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

KIANA CALAGARI, Sharif University of Technology, Iran

MOHAMMAD REZA PAKRAVAN, Sharif University of Technology, Iran

SHERVIN SHIRMOHAMMADI, University of Ottawa, Canada

MOHAMED HEEFEEDA, 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, specifically designed for applications such as video conferencing and video chats, where a main region of interest (ROI) exists in the video. 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

ACM Reference Format:

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][Microsoft Lync]. 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 a possible, video packets must be protected against packet loss. 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 [Javadtalab 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 increased bandwidth demand. As such, designing protection schemes for interactive HD video applications remains a significant challenge.

A variety of video protection schemes exist. 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 doesn't 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 double 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 condition.

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 network condition? and (2) exactly when to switch between different protection modes? Answering these two questions is the goal of this paper, specifically, we propose a novel Adaptive Loss Protection (ALP) scheme which adds negligible delay to the transmission process and therefore is suitable for delay sensitive and interactive video applications. The proposed ALP scheme adaptively applies four different protection modes to cope with the dynamic network conditions. The ALP scheme consists of four protection modes, each of these modes also utilizes a protection method. Two of the modes rely on the state-of-the-art protection methods, while the other two rely on our proposed ILP protection method. Our main contributions in this work are:

1. A novel video protection method for high packet loss conditions which 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 switching dynamically between the protection modes, which is based on the percentage of unrecovered lost packets of the video. This metric provides a systematic way to decide when to switch among different protection modes, so that the video quality is maximized in all network conditions. This metric enables us to reduce the variance in video quality in different protection modes. The less the variance in video quality, the more we can be sure of the actual video quality in that protection mode and the higher the improvement gained by switching to that mode. Using this novel metric, we examine the efficiency of each protection mode based on how successful it is in recovering lost packets, and show the benefits of this metric over using other metrics based on network statistics such as packet loss rate and/or average burst loss length.
In addition, we conduct extensive simulation study with multiple videos recorded. Our results show that: (1) The ALP scheme achieves up to 3dB PSNR improvement in video quality, compared to other related work. (2) The ALP scheme achieves up to 3dB PSNR improvement, compared to its own underlying protection modes. (3) The ALP scheme reduces the standard deviation in quality up to 0.7 dB, compared to when a single protection mode is used.

The rest of this article is organized as follows. We discuss the related work and background 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 conclude the paper in Section 5.

2. BACKGROUND AND RELATED WORK

2.1 Background

Coding Structure: In the coding structure of most video coding standards [Wiegand et al., 2003], several frames are grouped together to form a Group Of Pictures (GOP). The first frame in the beginning of a GOP is an Instantaneous Decoder Refresh (IDR) frame which does not depend on any other frame. Following the IDR frame are a number of P-frames and/or B-frames. P-frames are dependent on the previous frames and use them as references. B-frames can use both previous and future frames as references. To decode a P or B frame, first their references should be received and decoded correctly, otherwise it would cause errors. Therefore B-frames can only be decoded after some future frames used as their reference have been received and decoded. This causes delay and thus B-frames are usually not used in delay sensitive video applications.

Moving a level deeper into the coding structure, each frame is divided into Macro Blocks (MBs). Macro blocks are the coding units. During the encoding process, each macro block is coded in Intra or Inter mode. In Intra mode, the block is coded without referring to any other frame, whereas in inter mode a reference for that block is chosen from the previous frames (or future frames if the block is part of a B-frame) and only the residual is coded and sent. In this case, the position of the reference block should also be sent, which is called the Motion Vector. The motion vector for each macro block points to the macro block that has been chosen as its reference. Each frame can be divided into several independent pieces called slices. Slices are self-contained and can be decoded independently. Each slice consists of a number of macro blocks which are not allowed to depend on any part of that frame beyond the slice boundaries. During transmission, slices are encapsulated in separate packets and since they are independently decodable, if one packet is lost, other slices in that frame can still be decoded correctly and also help in concealing the error caused by lost packet.

Forward Error Correction: To protect the packets of an interactive video application against packet loss, redundancies are added. Forward Error Correction (FEC) codes are widely used to add controlled redundancy that can help when there is a loss in the channel between sender and receiver [Wang et al., 1998]. In case of error, the redundant information can be used to generate the lost parts. FEC codes can be mainly divided into two categories, block codes and convolutional codes. Block codes work on blocks or packets, while convolutional codes work on bit-streams. Reed-Solomon (RS) codes and Hamming codes are classic examples of block codes. For convolutional codes, some popular examples are Low-density parity-check (LDPC) codes and Turbo codes. However, terminated convolutional codes can also be considered as block codes.

For video transmission over lossy networks, methods such as Priority Encoding Transmission (PET) first add the redundancies to messages with different priorities, and then encapsulate them in packets [Albanese et al., 1996]. As a result, PET can be a suitable solution when used along with convolutional codes. However, our focus in this work is on scenarios where first the original packets are generated in the application layer, and afterwards the redundant packets are further added by the underlying FEC. Therefore, the focus of this work is on block codes.

Different FEC methods have different features. For example, Hamming codes are more suitable for low error rate conditions. LDPC codes are better for high code rates, which makes them suitable for large block transfers, while RS codes are more appropriate for small block size and real-time streams. However, RS codes suffer from high computation complexity, while LDPC codes have a high speed coding and are suitable for handheld devices [Nafaa et al., 2008]. The effectiveness of each FEC code strongly depends on the underlying physical layer, the channels error rate and the data rate. Regardless of which FEC method is chosen for the system, we claim that our overlaying protection scheme can function properly as long as the underlying FEC method is a block code. In other words, our scheme can be used with any type of block code. However, in this paper our scheme is primarily examined using RS codes.

RS codes are one of the most classic FEC methods used for video transmission over lossy networks [Girod et al., 1998][Stuhlmüller et al., 2000]. Although the block size of RS codes is constrained to be $2^m - 1$, where $m$ is the
number of bits in each symbol, by using shortened RS codes, any smaller block size can be used. As a result, shortened RS codes provide a great flexibility in assigning different amounts of protection to different block sizes. In practical systems, although the error correction efficiency will increase with larger block sizes, the block sizes need to be limited due to delay constraints. With the increase of block size, the introduced delay by the FEC scheme will also increase, since the sender/receiver have to buffer more packets before being able to encode/decode. Moreover, the computational complexity increases with the increase of block size, which causes an additional reason for choosing small block lengths in practice [Stuhlmüller et al., 2000].

In general, the functionality of block code FEC methods is similar: For each $n$ number of original packets, $k$ redundant packets are generated which together with the original packets form a block of packets with block size $n+k$. In the decoder, the $n$ original packets can be recovered by receiving any $n$ packets of these $n+k$ packets. For stronger protection, more redundant packets should be sent, since a loss of up to $k$ packets is tolerated in each block of packets. If more than $k$ packets are lost, then the lost original packets cannot be recovered anymore. Increasing the number of redundant packets causes more overhead. Therefore, there is always a trade-off between a stronger protection and more bandwidth overhead. However, using the same amount of bandwidth overhead, robustness against loss bursts can be increased by using a larger block of packets. For example, if 1 redundant packet is added to 2 original packets, it may not resist a loss burst of 2 packets. However, if 2 redundant packets are added to 4 original packets (a block size of 6 instead of 3) the original packets would certainly be recovered after a loss burst of 2 packets. However, as mentioned previously, increasing the block size may not always be a practical option.

**Flexible Macro block Ordering:** For a more effective trade-off between the amount of protection and bandwidth, unequal loss protection methods are used which apply a higher amount of protection on the more important parts of the video. The challenge is how to define the importance and also how to distribute the protection based on this importance. For separating different parts of a frame and putting them in different packets, the Flexible Macro block Ordering (FMO) feature of coding standards is used [Wiegand et al., 2003]. FMO is an error resilient tool which enables us to divide the frame into slices in a flexible manner. There are different modes for flexible macro block ordering, each having its own pros and cons. Common modes include raster scan, dispersed, foreground/background and explicit. In raster scan, slicing starts from the top left macro block and macro blocks are assigned to the slice one by one as we swipe each row from left to right and then go to the next row and so on. In dispersed mode, macro blocks are assigned to slices in a distributed manner such that each block is surrounded with macro blocks from other slices. If a frame is divided into 2 slices, the dispersed mode will look like a checker-board pattern. In the explicit mode, there are no restrictions for assigning macro blocks to slices. In the foreground/background mode, several rectangular regions of the frame can be defined and assigned to separate slices.

**ROI Detection:** For identifying the foreground regions, also known as regions of interest or salient regions, several methods have been proposed. In [Osberger et al., 1998] several features that influence visual attention are combined together, to automatically produce an Importance Map that detects perceptually important regions in an image. In [Mavlankar et al., 2007] a ROI tracking method for high resolution interactive video streaming applications is proposed. This method takes advantage of the motion information in the video frames to track the ROI parts in the video. In [Huang et al., 2009] and [Borji et al., 2012] benchmarks for detecting salient regions and regions of interest are offered, and several state-of-the-art ROI detection methods are compared. Recently, in [Lou et al., 2013] Kinect is used to detect ROI for HD video conferencing applications. Moreover, since in video conferencing applications the ROIs are usually the faces of people, finding ROIs in video conferencing applications also falls in the category of face detection. A face detection method for color images is proposed in [Hsu et al., 2002]. In [Viola et al., 2004] a real-time face detection technique is proposed. Surveys of different face detection techniques are provided by [Hjelm9sas et al., 2001] and [Zhang & Zhang, 2010].

**Data Partitioning:** Data Partitioning is another error resiliency tool in which each slice is divided into 3 partitions. Each partition is encapsulated in a separate packet. The first partition, referred to as Header partition, consists of header information and motion vectors. In other words, this partition contains necessary data for decoding the slice. By receiving the Header partition we can make a rough estimation of the slice, even without having the other partitions. The second partition, referred to as Intra partition, contains the data for intra coded macro blocks of that slice. Having the Header and Intra partitions, allows the receiver to decode the intra coded blocks properly. The third partition is the Inter partition which contains the residual data for inter coded macro blocks of that slice. So the Inter partition is needed in order to decode the inter coded blocks properly, although, even without this partition we are able to have a rough estimation of those blocks using the motion vectors in the Header partition. As a result, the three partitions have different amounts of importance. The Header partition is the most important one, since without it, the other partitions cannot be of any use. Intra partition is less important than the Header partition, and Inter partition is the least important one.
2.2 Related Work

In general there are two main approaches to protect a stream of data from packet loss: (i) Automatic Repeat reQuest (ARQ) and (ii) FEC. (i) In ARQ packets are retransmitted in case of loss. As a result ARQ is more efficient in terms of bandwidth but since it 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 [Connie et al., 2008] PR-SCTP has been used to apply a higher level of protection on I frames and Header partitions. In [Sanson et al., 2010], by using PR-SCTP, different levels of protection have been applied to I, P and B frames. In [Porter & 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. In [Zhao et al., 2013] the same TCP/UDP hybrid approach is used, where TCP is used for the Picture Parameter Set (PPS), Sequence Parameter Set (SPS) and the slice headers. Although partial reliability can make retransmission techniques suitable for video, it is still not suitable for interactive video applications which are very sensitive to delay.

The second category of protection (ii), which uses FEC, is more suitable for improving video quality when video packets are exposed to network packet loss. In general this category can be divided into two sub-categories, 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.

For minimizing the distortion sub-category, minimizing the error propagation is one of the main concerns. In [Chowdhury et al., 2013] error propagation was reduced by using the intra-refreshed macro blocks as reference rather than the inter-predicted mode macro blocks that may contain error from previous frames. However in interactive applications intra-refresh is usually disabled, in order to have higher encoding speed and lower bandwidth. Generally, since intra mode search causes a significant increase in the computational complexity of encoding a P-frame, skipping intra modes for higher encoding speed is an effective solution which has been studied in [Lee et al., 2009][Su, 2009][Kim et al., 2006]. Skipping intra modes can slightly degrade the encoding efficiency, but the degradation is relative to the video motion. As a result, in real-time applications where high speed encoding is needed, disabling intra modes improves the overall performance, especially when video motion is low. Therefore, in our scheme we disable intra mode in P-frames, since our focus is on video conferencing applications, where the video motion is low and the encoding is in real-time.

In [Kazemi et al., 2013] a comprehensive review on video protection schemes which use error resiliency techniques such Multiple Description Coding (MDC) can be found. However, these schemes cannot reduce error propagation. 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 [Thang 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 et al., 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 condition gets bad, the number of redundant packets and thus the redundancy bit-rate 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 group of picture (GOP). But since these methods try to minimize the distortion caused by error propagation by using the future frames and calculating the amount of distortion caused by missing each macro block, they cannot be used for real-time applications like video conferencing. In other words these methods assume that we have access to the future frames when we are deciding the protection for a frame, but in real-time applications we have no information about the future frames.

The method used in [Zhang et al., 2009] and [Zhang & Peng, 2009] is suitable for real-time applications. In [Zhang & Peng, 2009] 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. In [Zhang et al., 2009] the GOP is divided into 2 parts but the bit rate of the frame is also taken into account. Neither of these methods have considered the region of interest. In [Zhang et al., 2010] packets are classified similarly based on their partition and position in the GOP, and the amount of protection for each class is calculated adaptively for each GOP using an optimization problem based on the network loss and the amount of error propagation their loss can cause. Again for solving this...
optimization problem future frames in the GOP are assumed to be known. So although their main idea of reducing error propagation is useful for real time applications but their adaptive protection method is not.

There are several other methods in the second sub-category. 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 [Muntean et al., 2008] several levels have been introduced for ROI, which are inside each other in a way that each level is a smaller part of the previous level and contains an even more important part of that level. So as the levels go smaller and deeper they become more important. As the loss rate increases, the transmission rate is decreased in order to avoid congestion. The rate reduction is first done by decreasing the quality of the background and then the upper ROI levels, trying to maintain high quality for the most important parts which are in the smallest level. Similarly, in [Ciubotaru et al., 2009] various ROIs are defined, concentric around the area of maximum user interest. When exposed to congestion, the video bit-rate is reduced adaptively. However, instead of reducing the video bit-rate uniformly, 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 [Boulos et al., 2009] the region of interest is extracted and the macro blocks of the region of interest are coded using the intra-prediction mode to avoid error propagation but at the expense of more bit rate. 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. The Gilbert model has been used as the network loss model. In [Wang et al., 2011] the region of interest is also extracted and protected based on the channel condition. It uses an interlaced protection mode for bad channel conditions and a complete mode for good channel conditions. Its main challenge is implementing an extraFMO in x264 for extracting and protecting the region of interest in x264 which does not support flexible macro block ordering. In [Xue et al., 2008] the region of interest is detected and using discrete wavelet transform its low frequency is extracted and protected.

In [Arachchi et al., 2006] the region of interest is divided into two slices using a checker-board pattern and more protection is applied to them compared to the background slice. Unlike the other discussed methods that basically just put more protection on the region of interest, this method triggers a new idea for improving the region of interest quality which is not related to applying more redundancy. 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 in the same way to the foreground and background parts, but with only one slice of foreground. This improvement is due to the better concealment we can get when using the dispersed mode of flexible macro block ordering. Since each macro block is surrounded by macro blocks from the other slice, if a slice gets lost, its macro blocks can still be concealed using its neighboring macro blocks. It can be concluded that the quality may improve by increasing the number of slices, but the point is that as the number of slices increase the overhead will also increase. As mentioned before, this overhead is mostly because of the packet headers added to each slice and also because of the reduction in coding efficiency due to not crossing the slice boundaries for prediction.

In the core of our proposed ALP scheme, we combine the two sub-categories of region of interest protection and distortion minimizing techniques 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 network condition. This paper 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 the protection modes.

Generally, adaptive video transmission techniques can be classified into two categories: (1) Adaptive techniques that focus on congestion control, such as [Muntean et al., 2008], [Ciubotaru et al., 2009] and [Fernandez et al., 2012], and (2) Adaptive techniques that focus on FEC protection, such as [Zhong et al., 2010], [Xu et al., 2013], [Thomos et al., 2006] and [Zhang et al., 2010]. Basically, there are many situations in the network that can cause packet loss, such as network congestion, or limitations of the physical connection. Network congestion usually happens in the network routers that are not connected directly to the end users. The purpose of congestion control is to prevent packet loss by
reducing the overall bit rate. However, reducing the bit rate might not be useful in cases where there are limitations in the physical connection. Physical connection limitations are common in both wireless and wired networks. For example, a user with an ADSL connection may have a bit rate limitation set by his service provider. Usually, this rate is related to the terms and conditions of the service and not necessarily the physical capacity of the link. When this user is engaged in a video conference and the network condition becomes bad, it may not be a good idea to reduce his/her bit rate, because in most cases, the congestion is not related to the transmission rate of this individual user and therefore, reducing the transmission rate has little or no effect on the overall end to end network congestion. Therefore, keeping the rate fixed and making the best use of whatever bandwidth is available by changing the encoding process to adapt to the network conditions is a solution that can be recommended in many cases. As a result, a real-time video transmission system should include two separate components: congestion control and error control. The congestion control component consists of rate control, rate-adaptive encoding, and rate shaping; while the error control component consists of FEC, retransmission and error concealment [Wu et al., 2000]. A survey of different congestion control and error control mechanisms can also be found in [Wu et al., 2000].

Our focus in this work is on FEC protection. Among the adaptive techniques that focus on FEC, some exploit optimization problems that involve future frames, such as [Thomos et al., 2006][Zhang et al., 2010]. Although optimum, they cannot be used in interactive applications where future frames are not available. To the best of our knowledge, all adaptive FEC techniques that are suitable for interactive environments react to loss by further increasing their FEC packets and thus FEC bit rate. In [Zhong et al., 2010], adaptive protection is applied based on the average burst loss length, and the protection bit rate increases as the average burst loss length increases. In [Xu et al., 2013] when the network condition gets bad, the number of redundant packets and thus the redundancy bit rate increases. The number of redundant packets is set based on a repair performance index which measures the performance of the FEC. The repair performance index is measured by the number of redundant packets over the number of unrecovered packets.

In a practical system, as mentioned previously, there is always a rate control component which dictates the allowed rate. In this case, under a constant allowed rate, any increase in the FEC bit rate means a decrease in the video bit rate and thus the video quality. One of the main features of our ALP scheme is that unlike other adaptive FEC techniques, under a constant allowed rate, it maintains a constant bit rate for FEC, while effectively enhancing the video quality.

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 some of additional points of importance.

3.1 Overview

Our 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 defining the first two modes we use state-of-the-art methods, while for defining the third and forth mode we propose an Integrated Loss Protection (ILP) method. Fig. 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, based on the current mode, the receiver counts the number of zone packets that have been lost and/or recovered throughout the whole GOP. For example if the corresponding zone is ROI, the receiver will count all the lost and/or 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(zone) calculates the percentage of lost packets in that zone that are not recovered by the protection mode. The receiver also calculates the SM(previous zone). SM(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 allocation.
The flow chart for each mode of our proposed adaptive loss protection scheme is shown in Fig. 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, several lossy GOPs are needed in order to reach the ILP_Hloss mode.

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, the same number
of redundant packets that were used in ELP_Frame to protect the whole frame is used, but since the same number of redundant packets is 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 condition has gotten 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 worse and the SM(ROI) has also increased beyond its switching point, the sender would be ordered to switch to the ILP_Loss mode. If none of these is the case, then the sender will keep on sending the frames using ELP_ROI.

In the third mode, named ILP_Loss (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_Loss protection mode. If the network condition is so severe and the SM(ROI_HEADER) increases its threshold, we switch to the ILP_HLoss mode. While, if the network condition has gotten better and SM(ROI) has decreased below its threshold, we would go back to ELP_ROI.

The forth 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 the ILP_HLoss mode we take all the protection from the less important packets and give them to the more important ones.

Both of the ILP_Loss 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 condition gets 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 condition, 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_Loss and ILP_HLoss modes and how to 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 the 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 the packets of the future frames for the protection of the current frame. Otherwise we would be adding an extra amount of delay which 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 which 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 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:

![Diagram showingGOP dividing](image)

**Table. 1. Example of a description table for the FEC distribution.**

<table>
<thead>
<tr>
<th>No. of added packets</th>
<th>Part 1</th>
<th>Part 2</th>
<th>Part 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header partition</td>
<td>4</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>Inter partition</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
**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 still any bandwidth is 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 which 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 to the Header partition classes. So, this scenario gives all the protection bandwidth for protecting the Header partition packets.

As an example, Fig. 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 1 shows an example of the protection distribution for its different classes, 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 loss burst length and packet loss rates 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 in which point 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 finds the point where the current protection mode cannot be effectively used any more. This metric is defined as:

$$
SM(\text{zone}) = \frac{\text{Unrecovered losses in zone}}{\text{All losses occurred in zone}} \times 100,
$$

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_Loss 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_Loss and ILP_Hloss, since ILP_Hloss only protects the Header partitions of the ROI, a suitable switching point is the point where ILP_Loss 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_Loss and ILP_Hloss modes. SM(ROI_HEADER) means the percentage of lost Header partition packets of ROI, that could not be recovered by ILP_Loss.

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 quality of the decoded video. 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 above mentioned 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 in each frame. 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 which 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 switching occurrence, 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 ranges of packet loss rates and even out performs each of the 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 Video Test Media][Video Quality Experts Group (VQEG)][ HD-VideoBench]. 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 [TestVid][Ultra Video Group][Elemental][EBU][Consumer Digital Video Library][Mannheim University Test Sequences], 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 30fps, 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 [JM 18.0]. 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 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 table is 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 which 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 \( \alpha \) and \( \beta \). 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 \( \alpha \) and \( \beta \) 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 \( \alpha \) and \( \beta \) 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 2.4, 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.
Packet Loss Rate = $\beta / (\alpha + \beta)$

Average Burst Loss Length = $1 / \alpha$

(1)

**Performance metrics:** For measuring and comparing decoded video qualities at the receiver, 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. SSIM is a complex but accurate quality metric which is based on pixel inter-dependencies [SSIM]. 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 sub-categories 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 sub-categories. In [Arachchi et al., 2006] as a representative of the ROI protection schemes, the ROI is divided into two slices using a checker-board 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 attempts 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 & Peng, 2009], as a representative of the distortion minimizing techniques which 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 & 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 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 2.5 dB compared to ULP_Frame and about 1.5 dB compared to ROI_Only.

![Fig. 4](image_url)

(a) Quality in terms of PSNR.  
(b) Quality in terms of SSIM.

Fig. 4. Our ALP scheme outperforms both ROI protection methods, and unequal loss protection methods that don't consider ROI.
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 the quality. The components are the four modes ELP_Frame, ELP_ROI, ILP_Lloss and ILP_Hloss, each designed for a different network condition. 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 2 and 3 show 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%.

Table 2. Description table of the FEC distribution for the ILP_Lloss mode of the first video sample.

<table>
<thead>
<tr>
<th>ILP_Lloss</th>
<th>No. of added packets</th>
<th>Part 1</th>
<th>Part 2</th>
<th>Part 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header partition</td>
<td>3</td>
<td>3</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Inter partition</td>
<td>3</td>
<td>1</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 3. Description table of the FEC distribution for the ILP_Hloss mode of the first video sample.

<table>
<thead>
<tr>
<th>ILP_Hloss</th>
<th>No. of added packets</th>
<th>Part 1</th>
<th>Part 2</th>
<th>Part 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header partition</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td></td>
</tr>
<tr>
<td>Inter partition</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

We test our full adaptive loss protection scheme, and the simulation results are shown in Fig. 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 of the results is presented in the figures below.

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 condition 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 which suits it the best. So the average quality increases and the variance between them is decreased. Table 4. 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.

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 under more protection but not more than enough, so that some protection would remain for the less important classes if possible.

Table 4. Our ALP scheme achieves a lower standard deviation of quality for all PLRs, compared to the 4 different modes.

<table>
<thead>
<tr>
<th>Standard Deviation of PSNR (dB)</th>
<th>PLR=1%</th>
<th>PLR=5%</th>
<th>PLR=10%</th>
<th>PLR=15%</th>
<th>PLR=20%</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our ALP Scheme</td>
<td>0.3</td>
<td>0.84</td>
<td>0.80</td>
<td>0.84</td>
<td>0.57</td>
</tr>
<tr>
<td>ELP_Frame</td>
<td>0.37</td>
<td>1.55</td>
<td>1.55</td>
<td>1.5</td>
<td>1.40</td>
</tr>
<tr>
<td>ELP_ROI</td>
<td>0.82</td>
<td>1.47</td>
<td>1.50</td>
<td>1.18</td>
<td>1.04</td>
</tr>
<tr>
<td>ILP_Lloss</td>
<td>0.9</td>
<td>1.11</td>
<td>0.99</td>
<td>1.28</td>
<td>1.01</td>
</tr>
<tr>
<td>ILP_Hloss</td>
<td>1</td>
<td>0.95</td>
<td>0.98</td>
<td>1.04</td>
<td>1.06</td>
</tr>
</tbody>
</table>

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

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. Fig. 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. Fig. 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 see that ELP_ROI achieves a better quality. Fig. 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 receive most of the Header partitions correctly.
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. 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. Fig. 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. Fig. 9 (a) is with respect to our SM(Frame) metric, while Fig. 9 (b) is with respect to the packet loss rate. It can be seen that when 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. However, 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 switching occurs, would increase.

![Fig. 9: Quality Comparison](image)

The next experiment is for testing the switching point between the ELP_ROI and ILP_Lloss modes. Since ILP_Lloss is data partitioned, we have used the average of lost Header and Inter partitions of ROI, as the number of lost ROI packets. Fig. 10, shows the quality of these two modes around the switching point, with respect to the

![Fig. 10: Quality Comparison](image)
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.

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 Fig. 11, which again 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 occurs. 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 in 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 paper we have proposed an Adaptive Loss Protection (ALP) scheme with negligible delay for protecting interactive video applications, such as video conferencing. This scheme doesn't 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 between these modes. The ALP scheme is stable against sudden and short network changes, since it avoids changing the protection suddenly and dramatically.

We have proposed a ILP method for loss protection which 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. We have proposed a systematic way of creating these modes so that they can be automatically made by the system.

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 will 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 assure us that a better quality will be achieved after each switching event.

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 0.7 dB.
REFERENCES


Consumer Digital Video Library (CDVL). http://www.cdvlib.org/


EBU. https://tech.ebu.ch/hdvi technician


Mannheim University-Test sequences. http://ls.wim.uni-mannheim.de/de/p4research/projects/retargeting/test-sequences/


Xiph.org Video Test Media [derf's collection]. https://media.xiph.org/video/derf/


