, as this is intended to be provided by UDP/IP, with header compression of the three headers often applied to wireless communication.

Each slice (refer back to Section II.A) is by default placed within its own NALU. However, when data partitioning is enabled, every slice (including FMO dispersed mode slices) is divided into up to three separate partitions and packed into separate NALUs. For ease of representation, this process is illustrated in Fig. 2 for geometrical slicing, but partitioning can equally be applied to slices created by a dispersed mode pattern of MB slice selection, as in Fig. 1. The end result of employing the data flow of Fig. 2 is that compressed data are both partitioned according to reconstruction importance and divided into slices in such a way that path diversity can be exploited.

DP-A comprises the most important information of the slice, including the MB addresses and types, together with their motion vectors and quantization parameters. Partition B consists of transform coefficients from intra-coded (spatially predicted) MBs (see Fig. 2), along with Coded Block Patterns (CBP), a compact map indicating which blocks within each MB contains non-zero coefficients. As intra-coded MBs can be coded in a number of different prediction patterns, these per-MB intra prediction modes are specified in DP-A as the MB type. Natural intra-coded MBs will occur in an otherwise predictively-coded frame, as an encoder will select such MBs if there is no suitable MB in a reference frame upon which to base a prediction. In fact, an encoder could conceivably select more intra-coded MBs than was practically required if its bit budget enabled it, though in the extreme case there must be at least one inter-coded MB in a predictive coded frame. Turning to partition C, this consists of transform coefficients from inter coded (temporally predicted) MBs, their CBPs, with inter prediction modes specified in DP-A as the MB type. Thus, the residual video data after temporal prediction is transformed into a frequency space as a form of de-correlation, after which quantization occurs to form partition C’s transform coefficients.

In order to decode partition-B and -C, the decoder must know the location from which each MB was predicted, which implies that partitions B and C cannot be reconstructed if DP-A is lost. Though DP-A is independent of partitions B and C, Constrained Intra Prediction (CIP) should be set [8] to make partition-B independent of partition-C. By setting this option, partition-B MBs can no longer be predicted from neighboring inter-coded MBs, the prediction residuals of which reside in partition-C, implying some loss of coding gain. Thus, with CIP set, partition-B MBs can only be intra-coded by spatial prediction from other intra-coded MBs, which, as these MBs are not usually spatially adjacent, will not be well correlated. For source coding reasons detailed in [8], even with CIP set, partition-C cannot be made independent of partition-B for the profile structure of H.264/AVC. There is a small coding overhead from employing data-partitioning, though the results that follow show that the gain from using it can considerably outweigh data partitioning's absence.

To assess the effect of dropping of different data partitions on video quality, the *Paris* sequence was encoded with data partitioning enabled at a constant target bitrate of 1 Mbps. Different percentages of data from each partition were individually dropped and the sequence was reconstructed from the remaining parts. The resulting objective video quality (PSNR) is shown in Fig. 3. (Plots in this Figure and elsewhere include error bars of one standard deviation.) For the same percentage of video data loss rate, it is easy to observe a severe adverse effect on video quality when DP-A is lost. Therefore, this partition is given the highest protection level.

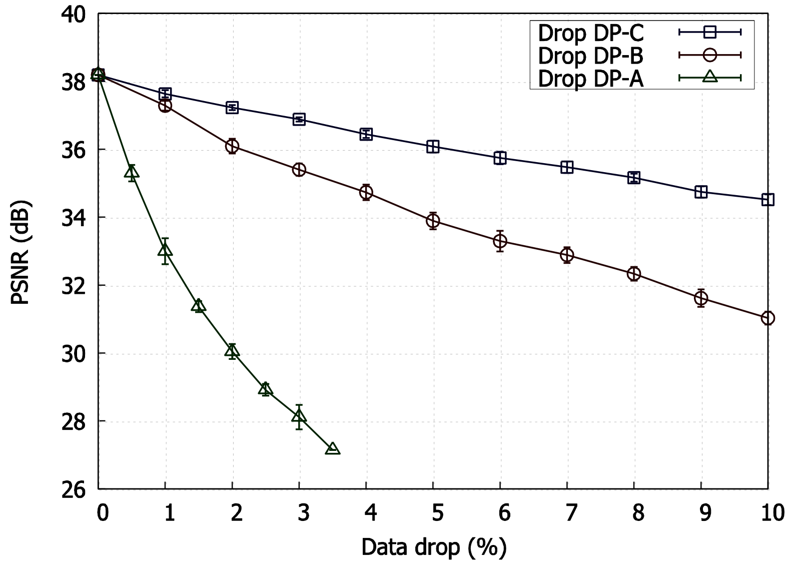

Fig. 3. Video quality when dropping different data partitions for the *Paris* sequence.

As previously mentioned, DP-A represents motion vectors and MB addresses, which are almost independent of the quantization parameter (QP) governing the extent of compression arising at the quantization stage of the codec. However, changing the QP will alter bits assigned to partitions B and C, and, hence, the percentage of bits in each frame assigned to DP-A can also vary with QP. Assuming a PSNR of around 40 dB to be an acceptable broadcast video quality, then based on encoding the test sequences *Bus*, *Mobile*, *Paris*, and *Stefan* with varying degrees of spatial and motion coding complexity, the corresponding QP is about 24 (from a range 0 to 50) for most sequences. For the same set of video sequences, it was found that a QP of 24 corresponds to no more than a 20% bitrate share for DP-A.

III. Proposed Method

MDC is an efficient error resilient video coding technique for video transmission over error-prone networks. Its basic concept is the splitting of video into two or more descriptions such that the video contents can be sent over multiple paths. As descriptions are independently decodable, the video can still be decoded, even with complete link disruption in one path. However, in order to achieve the original video quality, all descriptions need to be received. As this is not always possible, error concealment [1] can be used to recover the lost slices based on the received information. In H.264/AVC, lost intra-coded MBs are recovered by pixel-wise spatial interpolation from MB pixels bordering a lost MB. For inter-coded MBs, the ‘Frame Copy’ error concealment technique simply copies the lost slice from a collocated slice in a previous frame. However, we selected H.264/AVC’s ‘Motion Copy’, which goes one step further by estimating the motion activity of MBs in the lost slice based on the available motion information. The quality of this technique depends on the availability of motion vectors neighboring a lost MB. Motion activity of the lost blocks is estimated from successfully decoded neighboring blocks, by selecting the motion vector of the block that results in the smallest luminance change across block boundaries. Using FMO’s dispersed mode, as previously mentioned, a lost MB belonging to one slice group will be surrounded by safely received MBs from the other group, and, thus, motion copy works well. In the H.264/AVC codec, neighboring MBs can be subdivided into blocks in a variety of ways or modes. Therefore, the motion vectors of any sub-divided 8×8 pixel blocks need to be consolidated as one motion vector. This motion vector is formed as the average of the sub-block vectors. However, since the estimated motion vector might not accurately represent the original motion activity, a lower quality than the original can be expected. When DP-A packets are received, motion vector estimation is not needed, as the original motion vectors are available. Thus, better quality concealment is achieved.

Duplicating DP-A packets along the same path is not always effective in the presence of long error bursts. When the original packets are lost, the duplicated packets will probably be lost as well. As a result, we examine an alternative technique: packets containing motion information (DP-A packets) of each description are duplicated and sent over the paths of the other description.

This article analyzes the scenario where two spatial descriptions are created and sent over two different paths. Each video description is created by spatially splitting the H.264/AVC stream using FMO with dispersed mode. Data partitioning was also enabled to packetize the bitstream into data partitions A, B and C in order to allow protection of the most important video information (residing in DP-A). DP-A packets of each description are duplicated and sent along the path of the other description to increase their chance of being received when one path is impaired.

IV. RESULTS

We now present simulation results for an ad hoc network. As a point of comparison, two variants of the scheme are compared with the proposed one. Subsequently, the computational cost and coding overhead are evaluated.

A. Simulation model

In order to assess the performance of the proposed method, four sets of experiments were performed, which are illustrated in Fig. 4 as an aid to the following description. The test sequence *Paris* was coded in Common Intermediate Format (CIF) at 30 frame/s. The encoder was configured to use an IPPP... coding structure, that is an initial intra-coded frame followed by all predictively coded P-frames. In the first set of tests (T1), DP-A packets for each description were duplicated (Dup DP-A) and sent along the same path as their description. In the second set of tests (T2), DP-A packets for each description were also duplicated but sent along the path of the other description. For tests T1 and T2, the video was CBR encoded with a target bitrate of 1 Mbps. For the next set of tests T3, the descriptions were sent without duplicating DP-A packets. To achieve a fair comparison, the video of test T3 was coded at a higher quality with a target bitrate equivalent to the bitrate of the duplicated DP-A packets plus the original 1 Mbps bitrate. Measurement showed that, in tests T1 and T2, the bitrates of partitions A, B, and C were in the percentages 19%, 31%, and 50% respectively. Notice also that in T3 the data were partitioned for comparison purposes even though this is not strictly necessary, as there is no preference for any partition. In tests T1, T2 and T3, a single (a row of 22 MBs for CIF) cyclic intra-refresh line [9] was employed to mitigate temporal error propagation, which would otherwise occur due to the absence of intra-coded frames after the first frame. Finally, to establish that the measures taken really did improve the video quality, a fourth test labeled T4 was conducted. For this test, just 11 forced intra-coded MBs inserted randomly per frame. The video was then CBR encoded with a target bitrate of 1 Mbps. FMO was not applied, resulting in a single slice per frame. This slice was then broken up into three packets, one packet for each of the partition types. In the case of data-partitions B and C, video packets were divided evenly between the two paths. In addition to the 1 Mbps CBR stream, DP-A packets were duplicated so that a copy of each DP-A packet was sent over each path as a simple form of FEC.

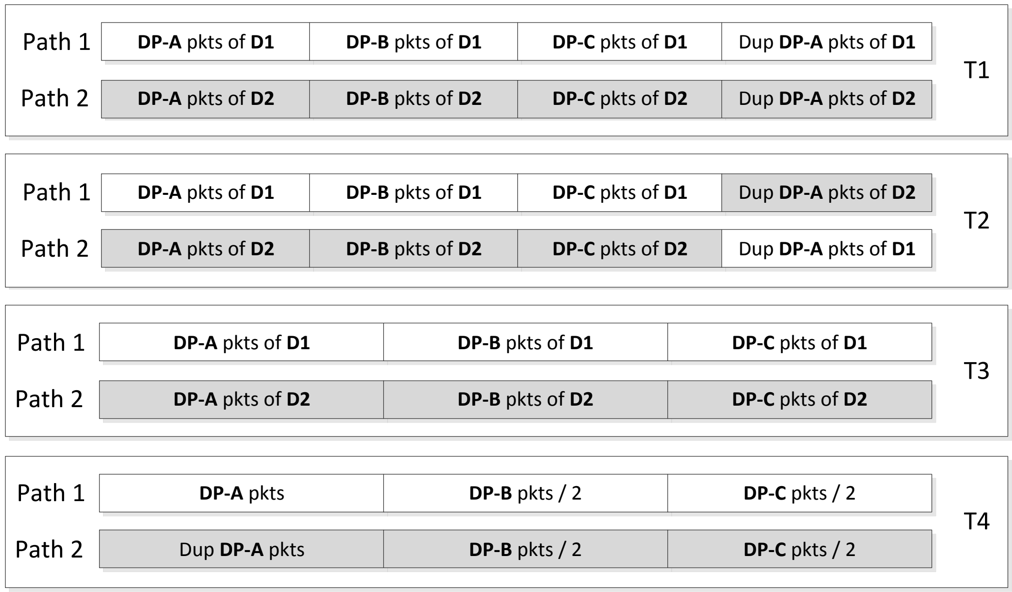

Fig. 4. The four experimental tests conducted.

The test network scenario was as follows. A video source sent the two video descriptions over two disjoint paths, each comprising of two wireless devices before the video sink. Thus, the video descriptions each pass over three wireless links each of distance 116 m. All devices were also 116 m from each other.

The simulations were carried out by means of the well-known network simulator ns-2. The simulator allows the impact of wireless channel generated errors to be modeled. Dropped packets as reported in the output file from the simulator were later removed from the original input packet stream to create the resulting erroneous received stream. The received stream was then decoded by the codec to assess the video quality at the receiver side. For ease of modeling, the nodes were equipped with 802.11b radios with a data rate of 11 Mbps. For the wireless channel, the widely adopted shadowing propagation model [10] was employed. In the simulations, the settings matched an outdoor scenario according to in-the-field measurements given in [10].

B. Resulting video quality

For each set of tests, 400 runs were conducted. The average of the runs and standard deviations of the PSNR were calculated and plotted in Fig. 5 versus the percentage video data drop. To generate increased video data losses, the path loss exponent parameter of the shadowing channel model [10] embedded within the simulation was increased. Thus, all runs with a given data loss, generated by application of the channel model were grouped into percentage loss classes, e.g. centered at 5, 7, 9, 11%, and plotted against the mean PSNR with that loss percentage. For the first set of tests T1, if there is a packet loss in one path, that path is likely to be experiencing a packet loss burst. Therefore, transmitting the duplicated DP-A packet over the same path is unlikely to be successful. Thus this method does not provide good protection against burst errors. For the second test T2, as the loss patterns of the paths are *not* correlated, transmitting the duplicate DP-A packets using the other path could have a higher success probability. In this case, more DP-A packets will arrive, resulting in better video quality.

Comparing with test T3, Fig. 5 shows that at lower percentage drops (good channel conditions) and for the same available bitrate, it is better to spend the excess bitrate to get better video quality instead of sending duplicated DP-A packets. However, when the percentage drop rate increases due to poor channel conditions, this is not the case. In such conditions, the Figure shows that this article’s scheme (T2) can achieve up to 3 dB video quality gain over other test schemes, including T4 when no FMO is employed. Such data losses may occur in deep fades but a significant advantage in video quality also occurs at more typical loss rates between 5 and 10%.

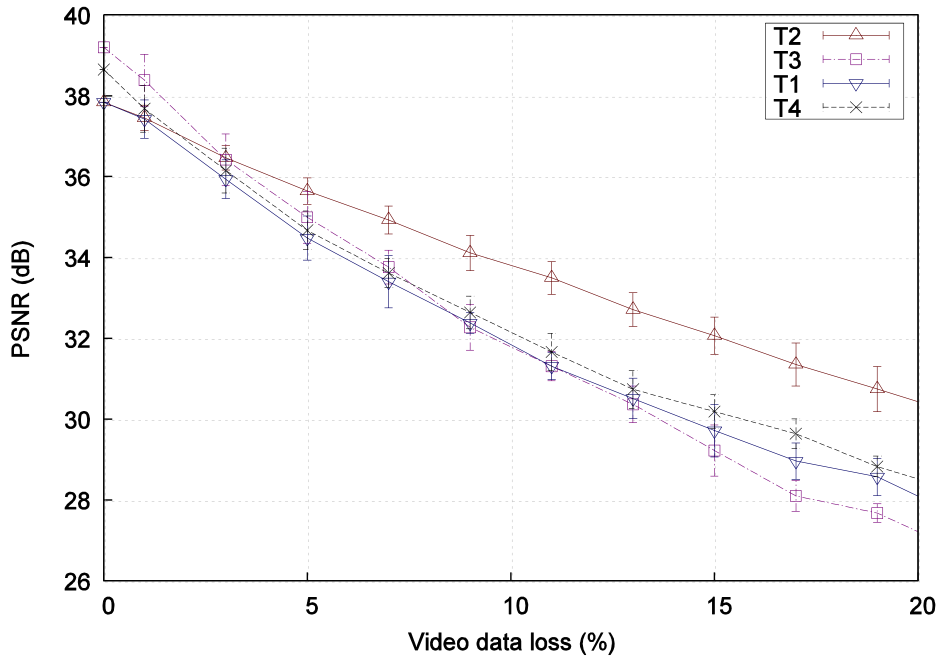

Fig. 5. Mean video quality (PSNR) versus percentage video data drop for the *Paris* sequence.

C. Computational Complexity

When targeting an application scenario in which the devices might have battery and processing power constraints, it is mandatory to determine the computational complexity involved in the proposed scheme. The average time required to code a frame can act as a measure of complexity. Additionally, the bitrate overhead of the proposed technique should be assessed.

Fig. 6 shows the time to code 100 frames of the *Paris* test sequence using different error resiliency tools. The standard JM 15.1 H.264/AVC reference software was again selected and the encoding time is plotted versus the bitrate. The first bitstream was encoded without any error resiliency technique (NoRes) using the IPPP… coding structure. For the second test, cyclic intra refresh lines (IR) were used, the third added data partitioning and FMO's dispersed mode (IR+DP+FMO). For comparison purposes, the sequence was also coded with periodic I-frames inserted every 18 frames, a refresh rate that is equivalent to employing a single intra-refresh line per frame for CIF resolution. Context-Adaptive Binary Arithmetic Coding (CABAC) or Context-Adaptive Variable-Length Coding (CAVLC) is the final source encoding stage, which exploits statistical redundancy in the bitstream. For all the encodings, CABAC was disabled and the less computationally demanding CAVLC was used instead, as is normal for real-time operation on mobile devices. The Figure shows that the encoding time is significantly affected by the bitrate used for encoding rather than the encoding scheme itself. Additionally, it can be seen that the time to code the bitstream without any error resiliency is the highest. The main reason for this is that when intra refresh lines (or I-frames) are used, there is one less MB line per frame (or one less frame per Group of Pictures (GOP)) to be encoded with inter prediction. Therefore, for that row the complex motion estimation process is skipped. Adding data partitioning and FMO on top of IR does not add a noticeable overhead in terms of encoding time.

The compression overhead in scheme T2 is shown in Fig. 7. FMO adds some overhead, due to the broken in-picture prediction mechanisms, which tends to be higher for active sequences. On the other hand, the inclusion of intra-refresh lines adds a considerable overhead because of the less efficient intra (spatial) prediction when compared with inter (temporal) prediction. However, it is essential to include some sort of intra update in the bitstream to limit temporal error propagation. It can be seen that scheme T2 introduces a slightly higher overhead than the inclusion of I-frames at an equivalent refresh rate.

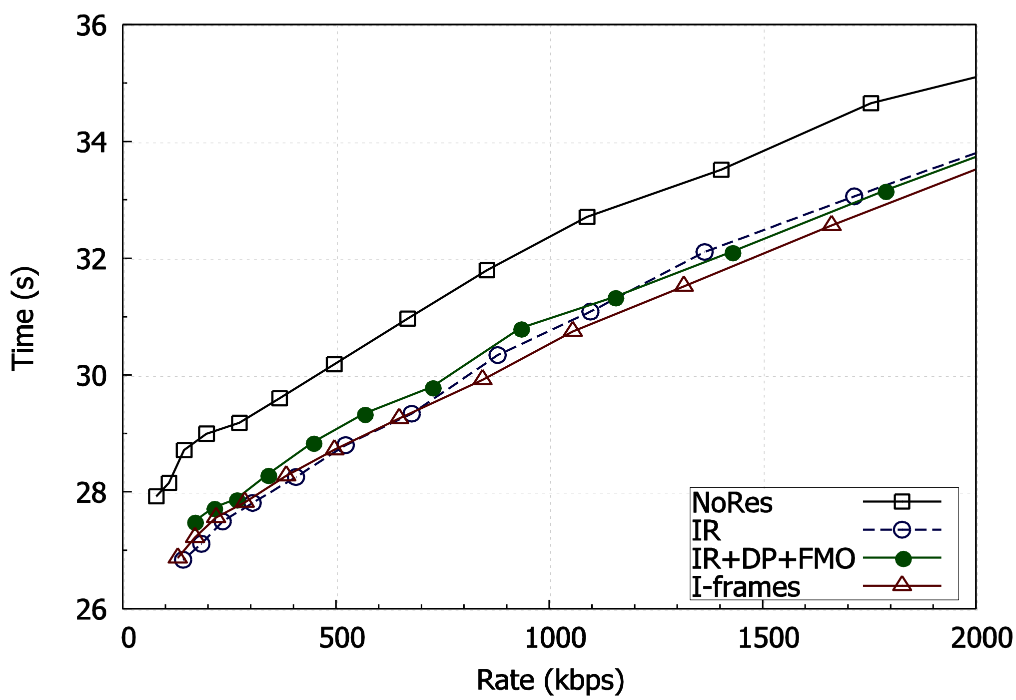

Fig. 6. Time to encode 100 frames of *Paris* CIF sequence using various schemes.

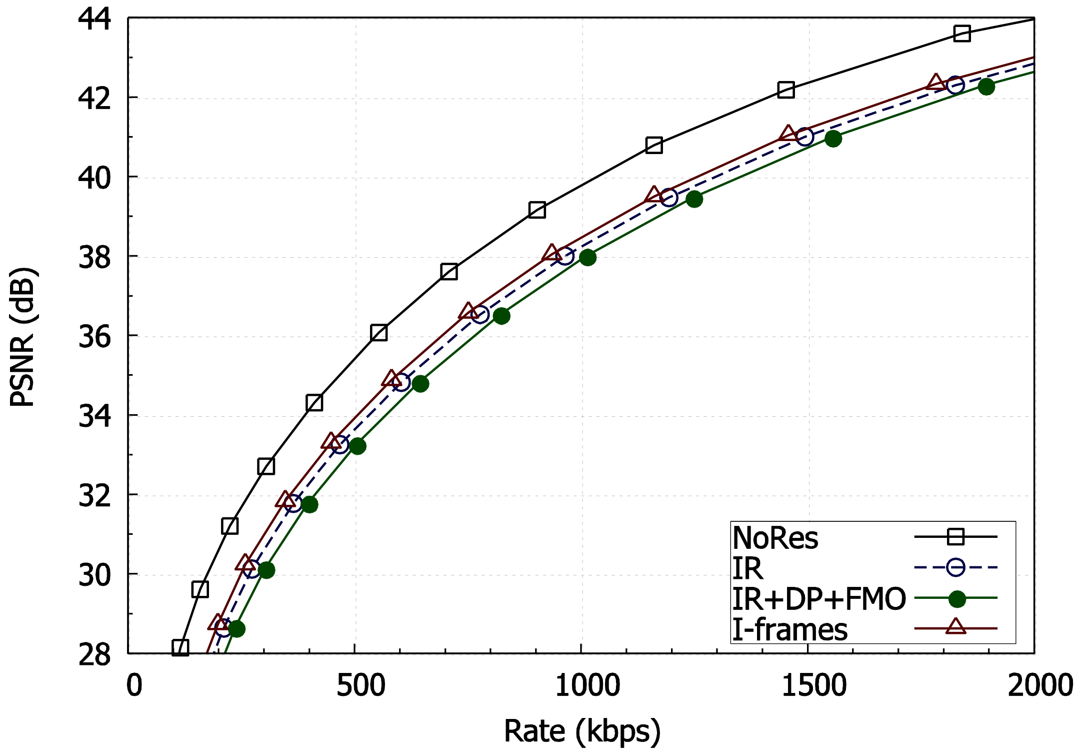

Fig. 7. Rate-Distortion curves showing the video quality penalty due to the technique overhead for the *Paris* sequence.

V. Conclusion

This article presented an improved error resiliency technique for delivering MDC video over 'lossy' wireless channels, whereby partition A data was sent over a different path to its matching description. The scheme with data-partitioning can achieve an increased error resiliency against packet drops. Although the scheme introduces a quality penalty at low packet loss rates in comparison to MDC without duplication, for higher video data loss rates (> 3 %), experimental results show that the proposed scheme is far superior, achieving a quality gain of up to 3 dB. Spatial decomposition through FMO is present in H.264/AVC's Baseline profile and, hence, is present in hardware codecs. The main current limitation to widespread application of the technique is that data-partitioning in the Extended profile is not as yet implemented in hardware codecs. However, there is reason to believe that a real-time codec with this feature will become available. In fact, a more complex scalable software codec is now implemented for tablets and smartphones, as part of a scheme to extend video conferencing to these devices. Hence, using the technique described in this article, two-way video streaming has become possible across wireless ad hoc networks formed from such devices.

REFERENCES

[1] S. Wenger, "H.264/AVC over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 645-656, 2003.

[2] B. Barmada, M. M. Ghandi, E. V. Jones, and M. Ghanbari, "Prioritized transmission of data partitioned H.264 video with hierarchical QAM," *IEEE Signal Proc. Letters*, vol. 12, no.8, pp. 577-580, Aug. 2005.

[3] Y. Wang, A. R. Reibman, and S. Lee, "Multiple description coding for video delivery", *Proc. of the IEEE*, vol. 93, no. 1, pp. 57-70, 2005.

[4] J. Liao, J. Wang, X. Zhu, and T. Li, "Exploiting path diversity to enhance aggregating throughput for multi-homed wireless devices," *IEEE Trans. Consumer Electronics*, vol. 56, no. 2, pp. 613-619, 2010.

[5] S. Mao, S. Lin, S.S. Panwar, and Y. Wang, "Reliable transmission of video over ad-hoc networks using automatic repeat request and multi-path transport," in *IEEE Vehicular Technology Conf.*, Oct. 2001, pp. 615-619.

[6] J. Chakereseki, S. Han, and B. Girod, "Layered coding vs. multiple descriptions for video streaming over multiple paths," *Multimedia Systems*, vol. 10, no. 4, pp. 275-185, 2005.

[7] W. Wei and A. Zakhor, "Interference aware multipath selection for video streaming in wireless ad hoc networks," *IEEE Trans. Circuits Syst. Video Technol*., vol.19, no.2, pp.165-178, 2009.

[8] Y. Dhondt, S. Mys, K. Kermeirsch, and R. van de Walle, "Constrained inter prediction: Removing dependencies between different data partitions," in *Advanced Concepts for Intelligent Visual Systems*, 2007, pp. 720-731.

[9] R. Schreier, and A. Rothermel, “Motion adaptive intra refresh for the H.264 video coding standard,” *IEEE Trans. Consumer Electronics*, vol. 52, no. 1, pp. 249-253, 2006.

[10] T.S. Rappaport, *Wireless Communications*: *Principles and Practice*, 2nd ed., Prentice Hall, Upper Saddle River, NJ, 2009.