

ALP: Adaptive Loss Protection Scheme with Constant Overhead for Interactive Video Applications

KIANA CALAGARI and MOHAMMAD REZA PAKRAVAN, Sharif University of Technology, Iran
 SHERVIN SHIRMOHAMMADI, University of Tehran, Iran and University of Ottawa, Canada
 MOHAMED HEFEEDA, Simon Fraser University, Canada

There has been an increasing demand for interactive video transmission over the Internet for applications such as video conferencing, video calls, and telepresence applications. These applications are increasingly moving towards providing High Definition (HD) video quality to users. A key challenge in these applications is to preserve the quality of video when it is transported over best-effort networks that do not guarantee lossless transport of video packets. In such conditions, it is important to protect the transmitted video by using intelligent and adaptive protection schemes. Applications such as HD video conferencing require live interaction among participants, which limits the overall delay the system can tolerate. Therefore, the protection scheme should add little or no extra delay to video transport. We propose a novel Adaptive Loss Protection (ALP) scheme for interactive HD video applications such as video conferencing and video chats. This scheme adds negligible delay to the transmission process and is shown to achieve better quality than other schemes in lossy networks. The proposed ALP scheme adaptively applies four different protection modes to cope with the dynamic network conditions, which results in high video quality in all network conditions. Our ALP *scheme* consists of four protection *modes*; each of these modes utilizes a protection *method*. Two of the modes rely on the state-of-the-art protection methods, and we propose a new Integrated Loss Protection (ILP) method for the other two modes. In the ILP method we integrate three factors for distributing the protection among packets. These three factors are error propagation, region of interest and header information. In order to decide when to switch between the protection modes, a new metric is proposed based on the effectiveness of each mode in performing protection, rather than just considering network statistics such as packet loss rate. Results show that by using this metric not only the overall quality will be improved but also the variance of quality will decrease. One of the main advantages of the proposed ALP scheme is that it does not increase the bit rate overhead in poor network conditions. Our results show a significant gain in video quality, up to 3dB PSNR improvement is achieved using our scheme, compared to protecting all packets equally with the same amount of overhead.

Categories and Subject Descriptors: H.4.3 [Information Systems Applications]: Communications Applications—*Computer conferencing, teleconferencing, and videoconferencing*

General Terms: Performance, Measurement

Additional Key Words and Phrases: Adaptive video protection, interactive video, ROI, HD video, H.264/AVC, data partitioning, error propagation, slicing, FMO, FEC, ULP

Authors' addresses: K. Calagari, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, V5A 1S6; email: kcalagar@sfu.ca; M. R. Pakravan, Department of Electrical Engineering, Sharif University of Technology, Azadi street, Tehran, Iran; email: pakravan@sharif.edu; S. Shirmohammadi, Multimedia Processing Laboratory, School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Iran; email: sshirmohammadi@ut.ac.ir, and Distributed and Collaborative Virtual Environments Research Laboratory (DISCOVER Lab), School of Electrical Engineering and Computer Science, University of Ottawa, Canada; email: shervin@eecs.uottawa.ca; M. Hefeeda, School of Computing Science, Simon Fraser University, Burnaby, BC, Canada, V5A 1S6; email: mhefeeda@cs.sfu.ca.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

© 2014 ACM 1551-6857/2014/12-ART25 \$15.00

DOI: <http://dx.doi.org/10.1145/2656203>

ACM Reference Format:

Kiana Calagari, Mohammad Reza Pakravan, Shervin Shirmohammadi, and Mohamed Hefeeda. 2014. ALP: Adaptive loss protection scheme with constant overhead for interactive video applications. *ACM Trans. Multimedia Comput. Commun. Appl.* 11, 2, Article 25 (December 2014), 23 pages. DOI: <http://dx.doi.org/10.1145/2656203>

1. INTRODUCTION

Due to the steady rise of bandwidth available for Internet users, the demand for high quality and high resolution video in interactive video applications such as video calls, video conferences and video chats has considerably increased. As a result, in recent years most service providers of such applications are upgrading their services to High Definition (HD) video qualities [Skype][Lync]¹. Despite this increase in available bandwidth, still most users are using best-effort networks (such as the Internet), which do not provide any guarantee for packet delivery and quality of service, and thus events such as congestion can and do occur, which result in packet losses and degradation in the quality. Therefore, for interactive applications such as HD video chat and video conferencing, where buffering and retransmission are not possible, video packets must be protected against losses. It should be noted that an efficient protection scheme becomes even more important for HD videos since customers have high expectations and so quality degradations caused by loss will have an even higher negative influence on their quality of experience [Javdtalab et al. 2011]. On the other hand, given the high bit rates of HD videos, even a small increase in the percentage of the added redundancy can have a considerable impact on the bandwidth demand. As such, designing protection schemes for interactive HD video applications remains a significant challenge.

A variety of video protection schemes exist [Wu et al. 2000; Wang and Zhu 1998]. Traditionally, to protect any transmitted real-time data and not just video, for each block of packets, some redundant packets are generated and sent with the original data. If an original packet is lost, its information can be retrieved using the redundant packets. But even with this redundancy approach, there are still challenges specifically for interactive applications. When sending a block of packets, the redundant packets will be sent after the last original packet of that block. However, since the application is delay sensitive, the redundant data will be of no use if it does not arrive on time. In an interactive video application, delays more than 100ms are usually noticeable and considering that long distance Internet connections can have delays higher than 100ms already, there is a need for a protection scheme that does not add any extra delay. Trying to protect each packet individually is also not a good option since it adds a considerable amount of overhead. For example, if we add one redundant packet to each original packet, the bit rate will be doubled while the protection is still low since it still cannot resist a burst loss of 2 consecutive packets.

For a more efficient use of bandwidth and redundant packets, Unequal Loss Protection (ULP) methods have been proposed, in which some packets are protected more than others based on their importance. The most basic question in unequal loss protection methods is the definition of importance and the amount of importance assigned to different parts of data. In addition, as we shall see in the related work section, many schemes have been developed specifically for video protection. Each of these, however, is suitable for a specific situation and/or network conditions.

To have a practical and efficient system, and to increase the effectiveness of redundant packets, one can make use of multiple protection modes and dynamically switch among them, based on the network conditions. Doing so entails answering two questions: (1) which protection mode to use in each of the network conditions? and (2) exactly

¹Skype. <http://www.skype.com/>. Microsoft Lync. Microsoft Lync Conference 2014. <http://www.lyncconf.com/>.

when to switch between different protection modes? Answering these two questions is the goal of this article. Specifically, we propose a novel Adaptive Loss Protection (ALP) scheme that adds negligible delay to the transmission process and therefore is suitable for delay sensitive and interactive video applications. The proposed ALP scheme adaptively applies different protection modes to cope with the dynamic network conditions. The ALP *scheme* consists of four protection *modes*, each of these modes utilizes a protection *method*. Two of the modes rely on the state-of-the-art protection methods, while the other two rely on a new protection method that we propose in this article. The main contributions of this article can be summarized as follows:

(1) A novel video protection method for high packet loss conditions that makes effective use of unequal loss protection and applies more protection to the important information. We name this method Integrated Loss Protection (ILP) since it integrates the use of different factors such as error propagation, Region Of Interest (ROI), slice headers, and motion vectors, for classifying the important information.

(2) A general and adaptive loss protection scheme, which is composed of multiple protection modes. The main advantage of this proposed scheme is that unlike most other adaptive schemes, it does not increase the amount of overhead when facing adverse network conditions.

(3) A new metric for dynamically switching among the protection modes. This metric is based on the percentage of unrecovered lost video packets, and provides a systematic way to decide when to switch among different protection modes, in order to maximize the video quality in all network conditions. This metric enables us to reduce the variance in video quality. The less the variance in video quality, the more we can be sure of the actual video quality and the higher the improvement gained by switching. Using this novel metric, we examine the efficiency of each protection mode based on how successful it is in recovering lost packets. We show the benefits of this metric over using other metrics such as packet loss rate or average burst loss length, which are based on network statistics.

In addition, we conduct extensive simulation study with multiple videos. Our results show that the ALP scheme: (i) achieves up to 3dB PSNR improvement in video quality compared to previous methods, (ii) achieves up to 3dB PSNR improvement compared to its own individual protection modes, and (iii) reduces the standard deviation in quality by 0.7 dB compared to when a single protection mode is used.

The rest of this article is organized as follows: We discuss the background and related work in Section 2. We present our proposed ALP scheme in Section 3, followed by a full evaluation of different parts of our scheme in Section 4. We then conclude in Section 5.

2. BACKGROUND AND RELATED WORK

2.1. Background: Video Coding, Protection and Transmission

A detailed background description is provided in the online appendix. In short, a coded video consists of several independent Group Of Pictures (GOP). A GOP begins with an I-frame, followed by several P and B-frames. Each frame can be divided into several independent slices using the Flexible Macro block Ordering (FMO) tool [Wiegand et al. 2003]. In addition, each slice can be divided into 3 partitions (Header, Intra and Inter partitions) using the data partitioning tool. Each partition is encapsulated in a separate packet. In real-time applications, packets are then protected by Forward Error Correction (FEC) techniques.

2.2. Related Work: Approaches for Protecting Data Streams

There are two main approaches to protect a stream of data from packet loss: (i) Automatic Repeat reQuest (ARQ) and (ii) FEC. In ARQ, packets are retransmitted in

case of loss. As a result ARQ is more efficient in terms of bandwidth. Since ARQ causes extra delay and jitter, it is generally not used for multimedia applications, especially interactive ones. However, partially reliable transport protocols with retransmission, such as Partially Reliable Stream Control Transmission Protocol (PR-SCTP), can be reasonable in terms of delay and jitter and therefore suitable for multimedia applications. In Sanson et al. [2010], by using PR-SCTP, different levels of protection have been applied to I, P and B frames. In Porter and Peng [2011], data partitioning is used and Header partitions are sent over TCP in order to have a reliable transmission, but other partitions are sent over UDP. Although partial reliability can make retransmission techniques suitable for video, it is still not suitable for interactive video applications that are very sensitive to delay.

The second category of protection, which uses FEC, is more suitable for improving video quality when video packets are exposed to network packet loss. This category can be divided into two subcategories. The first contains methods that try to minimize the overall distortion regardless of which parts of the frame are of more interest, and the second contains methods that apply more protection on the ROI and improve the ROI quality even if the overall distortion increases. They are described in the following.

2.2.1. FEC to Minimize Distortion. A comprehensive review on video protection schemes that use error resiliency techniques such as Multiple Description Coding (MDC) can be found in Kazemi et al. [2014]. However, to reduce the effect of distortion and error propagation, a promising approach is using ULP techniques. When using ULP, first the important parts should be somehow identified and then separated in a way that different amounts of protection can be applied to them. In Cao et al. [2013], the queue management mechanism inside the network routers is improved, in order to prioritize video packets before dropping occurs. In Xu and Hu [2013], an adaptive video transmission method is proposed, which assigns more protection to I-frames over P-frames. Video packets are protected by RS codes. When the network conditions get bad, the number of redundant packets and thus the redundancy bitrate increases.

To divide a frame into packets with different importance, slicing can be used. Using the explicit mode of flexible macro block ordering, we can assign each macro block to a different class based on its importance factor. In Thomos et al. [2006], each frame is divided into three slices and the distortion is minimized by solving an iterative optimization problem based on the distortion caused by the loss of each macro block. The answer of this optimization problem defines which slice should each macro block be assigned to, and the protection needed for each slice. In Dhondt et al. [2006], the importance of each macro block is calculated based on the number of times each pixel of that macro block is used in that frame or future frames till the end of the GOP. But since these methods try to minimize the distortion caused by error propagation using future frames, they cannot be used for real-time applications such as video conferencing, where we do not have any information about the future frames.

The method used in Zhang et al. [2009] and Zhang and Peng [2009] is suitable for real-time applications. In this method each GOP is divided into several parts and data partitioning is used. Packets are classified based on their partition and position in the GOP, and more protection is applied to earlier frames and Header partitions. However, region of interests are not considered.

2.2.2. FEC to Protect ROI. There are several other methods in the second subcategory. These methods propose different ways of extracting and protecting the ROI of the video. In Wang et al. [2010], each frame is divided into three slices, which are non-ROI, static areas in ROI, and moving areas in ROI, and more protection is given to the moving parts in ROI, since they can't be easily concealed using previous frames.

In Ciubotaru et al. [2009], various ROIs are defined, concentric around the area of maximum user interest. When exposed to congestion, the bit-rate/quality of each ROI is decreased based on its distance to the area of maximum user interest. In Fernandez et al. [2012], with the increase of network loss, the transmission rate is decreased by cropping the ROI and reducing the video quality. In Song et al. [2010], the impact of ROI enhancement for sports videos and talk shows is studied. It is shown that in talk shows, by cropping ROI or using a better quantization parameter for ROI, the overall user experience improves significantly.

In Chen et al. [2007], error propagation from the background to the ROI part is prevented by using a leaky prediction ROI coding, which prevents some ROI blocks to use non-ROI blocks as reference. In Zhong et al. [2010], a greedy algorithm is proposed for separating the region of interest and an adaptive protection is applied to the region of interest slice based on the average burst loss length of the network, which increases the protection rate as the average burst loss length increases.

In Arachchi et al. [2006], ROI is divided into two slices using a checkerboard pattern and more protection is applied to them compared to the background slice. Unlike other methods that just add more protection to the region of interest, this method triggers a new idea for improving the region of interest quality. From its results it can be seen that dividing the foreground into two slices causes a high improvement in quality compared to the same amount of redundancy applied to a one slice foreground. This improvement is due to the better concealment achieved when using the dispersed mode of flexible macro block ordering. It can be concluded that the quality improves by increasing the number of slices, but as the number of slices increase the overhead will also increase. This overhead is mostly because of the packet headers added to each slice and also because of the reduction in coding efficiency.

In the core of our proposed ALP scheme, we combine the two subcategories to achieve higher quality in real time. The results show that this combined method achieves significantly higher video quality than both categories. Our work started with Calagari et al. [2012] in which we proposed our new loss protection method under adverse network conditions and showed that different protection distributions with the same amount of overhead can each achieve a better quality in a different range of network conditions. This article extends that work by proposing a fully adaptive algorithm, which introduces an automatic way of generating different protection modes for different network conditions, and a novel metric for adaptively switching between these modes.

3. PROPOSED ADAPTIVE LOSS PROTECTION SCHEME

In this section, we first present an overview of the whole ALP scheme in Section 3.1, and then discuss the main parts of the scheme in more details in Sections 3.2 and 3.3. In Section 3.4, we analyze and discuss several practical issues for implementing ALP.

3.1. Overview

The proposed ALP scheme consists of two main components: The first component defines four different protection modes, each suitable for a different range of network conditions, but with the same amount of overhead. For the first two modes we use state-of-the-art methods, while for the third and fourth mode we propose an Integrated Loss Protection (ILP) method. Figure 1 shows the four modes and their range of effectiveness. The second component is our novel Switching Metric (SM), for adaptively switching back and forth between these protection modes. This metric helps us determine the right time to switch, in order to make it more effective in achieving high quality.

For each protection mode, a different zone is defined, which is the part of a frame that is supposed to be protected by that mode. The zone can be the whole frame, the ROI of each frame, or just the header partitions of the ROI. At the end of each GOP,

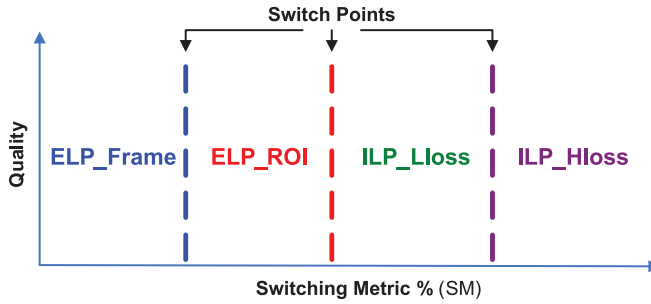


Fig. 1. The four modes of our ALP scheme and their ranges of effectiveness.

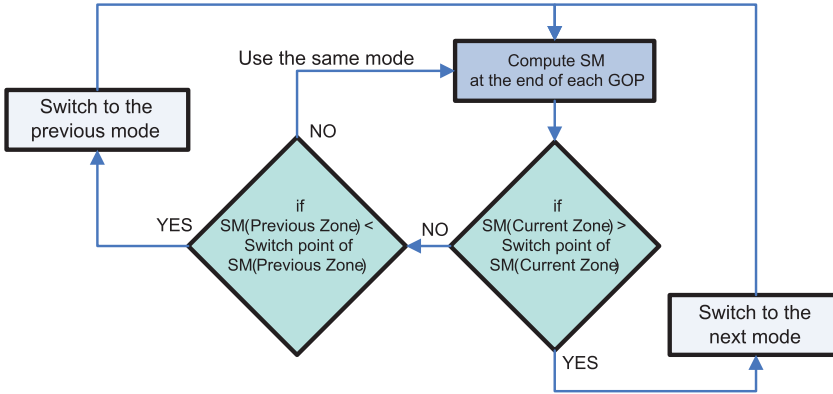


Fig. 2. Flowchart of each mode of the proposed Adaptive Loss Protection scheme.

based on the current mode, the receiver counts the number of zone packets that have been lost and the number of recovered ones throughout the whole GOP. For example, if the corresponding zone is ROI, the receiver will count all the lost and all the recovered ROI packets throughout the whole GOP. Using this information, the receiver then calculates the Switching Metric (SM) for that GOP, based on the corresponding zone. $SM(\text{zone})$ calculates the percentage of lost packets in that zone that are not recovered by the protection mode. The receiver also calculates the $SM(\text{previous zone})$. $SM(\text{previous zone})$ is the percentage of unrecovered lost packets that were supposed to be protected by the previous mode. If the switching metric crosses one of the boundaries between protection modes, the receiver sends a small feedback to the sender, ordering it to switch to the next or previous mode starting from the next GOP. It should be noted that the receiver also keeps track of the sender's protection status, both in mode and FEC configuration.

The flow chart for each mode of our proposed adaptive loss protection scheme is shown in Figure 2. Note that the ALP scheme moves between the modes one by one. This provides smooth transitions and avoids sudden and radical changes that may happen because of temporary and short network changes. Since quick and frequent transitions will harm the user experience, our scheme changes modes in case there is a real change in network conditions, not a short and temporary problem with a duration of few seconds, which would quickly recover and go back to normal. For example, if we are currently in the ELP_Frame mode, and the next GOP is suddenly exposed to a great amount of loss, we will not immediately switch to the ILP_Hloss mode. Rather, we will switch to the ILP_Lloss mode after several lossy GOPs.

The first protection mode named ELP_Frame (Equal Loss Protection for all packets in the Frame) simply applies a fixed and equal amount of protection to all packets. This mode has a good performance when the loss rate is low. In addition, this mode is used as the initial and default mode when a connection is established. In this mode, each frame is divided into multiple slices and all slices are equally protected. No ROI extraction and no data partitioning is used and all frames in the GOP are treated the same way. Slicing in frames is done using the dispersed mode of FMO to ease the concealing process. Each block of packets consists of all slices in one frame and a number of added redundant packets. The network's average burst loss length seems to be a fair choice for the number of added redundant packets in this mode. Because adding a smaller number of redundant packets means that most of the losses will not be recovered, while adding extra redundant packets will not cause much benefit and adds to the overhead. When the system is in the ELP_Frame mode, the receiver calculates the SM(Frame), which is the percentage of lost packets in the network that were not recovered in the error correction process. If SM(Frame) crosses the switch point of SM(Frame) the receiver will order a switch to the ELP_ROI mode.

In the second mode, named ELP_ROI (Equal Loss Protection for Region of Interest packets), we only protect the packets of ROI and concentrate all protection on them. This is done using the FMO feature of the coding standard [Wiegand et al. 2003]. Using the foreground/background mode of the flexible macro block ordering, we can encapsulate and send the ROI in different packets than the background. In this scenario, the ROI is extracted and both ROI and non-ROI parts are divided into slices in a dispersed manner. Only the ROI is protected, and it is protected equally throughout the GOP and no data partitioning is done. For protecting the ROI of each frame we use the same number of redundant packets that were used in ELP_Frame to protect the whole frame. But since the redundant packets are added to a smaller block of packets (just the slices from the ROI), the protection gets stronger. This happens at the expense of leaving the background without any protection. It should be noted that in both of the ELP_Frame and ELP_ROI modes where data partitioning is not needed, each slice remains as one partition during the encoding. This is important because our experiments have shown that if data partitioning is done but no extra protection is applied to the Header partition, and just the same amount of protection is applied to all partitions, the quality would degrade significantly due to the dependency that data partitioning has caused between the packets. To avoid this effect, data partitioning should not be done when there is no need for it. In the ELP_ROI mode, the receiver keeps calculating both the SM(Frame) and the SM(ROI) at the end of each GOP. SM(ROI) is the percentage of lost ROI packets that were not recovered by the ELP_ROI protection mode. If the network conditions have got better and the SM(Frame) value has decreased below its threshold, then the receiver would order the sender to switch back to the ELP_Frame mode. However, if SM(Frame) is still low, the sender will continue sending the frames using ELP_ROI unless the SM(ROI) has also increased beyond its switching point. In this case, the sender would be ordered to switch to the ILP_Lloss mode.

In the third mode, named ILP_Lloss (Integrated Loss Protection for Low loss conditions), similar to ELP_ROI, the ROI is extracted and both ROI and non-ROI parts are sliced in a dispersed manner. Also, in addition to ELP_ROI, data partitioning is used and ROI packets are classified based on the partition they contain and their positioning in the GOP. Unequal amounts of protection is applied to different classes based on their importance. In this mode the receiver calculates both the SM(ROI) and SM(ROI_HEADER). SM(ROI_HEADER) is the percentage of lost Header partitions of the ROI that were not recovered by the ILP_Lloss protection mode. If the network conditions are so severe and the SM(ROI_HEADER) increases its threshold, we switch

to the ILP_Hloss mode. While, if the network conditions have got better and SM(ROI) has decreased below its threshold, we would go back to ELP_ROI.

The fourth mode, named ILP_Hloss (Integrated Loss Protection for High loss conditions), is similar to ILP_Loss, but with a more unequal distribution of protection for saving the most important parts in severe cases. In ILP_Hloss, we take all the protection from the less important packets and give them to the more important ones. Both of the ILP_Lloss and ILP_Hloss modes use our proposed Integrated Loss Protection (ILP) method. This method is a combination of the two protection categories, the ROI protection schemes and distortion minimizing techniques, discussed in Section 2.

3.2. Proposed Integrated Loss Protection (ILP) Method

The basic fact in adaptive protection is that different network conditions need different protections. To the best of our knowledge, all existing adaptive methods adapt by increasing the amount of protection as the network conditions get worse. When adding redundancy is used as the protection method, this is translated to adding a larger amount of redundant packets to each block of packets or in other words, increasing the redundancy overhead. This approach will either increase the overall bit rate in a situation where the network is most probably already congested, or the video bit rate/quality should be decreased in order to make more space available for the extra overhead. This will cause degradation in video quality, but it is better than facing an extreme amount of loss. However, we are looking at this problem from another perspective. Instead of increasing the amount of overhead, we are aiming to achieve a better quality by only changing the coding configurations and the distribution of redundant packets among different parts of the video. In other words, our ALP scheme uses the same amount of redundancy overhead in different network conditions, but changes the protection distribution pattern according to the network conditions in a way that makes it suitable for the situation. As a result, it achieves a higher quality. To do so, the four proposed modes should have the same amount of overhead. The first two scenarios (ELP_Frame and ELP_ROI) have the same number of redundant packets and in the following we will discuss the detailed description of the ILP_Lloss and ILP_Hloss modes and describe how they have the same amount of overhead as the previous modes.

Our proposed ILP method aims to protect the ROI of the video, while also minimizing the distortion and the effect of error propagation. It classifies the video packets based on three factors and assigns unequal amounts of protection to each class. The three factors are error propagation, important partitions and regions of interest. Its main challenge is to do this in real-time and without adding extra delay to the transmission process so it can be used for delay sensitive applications such as video conferencing.

To be real-time, it is not enough to send the original packets in real time, but also the protection scheme should not add undesired extra delay. In other words, the redundant data and the extra packets should be sent in such a way that in case of packet loss the receiver can use this redundant data to recover the information without noticeable delay. Otherwise the redundant data would be of no use and the recovered packets would be considered lost, since their playback time has passed. In order to generate and send the redundant packets of a block in real-time, its original packets should be currently available. In applications such as video conferencing, future frames are not yet recorded and there can be 30ms to 50ms delay between the capturing of each frame and the next frame. As a result in such applications we cannot use packets of the future frames for the protection of the current frame. Otherwise we would be adding an extra amount of delay that will be added to the network delay that can already be high. Having in mind that the maximum amount of delay that can be accepted without degrading the quality of experience in an interactive video call is about 100ms,

adding an extra 30ms–50ms delay is significant. As a result, in order to have real-time protection, we restrict ourselves to include in each block of packets only packets from a single frame.

In each GOP, due to the dependency between frames, if a frame faces error, this error will propagate through all future frames that are dependent on that frame. This error propagation will stop with the start of the next GOP when an independent IDR frame arrives. Error propagation causes more severe degradation in the quality than the single error itself, and the earlier this error happens in the GOP, the more severe its effect would be. To minimize error propagation without having to use future frames, in the ILP method we divide the frames in each GOP into three parts based on their position in the group. The first part contains the one third of the frames at the beginning of that GOP. The second part contains the frames in the middle and the last part contains the one third of the frames close to the end of the GOP. The effect of error propagation can be minimized when the first part is protected the most and the last part is protected the least. Furthermore, since certain information in each slice is more important than the rest, such as slice headers and motion vectors, for an even more delicate classification we use the data partitioning tool. After dividing each slice to partitions, we apply a stronger protection on the Header partition. It should be noted that since we have disabled intra prediction in P frames, there is no Intra partition. After these classifications, when distributing the amount of protection between classes, we assign more protection for the more important classes.

Description tables of the protection distribution. To describe and show the protection distribution we use description tables. The columns of this table correspond to the three parts of the GOP. The first and second row correspond to Header partition and Inter partition, respectively. Each cell in the table indicates a separate class of block of packets. We name each cell using the name of its row and column. For example the first cell is called Header-Part1. The numbers in the table show the number of redundant packets added to each block of packet in that class. Each block of packet contains the Header partitions or the Inter partitions of all the ROI slices in one frame, and a number of extra packets added to it for redundancy (based on the table). While noting that in an equal protection case, all cells in the table would have the same number, we describe the ILP_Lloss and ILP_Hloss modes as follows.

ILP_Lloss. In this mode, the inequality of protection is not high and still some protection is provided for the Inter partitions, although more important packets have a stronger protection. In order to make the FEC distribution description tables for this mode, we shift the protection from the down-right side of the table to the top-right side. This shifting is done in a way that the total amount of overhead remains the same. We remove all the protection from the Inter-Part3 class and half of the protection from the Inter-Part2 class. Afterwards, with the bandwidth that has now become available due to removing these protections, we start adding some protection to other classes. We add one additional redundant packet to the Header-Part1 class, another one to the Header-Part2 class, and then one to the Inter-Part1 class. After this, if any bandwidth is still left, we will repeat the previous redundancy adding process, with the same sequence, until there is no bandwidth left. The result will be a table that is shifted to the upper-left side but with the same amount of bandwidth overhead. Since different videos have different Header partition to Inter partition size ratios, the table may differ from one video to another, but the algorithm is the same.

ILP_Hloss. ILP_Hloss is similar to ILP_Lloss, but it distributes the redundant packets in a more unequal manner. Shifting the protection to the more important parts is done such that basically no protection is left for the less important parts. To make the FEC distribution description tables in this mode, we start with the ILP_Lloss table. Then, we take all the protection from the Inter partition classes and add redundant packets

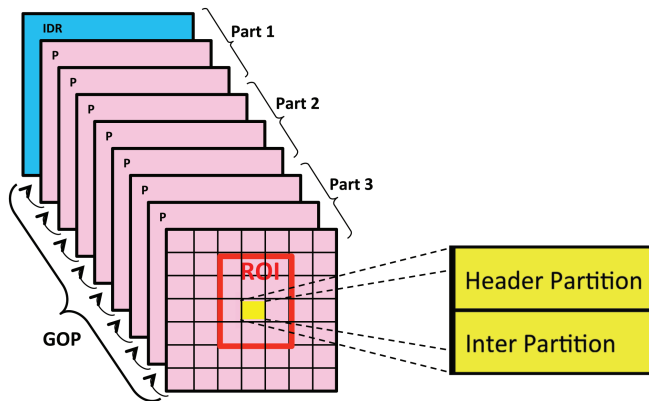


Fig. 3. Combining the use of data partitioning, region of interest and GOP dividing.

Table I. Example of a Description Table for the FEC Distribution

Number of added packets	Part 1	Part 2	Part 3
Header partition	4	4	3
Inter partition	0	0	0

to the Header partition classes. So, this scenario gives all the protection bandwidth for protecting the Header partition packets. As an example, Figure 3 shows how a GOP is divided to three parts and each P frame is divided to 12 ROI slices, and 37 non-ROI slices. Each ROI slice is partitioned to a Header partition and an Inter partition. Table I shows an example of the protection distribution when ILP_Hloss is applied. Note that the protection distribution table is only for the ROI, while the background is left without protection. Since the size of the ROI can be different in different videos, the block of packet sizes can also differ. Also, in different videos the size of Header partition and Inter partition may vary. In some videos the Header and Inter partitions are almost the same size, while in other videos one can be larger than the other. It depends on how much motion prediction can be useful in coding the next frame. Therefore, to have almost the same amount of overhead in different videos, the protection distribution description table may change according to video properties.

3.3. Proposed Mode Switching Metric

In real networks, network parameters such as average burst loss length and packet loss rate can be quite different from network to network or even within the same network over time. As a result, the network model for different networks can be different and complex. We therefore need a metric that has a meaning independent from the network model; a metric that is closely related to the performance of each mode regardless of the network model. The metric should choose the suitable protection mode based on the impact of loss, regardless of whether the packet loss rate, average burst loss length or other complex network parameters have increased or decreased. Loss should be seen from a user-centric perspective and in terms of the effect it has on the visual quality. In other words, we do not need a metric that describes or models the loss, we want it to consider the loss effect on the video quality as perceived by a human. This concept can be more clearly understood when thinking of other protection techniques that can be combined with our method. For example, interleaving is a powerful tool in terms of error recovery and concealment [Wang et al. 2011]. Interleaving changes

the transmission ordering of packets. In other words, it shuffles them before sending so that if a loss burst happens, it would be less likely to lose many packets from the same block of packets, and thus the probability of recovering the lost packets increases. When interleaving is used, each mode can tolerate loss at a higher level. A suitable metric should be transparent to these kinds of techniques.

Our proposed metric considers the strengths of each mode. It shows us when the amount of loss on a certain part is more than it can be tolerated by that mode and we should switch to the next mode. Each mode has the duty of protecting specific parts of the frame. In order to consider the influence of loss on video quality, our metric identifies when the current protection mode cannot be effectively used anymore. This metric is defined as

$$SM(zone) = (Unrecovered\ losses\ in\ zone / All\ losses\ occurred\ in\ zone) \times 100,$$

where zone identifies all or part of the frame that its packets are supposed to be protected by that mode. $SM(zone)$ calculates the percentage of packets that were supposed to be protected from loss but were not recovered by the protection mode. Based on the system's current mode, the zone input of SM changes accordingly. Starting from the first and default mode, `ELP_Frame`, which has the duty of protecting the whole frame. Our metric finds the point where `ELP_Frame` is no longer capable of protecting the whole frame and the loss is more than it can recover. So $SM(Frame)$ means the percentage of all lost packets that could not be recovered by `ELP_Frame`. $SM(Frame)$ has a small value when most of the losses are recovered and its value increases as we get to the point where `ELP_Frame` is not able to protect the frame anymore and we should switch to the next mode, which is `ELP_ROI`.

`ELP_ROI` is designed to protect and recover the ROI. Since in this mode we do not care about the background anymore, switching between `ELP_ROI` and `ILP_Lloss` should be only based on the amount of loss on the ROI packets. Our switching metric for this mode shows us the point where `ELP_ROI` can no longer recover the ROI packets. So $SM(ROI)$ means the percentage of lost ROI packets that could not be recovered by `ELP_ROI`. Again as the network conditions get worse and `ELP_ROI` cannot recover the lost ROIs, $SM(ROI)$ will start to increase. For instance, if `ELP_ROI` is not able to recover any losses, and all losses occurred during the transmission remain unrecovered, $SM(ROI)$ will equal 100%. For switching between `ILP_Lloss` and `ILP_Hloss`, since `ILP_Hloss` only protects the Header partitions of the ROI, a suitable switching point is the point where `ILP_Lloss` is not able to protect the Header partition packets. At this point, it would be better to switch to the `ILP_Hloss` mode for stronger protection on the Header partition. Based on this, we would use $SM(ROI_HEADER)$ for switching between the `ILP_Lloss` and `ILP_Hloss` modes. $SM(ROI_HEADER)$ means the percentage of lost Header partition packets of ROI, that could not be recovered by `ILP_Lloss`.

One of the main advantages of our switching metric is that it doesn't switch the mode just because of a big number of packets lost in the network. As long as the current mode is able to perform its duties, everything is considered fine. In this case switching to the next mode and overprotecting the most important parts while leaving the other parts without any protection can even reduce the overall quality. It can be seen that the proposed metric focuses on the video quality. Therefore, it can be used when other methods such as interleaving are also used. It does not directly include network parameters such as loss or burst rate and only the impact of lost packets is considered in the metric. Therefore, different ranges of network parameters and their variations are indirectly addressed by this metric.

Other than all of the aforementioned issues, a main issue that should be considered is that a good metric should assure us that after switching to another mode we will

achieve a better quality. Having a higher average quality is not enough for ensuring a better quality for all the videos after the mode switch. A high average might just be biased by some good outliers. In this case, to assure better quality we should consider measuring the variance of quality around the switching point. If the standard deviation bars of two subsequent modes have a big overlap, we cannot assure better quality after switching between these two subsequent modes. Results in Section 4.4 compare our proposed metric to the packet loss rate metric and show that by using our metric the probability of having a better video quality after switching significantly increases compared to packet loss rate.

3.4. Analysis and Discussion

In this section we discuss a few important issues about our scheme.

Packetization and Frame Slicing. As discussed before, in order to achieve real-time protection, we have restricted ourselves to define a block of packets using only packets from a single frame. We do this by dividing each frame into multiple slices. As the number of slices increases, the number of packets and thus the size of the block of packets increases. As a result, robustness against loss bursts will also increase. However, more robustness is gained at the expense of more overhead. This overhead is caused by the header fields added to each packet, which increases proportional to the number of packets. Other than that, slicing reduces the encoding efficiency due to limiting the dependency on slice boundaries. When the encoding efficiency is decreased, more data has to be sent for each frame and this increases the rate. On the other hand, the high bit rate of each frame in HD videos causes each frame size to be a lot larger than the maximum packet size that can be normally supported. The Maximum Transmission Unit (MTU) is the largest packet size that can go through the network without fragmentation and is usually assumed to be 1500 bytes because of the maximum size of Ethernet packets. It is recommended to keep the maximum size of a coded slice below the MTU size to avoid fragmentation [Wenger 2003]. If a packet exceeds the MTU size it will get fragmented in the lower layers. In this case, even if one of these fragments is lost in the network, the packet cannot be recovered and all other received fragments are considered lost. This effect will get worse as the original packet size increases. In order to avoid this effect when transmitting HD video, it is useful to divide a frame to packets that can be decoded independently, by using frame slicing.

Since each frame of the HD video should be divided into multiple slices to reach the MTU size, instead of using the usual raster-scan mode for slicing, we take advantage of the dispersed mode, which has showed to be useful for concealing errors. Since dividing a frame into too many slices causes extra overhead, we use the MTU size to decide on the number of slices. If the number of slices is as much as needed for reaching the MTU size, then no extra overhead due to packet header is added, since it should have been divided into this number of packets anyway. Also, since usually in real-time encoders, intra coding within P frames is disabled due to more encoding speed, slicing does not have any effect on the coding efficiency. As a result, slicing the frame, to the point that we reach the MTU size, does not cause any extra overhead in our applications.

Number of Redundant Packets. We have used the average burst loss length as a fair choice for the number of redundant packets that should be added. In order to keep the redundancy bit rate constant, we should reduce the size of each redundant packet as the number of redundant packets increases. Since the number of redundant packets increases based on the average burst loss length, we should reduce the packet sizes accordingly. To do so, we will divide each frame into more slices so the size of the packets would decrease. The reduction in packet size would then compensate the larger number of redundant packets, in terms of bit rate. Therefore, it is better to assign the minimum number of slices (the minimum number of slices needed to reach the MTU

size), to the minimum value of average burst loss length. This way we can increase the number of slices as the average burst loss length increases, while keeping the bit rate fixed. As a result, in our simulations we have divided each frame into as many slices needed to reach the MTU size (minimum number of slices possible) and also assumed the average burst loss length to be two (minimum average burst loss length possible).

Handling I-frame Losses. All the mentioned protection modes are applied only to P-frames. I-frames are the most important frames and without an I-frame the entire GOP will be lost. As a result, to ensure the delivery of I-frames, a very strong protection should be assigned to them. Otherwise, all the bit rate spent on sending the entire GOP and its protection will be a waste. However, I-frames are big in size and have a high bit rate. Therefore, due to the high bit rate of I-frames and the strong protection needed for them, using redundancy is not efficient in terms of bandwidth. Especially in our ALP scheme, where one of the main goals is keeping the redundancy bit rate constant, adding a huge amount of redundancy to I-frames is not feasible.

For this purpose, we have chosen to resend the I-frames in case of loss. But since retransmission causes delay, we don't send the same frame again, instead, we send the fresh and up to date information in a new I-frame, while the previous lost frames are concealed by keeping an old frame on display. After getting feedback from the decoder that the I-frame is lost the encoder will immediately code the next frame as an I-frame. In this case, no extra delay is caused and no huge amount of overhead has to be added. Also, not the entire GOP is lost because of a lost I-frame. We will only lose frames in a Round Trip Time (RTT) period, which causes the receiver to stall for a RTT. Since the RTT is usually an order of magnitude smaller than the duration of a GOP, even in severe network loss rates the delay caused by retransmission is less than losing an entire GOP. It should be noted that retransmitting fresh I-frames is practical in our application because the encoding is in real time and we are not transmitting a pre-encoded video.

Computation and Delay Overhead. The extra added delay of our proposed ALP scheme is negligible. This is because we send each frame and its redundancy in real time, and also because our scheme is computationally efficient. This is however without considering the computation needed for FEC, since FEC is a general concept that every delay sensitive applications needs to include and there are a lot of different FEC techniques with different ranges of complexity. Which technique to use, depends on the application. Our scheme; however, works with all FEC techniques. Other than the FEC, the additional computation required in our scheme is just counting the number of lost and unrecovered packets in the receiver, which can be considered negligible compared to the video coding and decoding itself, in terms of both computational cost and delay.

Finding the Switching Points. In our experiments the switching points are found empirically by data collected from three different videos. For each two subsequent modes, we measured and plotted their video quality in different network conditions. The point where the two video quality curves intersect, and the quality of one mode starts to go beyond the other, is chosen as the switching point between the two modes. Since most video chats are similar in content, such as having a person talking in front of a background, we expect our empirically found switching points to be suitable for videos across this category.

However, for more complex videos, switching points are tuned adaptively over time as the system is running. They have an initial value that gets refined after each switching event. The initial value is the empirically found switching points. The refinement is done based on whether or not the quality was enhanced after switching to the next mode. If after the switch the quality had improved, then it was the right time to switch. If the quality had improved significantly then the switch should have been done sooner and if the quality hadn't improved yet, then it was too early to switch. Based on these

facts after each switch, the switching points will be automatically updated. It should be noted that the receiver uses no reference quality measurement techniques to estimate the quality without having the original video.

4. EVALUATION

In this section, we compare our proposed ALP scheme with the two protection categories described in Section 2. We then evaluate our proposed ALP scheme, and show how it improves the quality in all packet loss rate ranges, and even outperforms each of its individual protection modes in their peak performance. Finally, we compare our proposed switching metric against packet loss rate and show that the variance of quality is smaller using our proposed switching metric, which makes it more reliable for switching.

4.1. Setup

Videos. To evaluate the proposed scheme, HD videos representing common video conference/chats are required. Among the low resolution standard test sequences, there are multiple sequences, such as “Miss America,” “Akiyo,” and “Claire,” that can match the corresponding features and characteristics of a common video conference/chat. However, among the existing HD standard test sequences, none can resemble a video conference/chat sequence [Xiph.org][VQEG][HD-VideoBench]². They are all focused on scenes from nature, streets and city shots, cartoons and a few sports videos. We also went through an extensive search for any suitable HD video, in any video database such as [TestVid][Ultra Video Group][Elemental][EBU][CDVL],³ but unfortunately none existed. Thus, we had to record our own HD (720p) videos that have the same features and characteristics of a common video chat. All of our recorded videos are head and shoulder shots of various participants talking in front of a camera. For simplicity, we select the ROI as a rectangle around the face. The background is almost steady, but the foregrounds of different videos have different amounts of motion. In general, most video conference sequences, including our recorded video sequences, are considered as low motion sequences.

The videos are encoded using the H.264 standard with a frame rate of 30 fps, GOP size of 30 and a frame structure of IPPP (as explained before, B frames cause delay and are not used in interactive and delay sensitive applications). The Quantization Parameter (QP) used is 30. The encoding is done by the JM software [JM18.0]⁴. We also adjust some of the encoding parameters to increase the speed. For example, Intra prediction in P frames is disabled, the number of reference frames is decreased to one, the search range is limited and a fast search mode is used. For error concealment the Frame Copy concealment mode of JM is used. Three different videos are tested, the first video has a bit rate of 2.3 Mbps and each of its frames are divided into 2 ROI slices and 5 non-ROI slices to reach the MTU size. The second video has a bit rate of 2 Mbps and in order to reach the MTU size its frames are again divided into 2 ROI slices and 5 non-ROI slices. The third video has a bit rate of 3.2 Mbps and needs 3 ROI slices and 6 non-ROI slices per frame. When the slices are partitioned, the average size of the ROI Header partition in the first video is 1280 Bytes and the average size of its Inter partition is 1007 Bytes. Low Inter partition sizes denote a high encoding

²Xiph.org. Video Test Media. <https://media.xiph.org/video/derf/>. VQEG. Video Quality Experts Group. ftp://vqeg.its.bldrdoc.gov/HDTV/NTIA_source/. HD-VideoBench. A Benchmark for Evaluating High Definition Digital Video Applications. <http://personals.ac.upc.edu/alvarez/hdvideobench/install.html>.

³TestVid. <http://www.testvid.com/>. Ultra Video Group. <http://ultravideo.cs.tut.fi/#testsequences>. Elemental. <http://www.elementaltechnologies.com/resources/4k-test-sequences>. EBU. <https://tech.ebu.ch/hdvt-test-sequences>. CDVL. Consumer Digital Video Library. <http://www.cdvl.org/>.

⁴JM18.0. H.264/AVC Software Coordination, JM, Ver. 18.0. <http://iphome.hhi.de/suehring/tml/>.

efficiency. Usually in videos with low motion, the Inter partition is small, since the motion prediction process in the encoder is executed efficiently. For the second video the average ROI partition sizes for the Header and Inter partition are 942 Bytes and 835 Bytes respectively. Also the average ROI partition sizes for the third video are 1682 Bytes, 2496 Bytes for the Header and Inter partitions respectively.

Since the Header and Inter partition sizes are different for different videos, the protection distribution description tables are changed accordingly, since the overhead has to be as close as possible to the equal protection mode. As described in Section 3.2, based on a video's packet sizes, the description tables are generated and updated automatically for that video, in both sender and receiver.

Network model. The lossy network environment and the loss protection modes are simulated in Matlab. For the network model, we use the Gilbert model in our simulations, which is a simplified version of the Gilbert-Elliott model. The Gilbert-Elliott model is a widely used model in transmission channels [Elliott 1963]. This model is a 2-state Markov model that consists of two states known as the Good (G) and Bad (B) states. The B state has a higher loss rate than the G state and the transition probabilities between these two states are α and β . The Gilbert model [Gilbert 1960] is when the G state is assumed to be loss free and the loss rate of the B state is 1. In this model, the total packet loss ratio and the average burst loss length can be calculated based on the α and β parameters using equation 1. In a real network, the receiver can measure the Packet Loss Rate (PLR) and the average burst loss length and send it as a feedback to the source. The α and β parameters of the model are calculated using the packet loss rate and average burst loss length. The simulations are done for different packet loss rate values, ranging from 1% to 20%, and a MTU size of 1500 Bytes is used. In addition, as discussed in Section 3.4, in order to keep the bit rate fixed, we increase the number of slices as the average burst loss length increases. As a result, in our simulations we divide each frame into as many slices needed to reach the MTU size (minimum number of slices possible) and assume the average burst loss length to be 2, as the minimum average burst loss length possible.

$$\begin{aligned} \text{Packet Loss Rate} &= \beta/(\alpha + \beta) \\ \text{Average Burst Loss Length} &= 1/\alpha. \end{aligned} \tag{1}$$

Performance metrics. For measuring and comparing decoded video qualities, we use the PSNR (Peak Signal to Noise Ratio) and SSIM (Structural SIMilarity) metrics. PSNR is the most common metric for measuring video quality. It is simple and fast and therefore widely used. However, PSNR is proven to be inconsistent with perceptual quality while the SSIM quality metric has shown to be more consistent with human perception. SSIM⁵ is a complex but accurate quality metric which is based on pixel interdependencies. In our experiments we show the results using both PSNR and SSIM metrics. For evaluating our switching metric we use standard deviation. Standard deviation shows the dispersion of the data. The standard deviation bars around each average value show the range where the data is most likely to be in.

4.2. Comparison of Our Proposed ALP Scheme

We claim, in Section 2, that our ILP method as the core of our ALP scheme, is able to combine the advantages of the two subcategories mentioned as the ROI protection schemes and the distortion minimizing techniques. Here we will compare our scheme against two schemes, each as a representative of one of these two subcategories. In Arachchi et al. [2006] as a representative of the ROI protection schemes, the ROI is

⁵SSIM. Structural Similarity. http://en.wikipedia.org/wiki/Structural_similarity.

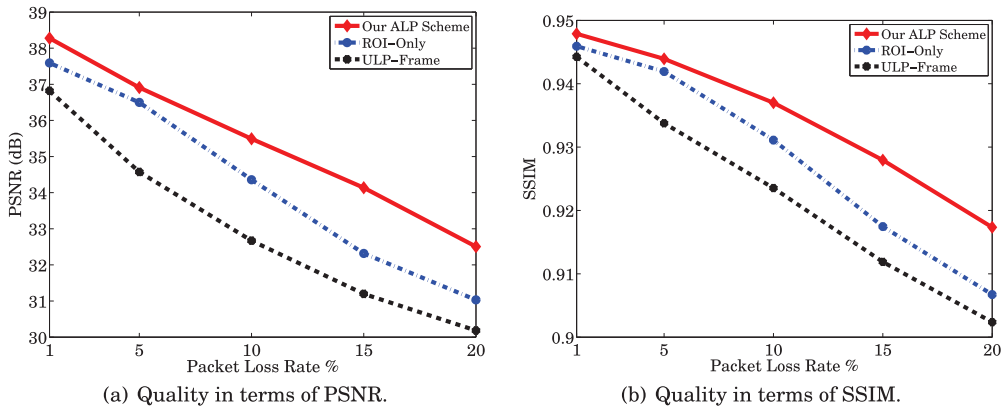


Fig. 4. Our ALP scheme outperforms both ROI protection methods, and unequal loss protection methods that don't consider ROI.

divided into two slices using a checkerboard pattern and more protection is applied to the ROI slices compared to the background slice. We define the ROI_Only scenario based on [Arachchi et al. 2006] as follows. We extract the ROI of video and slice it in a dispersed manner to reach the MTU size. There is no data partitioning and no attempt is made to reduce error propagation. Protection is only applied to the ROI slices. The protection overhead of this scenario is the same as our proposed scheme. To be precise, this scenario is actually an instance of the ELP_ROI mode.

In Zhang and Peng [2009], as a representative of the distortion minimizing techniques that reduce distortion and error propagation, each GOP is divided into 4 parts, and the earlier parts are the more important ones. In addition, data partitioning is used and more importance is given to the Header partitions. We define the ULP_Frame scenario based on Zhang and Peng [2009] as follows. The GOP is divided into three parts and data partitioning is used. In order to make the overhead same as our proposed scheme, the unequal distribution of redundant packets is the same as in our ILP_Hloss mode. However, the ROI is not extracted and no extra protection is assigned to it. Also for slicing each frame to reach the MTU size, the raster-scan mode is used instead of dispersed, because it is the default way of slicing.

The simulations are performed at least 30 times and the average value of the results is reported. Figure 4 shows the average PSNR and average SSIM for different packet loss rate conditions. From the results, it can be seen that our proposed ALP scheme achieves a significantly higher quality, in terms of both PSNR and SSIM. In the case of PLR = 20% it can be seen that our method increases the quality by an average of 2.5dB compared to ULP_Frame and about 1.5dB compared to ROI_Only.

4.3. Analysis of the Proposed ALP Scheme

In this Section we analyze our ALP scheme by comparing it against its own components, in terms of average quality and standard deviation of quality. The components are the four modes ELP_Frame, ELP_ROI, ILP_Lloss and ILP_Hloss. Based on the definition provided for each mode in Section 3, in this experiment, assuming an average burst loss length of 2, the four modes are as follows. The ELP_Frame mode adds 2 redundant packets to each block of packets. The ELP_ROI also adds 2 redundant packets but only to the ROI packets of each frame. For ILP_Lloss and ILP_Hloss, the protection is applied based on the protection distribution description tables. As described in Section 3.2, the sender automatically makes the description tables for each video, based on the packet sizes of that video, and the tables may differ for different videos. Tables II and III show

Table II.
Description table of the FEC distribution for the ILP_Lloss mode of the first video sample.

Number of added packets	Part 1	Part 2	Part 3
Header partition	3	3	2
Inter partition	3	1	0

Table III.
Description table of the FEC distribution for the ILP_Hloss mode of the first video sample.

Number of added packets	Part 1	Part 2	Part 3
Header partition	4	4	3
Inter partition	0	0	0

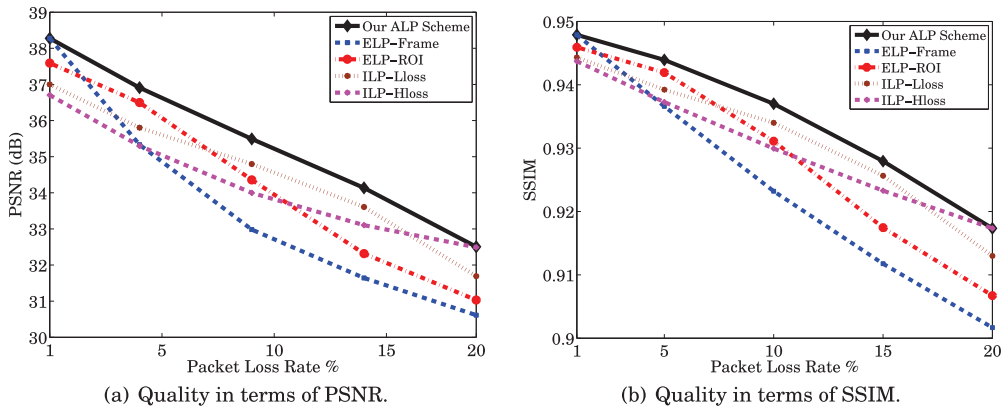


Fig. 5. Our proposed ALP scheme achieves a higher average quality compared to each of the 4 modes individually.

the FEC distribution description tables for the first video. Each frame in this video is divided into 2 ROI slices and 5 non-ROI slices. The overhead for all modes is around 20% with a maximum difference of 0.3%.

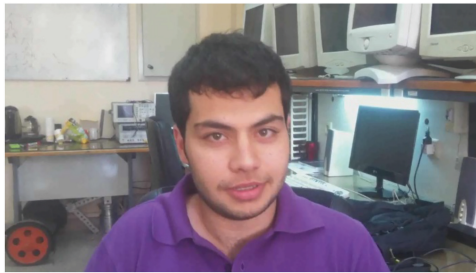
We test our full adaptive loss protection scheme, and the simulation results are shown in Figure 5(a) and (b), which show the decoded video quality based on the PSNR and SSIM metrics, respectively. The impact of network on the video transport process is simulated by applying the random packet loss process on the video packets. For each packet loss rate, the simulation was performed 100 times and the average value is reported.

It can be seen that using our proposed scheme, the achieved average quality is better than using each of the four modes alone, in their best performed conditions, and up to 3dB improvement is achieved compared to the basic protection method, ELP_Frame. The high performance is achieved because, for each packet loss rate and for each run, the best performing mode has been chosen independently, based on our switching metric (SM). In other words, at a certain packet loss rate, not all the runs choose the same mode, and each one chooses the mode that suits it the best. So the average quality increases and the variance between them is decreased. Table IV shows the standard deviation of quality for all modes and all packet loss rates. It can be seen that our ALP scheme also decreases the standard deviation of the PSNR, showing that it can provide a more consistent video quality under different packet loss conditions.

Table IV.

Our ALP scheme achieves a lower standard deviation of quality for all PLRs, compared to its individual modes.

Standard Deviation of PSNR (dB)	PLR = 1%	PLR = 5%	PLR = 10%	PLR = 15%	PLR = 20%
Our ALP Scheme	0.3	0.84	0.8	0.84	0.57
ELP_Frame	0.37	1.55	1.55	1.5	1.4
ELP_ROI	0.82	1.47	1.5	1.18	1.04
ILP_Lloss	0.9	1.11	0.99	1.28	1.01
ILP_Hloss	1	0.95	0.98	1.04	1.06



(a) When using the ELP_Frame mode



(b) When using the ELP_ROI mode

Fig. 6. A sample frame with PLR = 1%, showing that ELP_Frame can achieve higher qualities than ELP_ROI.

From the results, it can also be seen that each mode is good for a different packet loss range. For very low PLR conditions, ELP_Frame achieves a better quality, since loss is rare and a small amount of protection is enough. Since this rare loss can happen in any part of the frame, it is best to protect it all but with low protection, so that most of the losses will be recovered. As the loss increases, we will get to a point where most of the losses cannot be recovered using this low protection and so the ELP_Frame mode starts to get less efficient. At this point, it is better to concentrate the protection on the ROI rather than protecting the whole frame, so that at least we would be able to save the ROI even if we are experiencing some loss in the background part. This is why, as it can be seen in the results, after the packet loss increases a bit, the ELP_ROI mode reaches a better quality. However, ELP_ROI will also reach a point where this amount of protection is not enough for the ROI and most of the ROI slices are getting lost and cannot be recovered. From this point on, putting more protection on the more important parts will help achieve better results since even having the Header partition of each slice will give us a rough estimation of the slice and a rough estimation is better than totally losing the slice. As the inequality in the distribution of redundant packets increases, it gets more suitable for high packet loss rates, since it protects the important parts at the expense of losing the less important parts, which gives us a rough estimation of that frame. This is good only if there is no chance of saving the less important parts with that certain amount of overhead. Therefore, in an efficient protection scheme, important classes should be protected more but not overprotected, so that some protection would remain for the less important classes if possible.

Figures 6, 7, and 8 are some sample frames from three video sequences, which show the difference between different modes in different packet loss rates. Figure 6, shows the difference between ELP_Frame and ELP_ROI, in PLR = 1%. It can be seen that the ROI is fine in both, while ELP_Frame has a better background quality. Figure 7, is a sample frame in PLR = 5%, showing the difference between modes ELP_ROI and ILP_Lloss. In this packet loss range, ELP_ROI can still recover the ROI, and we can

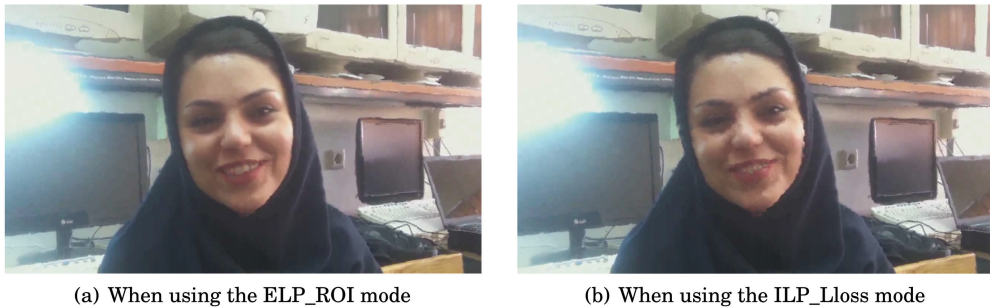


Fig. 7. A sample frame with PLR = 5%, showing that ELP_ROI can achieve higher qualities than ILP_Lloss.

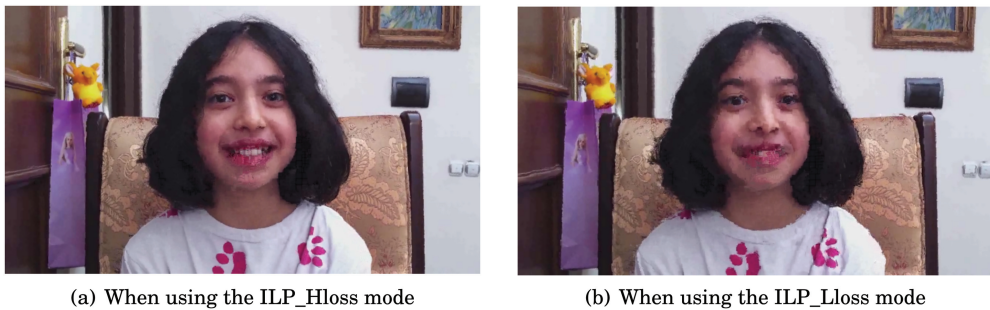


Fig. 8. A sample frame with PLR = 20%, showing that ILP_Hloss can achieve higher qualities than ILP_Lloss in high PLRs.

see that ELP_ROI achieves a better quality. Figure 8 shows the difference between ILP_Lloss and ILP_Hloss, in PLR = 20%. It can be seen that ILP_Hloss achieves a better quality, although both videos have experienced a large amount of loss and quality degradation. This is because ILP_Hloss is able to protect and recover most of the Header partitions.

4.4. Performance of the Proposed Switching Metric

A suitable metric should decide the switching point in a way that we could be sure that if we switch based on this metric we would achieve a better quality. Having a higher average quality is not enough for ensuring a better quality for all the videos after the mode switch. A high average quality might be biased by some outlier videos having good quality while actually the rest of the videos have worse quality. To assure better quality we should consider the standard deviation bars. If the standard deviation bars of two subsequent modes have a big overlap, we cannot assure better quality after switching between these two subsequent modes.

The alternative to our proposed switching metric is using packet loss rate for switching between protection modes. We compare our proposed switching metric against packet loss rate in terms of standard deviation of quality. The comparisons are done around each switching point. The results are obtained by running each mode 100 times for each PLR value.

The first experiment is for switching between the ELP_Frame and ELP_ROI modes. Figure 9 shows the standard deviation and average quality, for both modes, around the switch point. The switch point of ELP_Frame is the point where the two ELP_Frame and ELP_ROI modes intersect. Figure 9(a) is with respect to our SM(Frame) metric, while Figure 9(b) is with respect to the packet loss rate. It can be seen that when

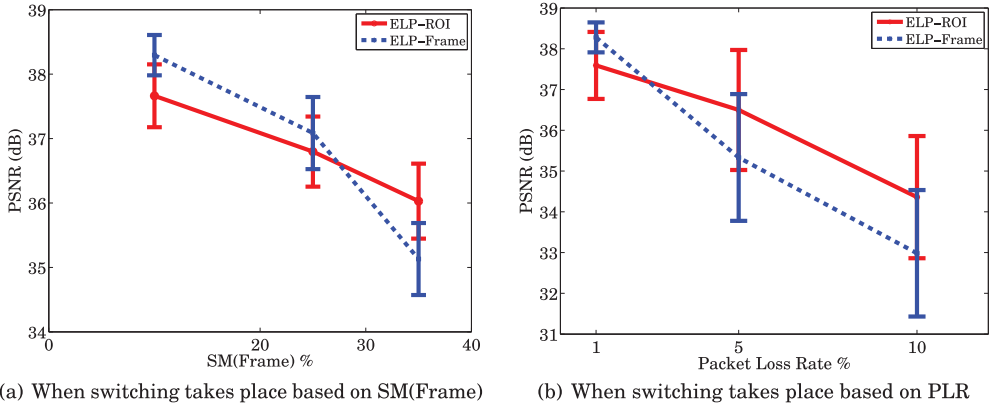


Fig. 9. When switching takes place based on SM(Frame), instead of PLR, the overlapping of standard deviation bars around the switch point of ELP_Frame and ELP_ROI decreases.

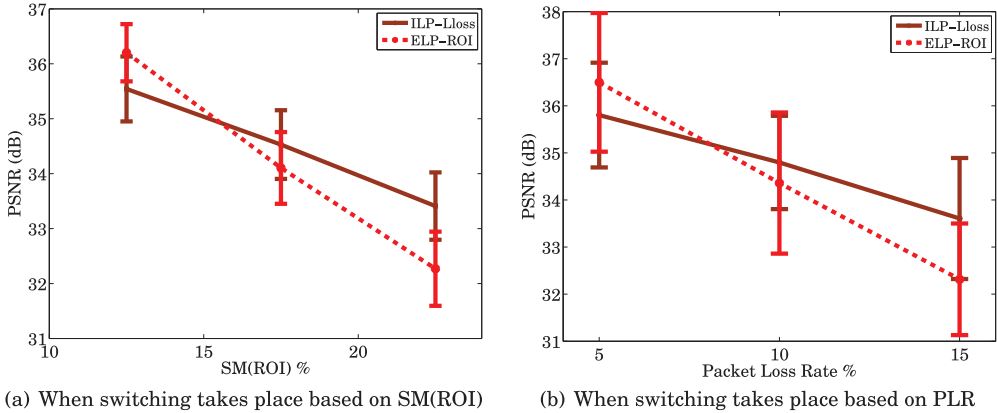


Fig. 10. When switching takes place based on SM(ROI), instead of PLR, the overlapping of standard deviation bars around the switching point of ELP_ROI and ILP_Loss decreases.

switching based on packet loss rate, the variance in each point and the overlapping of standard deviation bars is too much. As a result, for a single video we cannot be sure about the mode that will achieve the best quality, although it can still be said that generally and on average, ELP_ROI would be more useful in higher packet loss rates. There is a difference between finding a mode that is more effective on average and choosing a mode for one video, in a real-time manner and assuring that its quality will be maximized using this mode. Using our metric we can achieve a more acceptable variance of quality, such that the probability of achieving a higher quality after the switch, would increase.

The next experiment is for testing the switching point between the ELP_ROI and ILP_Loss modes. Since ILP_Loss is data partitioned, we have used the average of lost Header and Inter partitions of ROI, as the number of lost ROI packets. Figure 10, shows the quality of these two modes around the switching point, with respect to the SM(ROI) metric and packet loss rate. Again, it can be seen that the packet loss rate is not a suitable metric for deciding the switching time, while with our SM metric the overlapping of standard deviation bars is so small. As a result the SM(ROI) can ensure us that the quality will increase after switching is performed.

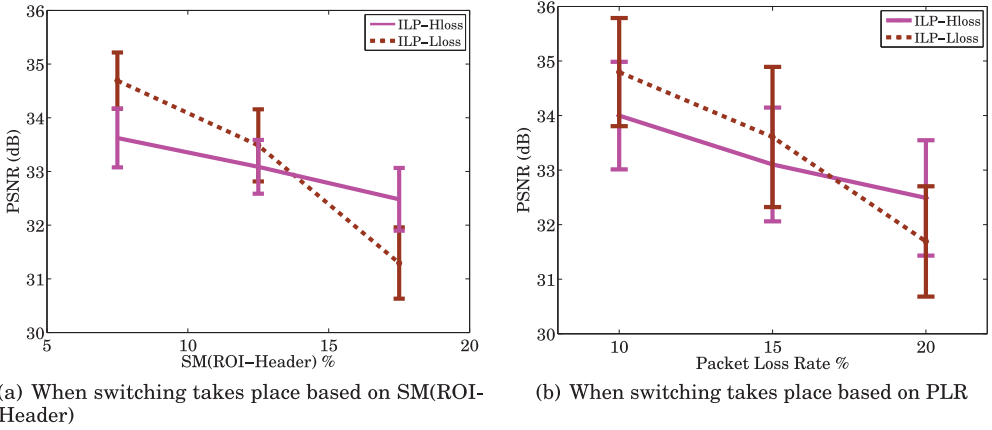


Fig. 11. When switching takes place based on SM(ROI-Header), instead of PLR, the overlapping of standard deviation bars around the switching point of ILP_Lloss and ILP_Hloss decreases.

For comparing our SM metric with packet loss rate in the switching point of ILP_Lloss and ILP_Hloss modes, we have also performed another experiment. The result is shown in Figure 11, which validates that the standard deviation based on our SM metric is more acceptable, since the standard deviation bars of the two modes do not overlap much, both before and after switching. So with a high probability, the quality of ILP_Lloss would be better before switching and ILP_Hloss would be better after switching.

Based on all the results and discussions, it can be concluded that unlike packet loss rate, our proposed metric is a suitable metric for our ALP scheme, which can reduce the variance of quality and increase the overall quality of the scheme. Combining the use of this metric with our proposed loss protection modes has significantly improved the overall performance of our ALP scheme.

5. CONCLUSIONS

In this article we have proposed an Adaptive Loss Protection (ALP) scheme with negligible delay for protecting interactive video applications, such as video conferencing. This scheme does not increase the protection overhead as network conditions worsen. Our proposed scheme consists of four protection modes, each suitable for a different range of network conditions, and a switching metric for adaptively switching among these modes. The ALP scheme is stable against sudden and short network changes, since it avoids changing the protection mode suddenly and dramatically.

In addition, we have proposed an integrated (ILP) method for loss protection that integrates three factors for distributing the protection among packets. The three factors are: reducing error propagation, protecting ROI, and protecting essential information such as headers. The ILP method is used in two of the modes in our ALP scheme. The first two modes of the ALP scheme are instances of equal loss protection, while the last two modes are instances of unequal loss protection using our ILP method.

The proposed switching metric increases the average quality of our scheme, while reducing the variance of quality. The increase in average quality together with the decrease in variance of quality mean a high chance of achieving a better quality. We have compared our switching metric with the commonly used packet loss rate metric, and showed that unlike packet loss rate, our switching metric can make sure that a better quality will be achieved after each switching event.

We evaluated the proposed ALP scheme and the switching metric using detailed simulations on real HD videos. The simulation results show that the ALP scheme outperforms the current state-of-the-art methods, by improving the quality up to 3dB. In addition, the ALP scheme achieves a higher quality compared to its own components individually, while reducing the standard deviation of quality up to 0.7dB.

ELECTRONIC APPENDIX

The electronic appendix for this article can be accessed in the ACM Digital Library.

REFERENCES

- Andres Albanese, Johannes Blomer, Jeff Edmonds, Michael Luby, and Madhu Sudan. 1996. Priority encoding transmission. *IEEE Trans. Inf. Theory* 42, 6, 1737–1744.
- Hemantha Kodikara Arachchi, Warnakulasuriya Anil Chandana Fernando, S. Panchadcharam, and W. A. Rajitha Jayaruwan Weerakkody. 2006. Unequal error protection technique for ROI based H.264 video coding. In *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE'06)*. 2033–2036.
- Ali Borji, Dicky N. Sihite, and Laurent Itti. 2012. Salient object detection: A benchmark. In *Proceedings of the 12th European Conference on Computer Vision (ECCV'12)*. Springer, 414–429.
- Kiana Calagari, Mohammad Reza Pakravan, and Shervin Shirmohammadi. 2012. ROI-based protection scheme for high definition interactive video applications. In *Proceedings of the 20th ACM International Conference on Multimedia (ACM MM'12)*. ACM, 1033–1036.
- Diep Thang Cao, Thuc Hai Nguyen, and Linh Giang Nguyen. 2013. Improving the video transmission quality over IP network. In *Proceedings of the 5th International Conference on Ubiquitous and Future Networks (ICUFN'13)*. IEEE, 570–575.
- Qian Chen, Li Song, Xiaokang Yang, and Wenjun Zhang. 2007. Robust region-of-interest scalable coding with leaky prediction in H.264/AVC. In *Proceedings of the IEEE Workshop on Signal Processing Systems*. IEEE, 357–362.
- Bogdan Ciubotaru, G. Muntean, and George Ghinea. 2009. Objective assessment of region of interest-aware adaptive multimedia streaming quality. *IEEE Trans. Broadcasting* 55, 2, 202–212.
- Yves Dhondt, Peter Lambert, and Rik Van de Walle. 2006. A flexible macroblock scheme for unequal error protection. In *Proceedings of the IEEE International Conference on Image Processing*. IEEE, 829–832.
- E. O. Elliott. 1963. Estimates of error rates for codes on burst-noise channels. *Bell Syst. Tech. J.* 42, 5, 1977–1997.
- Ivan Alen Fernandez, Christophe De Vleeschouwer, George Toma, and Laurent Schumacher. 2012. An interactive video streaming architecture featuring bitrate adaptation. *J. Commun.* 7, 4, 265–280.
- Edgar N. Gilbert. 1960. Capacity of a burst-noise channel. *Bell Syst. Tech. J.* 39, 5, 1253–1265.
- Bernd Girod, Klaus W. Stuhlmüller, M. Link, and U. Horn. 1998. Packet-loss-resilient Internet video streaming. In *Proceedings of SPIE Visual Communications and Image Processing*. SPIE, 833–844.
- Erik Hjeltnäs and Boon Kee Low. 2001. Face detection: A survey. *Comput. Vision Image Understanding* 83, 3, 236–274.
- Rein-Lien Hsu, Mohamed Abdel-Mottaleb, and Anil K. Jain. 2002. Face detection in color images. *IEEE Trans. Pattern Anal. Mach. Intell.* 24, 5, 696–706.
- Tz-Huan Huang, Kai-Yin Cheng, and Yung-Yu Chuang. 2009. A collaborative benchmark for region of interest detection algorithms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09)*. IEEE, 296–303.
- Abbas Javdtalab, Mona Omidyeganeh, Shervin Shirmohammadi, and Mojtaba Hosseini. 2011. A rate control algorithm for x264 high definition video conferencing. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'11)*. IEEE, 1–6.
- Mohammad Kazemi, Shervin Shirmohammadi, and Khosrow Haj Sadeghi. 2014. A review of multiple description coding techniques for error-resilient video delivery. *Multimedia Syst.* 20, 3, 283–309.
- Pei-Jun Lee and Yi-Jie Shih. 2009. Fast inter-frame coding with intra skip strategy in H.264 video coding. *IEEE Trans. Consum. Electron.* 55, 1, 158–164.
- Yihua Lou, Ting Yao, Yongquan Chen, and Wenjun Wu. 2013. A novel scheme of ROI detection and transcoding for mobile devices in high-definition videoconferencing. In *Proceedings of the 5th Workshop on Mobile Video (MoVid'13)*. ACM, 31–36.

- Aditya Mavlankar, David Varodayan, and Bernd Girod. 2007. Region-of-interest prediction for interactively streaming regions of high resolution video. In *Proceedings of the 16th International Packet Video Workshop (PV'07)*. IEEE, 68–77.
- Gabriel-Miro Muntean, Gheorghita Ghinea, and Timothy Noel Sheehan. 2008. Region of Interest-based adaptive multimedia streaming scheme. *IEEE Trans. Broadcasting* 54, 2, 296–303.
- Abdelhamid Nafaa, Tarik Taleb, and Liam Murphy. 2008. Forward error correction strategies for media streaming over wireless networks. *IEEE Commun. Mag.* 46, 1, 72–79.
- Wilfried Osberger and Anthony J. Maeder. 1998. Automatic identification of perceptually important regions in an image. In *Proceedings of the 14th International Conference on Pattern Recognition*. IEEE, 701–704.
- Timothy Porter and Xiao-Hong Peng. 2011. HYBRID TCP/UDP video transport for H.264/AVC content delivery in burst loss networks. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'11)*. IEEE, 1–5.
- Horacio Sanson, Alvaro Neira, Luis Loyola, and Mitsuji Matsumoto. 2010. PR-SCTP for real time H.264/AVC video streaming. In *Proceedings of the 12th International Conference on Advanced Communication Technology (ICACT'10)*. IEEE, 59–63.
- Wei Song, Dian W. Tjondronegoro, Shu-Hsien Wang, and Michael J. Docherty. 2010. Impact of zooming and enhancing region of interests for optimizing user experience on mobile sports video. In *Proceedings of the International Conference on Multimedia (ACM MM'10)*. ACM, 321–330.
- Klaus Stuhlmüller, Niko Farber, Michael Link, and Bernd Girod. 2000. Analysis of video transmission over lossy channels. *IEEE J. Selected Areas Commun.* 18, 6, 1012–1032.
- Hui Su. 2009. Intra-skip in inter-frame coding of H.264/AVC. *Int. J. Digital Multimedia Broadcasting*.
- Nikolaos Thomos, Savvas Argyropoulos, Nikolaos V. Boulgouris, and Michael G. Strintzis. 2006. Robust transmission of H.264/AVC video using adaptive slice grouping and unequal error protection. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME'06)*. IEEE, 593–596.
- Paul Viola and Michael J. Jones. 2004. Robust real-time face detection. *Int. J. Computer Vision* 57, 2, 137–154.
- Xiaolong Wang, Kun Tang, and Huijuan Cui. 2010. A novel unequal error protection scheme for ROI based video coding in H.264/AVC. In *Proceedings of the 2nd International Conference on Computer Modeling and Simulation (ICCMS'10)*. IEEE, 175–178.
- Yu Wang, Xuan Jing, Jo Yew Tham, Kwong Huang Goh, and Wei Siong Lee. 2011. Error-resilient slice interleaving for robust video transmission over bursty loss channels. In *Proceedings of the 17th IEEE International Conference on Networks (ICON'11)*. IEEE, 41–46.
- Yao Wang and Qin-Fan Zhu. 1998. Error control and concealment for video communication: A review. *Proc. IEEE* 86, 5, 974–997.
- Stephan Wenger. 2003. H.264/AVC over IP. *IEEE Trans. Circuits Syst. Video Technol.* 13, 7, 645–656.
- Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the H.264/AVC video coding standard. *IEEE Trans. Circuits Syst. Video Technol.* 13, 7, 560–576.
- Dapeng Wu, Yiwei Thomas Hou, and Ya-Qin Zhang. 2000. Transporting real-time video over the Internet: Challenges and approaches. *Proc. IEEE* 88, 12, 1855–1877.
- Yi-Fei Xu and Hai-Bo Hu. 2013. Development of a network adaptive H.264/AVC medical video transmission system. *Int. J. Future Computer Commun.* 2, 4, 292–295.
- Cha Zhang and Zhengyou Zhang. 2010. A survey of recent advances in face detection. Tech. Rep. MSR-TR-2010-66. Microsoft Research Technical Report, Redmond, WA.
- Xingjun Zhang and Xiaohong Peng. 2009. An unequal packet loss protection scheme for H.264/AVC video transmission. In *Proceedings of the 23rd International Conference on Information Networking (ICOIN'09)*. IEEE, 195–199.
- Xingjun Zhang, Xiaohong Peng, Scott Fowler, and Dajun Wu. 2010. Robust H.264/AVC video transmission using data partitioning and unequal loss protection. In *Proceedings of the 10th IEEE International Conference on Computer and Information Technology (CIT'10)*. IEEE, 2471–2477.
- Xingjun Zhang, Xiaohong Peng, Dajun Wu, Tim Porter, and Richard Haywood. 2009. A hierarchical unequal packet loss protection scheme for robust H.264/AVC transmission. In *Proceedings of the 6th IEEE Consumer Communications and Networking Conference (CCNC'09)*. IEEE, 1–5.
- Yanchun Zhong, Heng Li, and Jialei Bi. 2010. A feedback-based method for adaptive ROI protection in H.264/AVC. In *Proceedings of the 5th International Conference on Digital Information Management (ICDIM'10)*. 145–151.

Received January 2014; revised July 2014; accepted August 2014