NETWORK LAYER FEEDBACK ENABLED ADAPTIVE APPLICATION-LEVEL REROUTE

Liping Guo Gouri Landge
Department of Electrical & Computer Engineering
University of California, Davis

ABSTRACT

In this paper, we propose a network layer feedback enabled application layer reroute scheme to alleviate the quality degradation of the streamed video in the presence of link failures. In our scheme, we assume synergy feedback framework, proposed in [1], is used to provide quick and efficient network feedback to the multimedia server. The proposed application-level reroute scheme is evaluated against state-of-the-art Multiple Description Coding (MDC) and Path Diversity solution through comparing the decoded video quality (PSNR) at the receiver side. We also introduce a simple video streaming simulation system, which is developed based on an available scalable codec: In-Band Motion Compensated Temporal Filter (IBMCTF). Finally, we analyze the obtained simulation results and discuss possible future research.

1 INTRODUCTION

Delivering video content under a lossy network condition presents a great challenge due to the stringent requirement of the streaming media: video application is delay sensitive, and packets with delay exceeding a certain time threshold (e.g. 2 seconds) are treated as if they were lost; video application is loss tolerable, but burst loss can severely degrade the video performance at the receiver side. To combat with packet loss and other video streaming unfriendly network conditions, various solutions have been proposed among which Multiple Description Coding and Path Diversity (MDC-PD) is considered as an effective solution [2] [3]. However, depending on the underlying network conditions, the MDC-PD solution may lead to "overprotection." If the network condition is fairly stable and link failure occurs only occasionally, sending multiple correlated descriptions for the same video content over diverse paths is very inefficient in terms of network resource utilization.

In this paper, we propose a network feedback enabled application-level reroute scheme to cope with the occasionally occurred link failures for video streaming. Our application reroute scheme heavily relies on the network layer feedback system (NLFS) described in [1]. The NLFS provides quick, efficient, and real-time feedback about the network conditions, such as link failure, to the multimedia server. Upon receiving the feedback report, the media server adaptively switches to a lower quality video stream to compensate the delay caused by the link failure and reroutes the video stream through a backup path. It is important to note that the proposed scheme is not intended to be an alternative to the MDC-PD solution. In contrast, in combating with packet losses due to link failures, our scheme can be used as a complementary solution to the MDC-PD. More specifically, the solution presented here is more suitable in a situation where link failures do not happen often and the MDC-PD could potentially cause waste of network resources due to unnecessary protection.

The paper is organized as follows: section 2 provides a brief overview of our network layer feedback enabled application layer reroute scheme; in section 3, we introduce a simple video streaming system used in our simulation; section 4 contains description of detailed simulation procedure; In section 5 we present our simulation results; and the conclusion is presented in section 6.

2 THE APPLICATION LAYER REROUTE SCHEME

2.1 *Overview of NLFS*

In [1], the Network Layer Feedback System is supported by a synergy feedback architecture in which an overlay node called "Overlay Broker" is designated to maintain information of all other overlay nodes inside each domain. Before initiating a video session with a client, a multimedia server needs to register with the nearest overlay broker in its domain. To provide feedback to the server, a new logical layer, called *Synergy Layer*, is created on top of the IP layer and deployed in every router in various domains in the Internet to form a synergetic association with other layers.

When a failure event is detected by the optical layer in the router adjacent to the failure event, the failure event is passed to the synergy layer without any additional delay. The synergy layer then passes the failure info to either its upstream node or downstream depending on the IP-address of the neighboring node that has lost its connectivity. Once the overlay broker gets the failure info, it informs all registered multimedia servers that are using the affected path about the failure.

Simulation results presented in [1] shows that a significant decrease in feedback time by using a synergy feedback mechanism. The maximum delay experienced by the server with synergy feedback is approximately 0.26 seconds.

2.2 Adaptive Application-level Reroute

The application-level reroute scheme presented here can be viewed as an application of the *NLFS* described in [1]. It includes two components: routing and rate adaptation. In this paper, we focus on the rate adaptation component of the proposed reroute scheme.

2.2.1 Routing component

We propose that a routing table-like structure is maintained at each media server for every video session, which contains all possible alternative routes. When failure information (e.g. IP-address of the neighboring node that has lost its connectivity) reaches the server, the backup routing path is selected to avoid going through the failed node. Then the routing technique similar to Loose Source Record Route (LSRR) can be applied to reroute the video content.

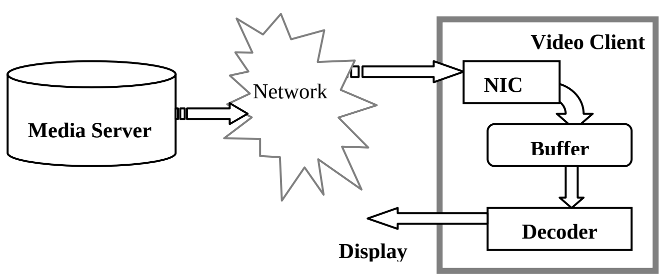

Figure 1: A simple video streaming system

2.2.2 Rate adaptation component

A simple video streaming system shown in Figure 1 is assumed here. A play out buffer is employed at the streaming client to overcome delay jitter. Commonly, a streaming media client performs a 5 to 15 second buffering before playback starts [4]; here, we assume 2 second buffering to reduce the simulation time. Once playback starts, the buffer is maintained at an equilibrium status, which means that the number of frames stored in the buffer should kept constant during the play out. In the event of link failure, delay due to failure feedback, reroute process, and propagation can break buffer's equilibrium status. In the worst case, the play out buffer is overplayed to empty and only a frozen screen can be displayed. Thus, from the streaming client's perspective, good solution to link failure is the one that enables the play out buffer to recover its equilibrium status fast enough such that no buffer overrun would happen.

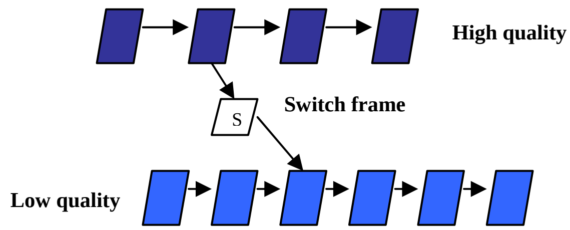

Figure 2: A switch from high bitrate to low bitrate

To avoid severe video quality degradation at the receiver side (i.e. frozen screen due to buffered frames running out) during the event of link failure, the media server in our proposed scheme can dynamically switch from original high bitrate stream to certain low bitrate stream (see Figure2), depending on the available bandwidth of the backup path. This high quality to low quality switch allows more frames to be delivered to the receiver's play out buffer. Hence, the running out of frames scenario is avoided and receiver's play out buffer can quickly recover its equilibrium status. To reduce server's processing time, a bitrate switch table, which contains the pre-calculated bandwidth-bitrate pair, is maintained by the server. Table 1 is an example bitrate switch table.

Backup path BW	New bitrate
640 kbps	468 kbps
768 kbps	512 kbps
1024 kbps	749 kbps
1280 kbps	937 kbps

Table 1: An example bitrate switch table

In order to clearly illustrate how the bandwidth-bitrate pair is calculated, some notations are defined below:

$\mathbf{T_{oc}}$: buffer over-consume time includes failure feedback delay (max 0.26s), routing process time, and 1/2 RTT (see Figure 3).

$\mathbf{T_{rf}}$: buffer refill time refers to the time spent on refilling the over consumed frames.

$\mathbf{T_{p}}$: routing process time.

$\mathbf{N_{oc}}$: number of over consumed frames during T_{oc}

$\mathbf{R_{o}}$: original streaming bitrate.

$\mathbf{R_{n}}$: newly adapted streaming bitrate.

$\mathbf{B_{n}}$: available bandwidth of the chosen backup path.

$\mathbf{R_{p}}$: play rate at the receiver side.

$\mathbf{Q_{o}}$: quality of the original video stream measured by number of bits per encoded frame.

$\mathbf{Q_{n}}$: quality of the newly adapted video stream.

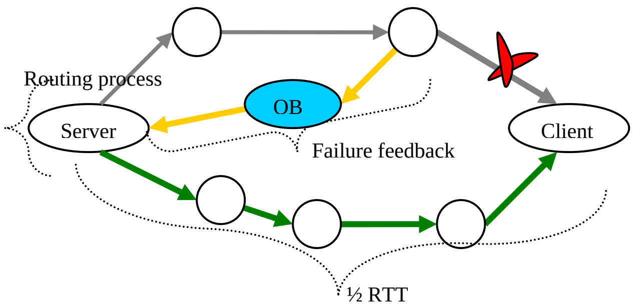

Figure 3: Failure feedback delay. OB: *Overlay Broker*

We assume that the media server can refill the over consumed frames inside the play out buffer within 1 second (T_{rf} =1s), and the play rate at the client side is fixed. Then, the newly adapted bitrate can be calculated as follows:

(1) $\mathbf{R_n = R_p * Q_n}$ (kbps)

(2) $\mathbf{Q_n = B_n / (R_p * 1\ sec + N_{oc})}$ (kbpf)

(3) $\mathbf{N_{oc} = R_p * T_{oc}}$

(4) $\mathbf{T_{oc} = 0.26s + T_p + ½\ RTT}$ (s)

3 A SIMPLE VIDEO STREAMING SYSTEM

3.1 Overview of the IBMCTF

In-Band Motion Compensated Temporal Filtering (*IBMCTF*) is a framework for fully-scalable video coding that performs open-loop motion compensated temporal filtering (*MCTF*) in the wavelet domain (in-band) [5]. The video streaming system used in our simulation is developed based on an available IBMCTF framework. In this section, we briefly introduce the operation of *IBMCTF*.

IBMCTF consists of three major components: an encoder, a pull function, and a decoder. The encoder compresses the input video sequence once according to the compression requirements specified in a parameter file, such as temporal-decomposition level, search range, and spatial decomposition level. Once the compressed video file is ready, the pull function can be called to pull the bit stream from the compressed file at different user preferred bitrate. The product of the pull function is a text file called "hinting file," which basically is a description of the packetized compressed video file. Finally, the decoder reconstructs the compressed video file based on the hinting file created by the pull function.

3.2 The simulation system

The simulation system is a slight augmentation of the *IBMCTF* framework. As shown in Figure 4, the network condition (here, link failure) is injected by changing the transmission status of certain packets in the hinting file. Of course, the decision on which packet gets dropped should comply the real network condition. For example, to simulate a path with 5% loss rate, a two state Markov-chain is applied on every packet; simulation is conduct 20 times, and the resulting PSNR values are averaged to obtain the final PSNR value.

The addition to the existing *IBMCTF* framework is a loss injection unit, called "*Loss Injector*". It simply contains two parts: a hinting file parser and a hinting file modifier. During simulation, the pull function produced hinting file is feed into the Loss Injector; and the loss injected hinting file is then given to the decoder to produce decompressed video file and calculate the PSNR value.

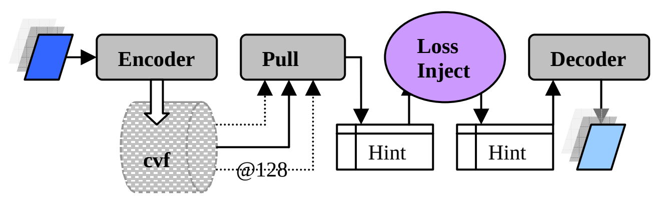

Figure 4: A simple simulation system

4 SIMULATION

In this section, we describe the detailed simulation procedure. We assume playback starts at the receiver side after it buffers 2 seconds data, in our case, 60 frames. We use video sequence Akiyo at cif resolution as our input video file. We encode the first 288 frames of the input video file. The play out rate at the client side is fixed at 30 frames per second. We choose 0.26 second as an average failure feedback delay [1], 0.04 second as routing process time, and 150 millisecond as round trip time (RTT).

4.1 Realization of rate adaptation

The compressed video file is pulled at rate of 768 kbps and decoded using the original hinting file generated by the pull function. We assume that the available bandwidth of the chosen backup path is the same as the original path, here 768 kbps. As described in section *2.2.2*, the bandwidth-bitrate pair is calculated as follows:

T_{oc} = 0.26s + T_p + ½ RTT = 0.26s + 0.04s + ½(0.15s) = 0.38s

$N_{oc} = R_p * T_{oc}$ = 30fps * 0.38s = 11 frames

$Q_n = B_n /(R_p$*1sec+ $N_{oc})$

=768kbps/(30+11)frames=18.73 kbpf

$R_n = R_p * Q_n$ = 30 fps * 18.73kbpf = 562 (kbps)

The compressed file is pulled again at the new bitrate, here 562 kbps. The produced hinting file for pull at 562kpbs is used to decode the compressed file. The decoder calculates and reports the PSNR value on per frame basis for both original high bitrate stream and the adapted low bitrate stream. To illustrate the effect of the bitrate switch or quality adaptation at the client side, we simply replace the PSNR values of the original frames that are affected during adaptation by the low bitrate frames' PSNR values. In this case, the PSNR values of 41 frames of the original high bitrate are replaced.

4.2 *Realization of the MDC-path diversity Solution*

MDC-path diversity is used to provide error resilience to multimedia streaming over lossy network. In MDC-PD, the video is coded and transmitted separately over different paths, and each of these descriptions is independently decodable but complementary to each other. The receipt of all the different descriptions offers the reconstructed video the best quality; and if any of the description is lost, the stream can still be decoded, of course, with less quality. By explicitly sending different descriptions over different network paths, MDC-PD minimizes simultaneous losses on multiple descriptions. In the event of link failure, the loss of one description over the failed path will affect the video quality; but the transmitted video can still be reconstructed due to the fact that the losses on different paths are often uncorrelated. However, to make each description independently decodable, redundant information must be send along with each description. In other words, MDC-PD explores the tradeoff between compression performance and error resilience.

As described in section 3.1, the pull function of IBMCTF generates a 'hinting file' for every pulling bitrate. This file contains the information about the packet attributes as well as the status of packet transmission as seen by the receiver. To simulate the effect of MDC-PD at the receiver, we use the *Loss Inject* unit (see Figure 4) to parse and modify the hinting file. To achieve unequal protection, we prioritize the packets by assigning different weights to them based on their attributes shown in the hinting file. Weight of a packet indicates its significance in the bit stream. Packets with weight above certain threshold are coded in both the descriptions. To satisfy the bandwidth constraint, the same number of packets with weight below certain threshold is discarded. The detailed criteria for weight assignment are as follows:

(1) Frame Type: frames are categorized as I-Frame (Intra-coded Frame) or H-Frame (Inter-coded Frame). Intra-coded frame is compressed by exploring the spatial redundancy within the frame. This type of frames does not depend on any other frame and is used as reference by several other inter-coded frames. The loss of I-frames can cause catastrophe to the decoded video. Hence, Intra-coded frame is the most significant frame. Packets belonging to I- Frames are assigned high weight (400). Inter-coded frame makes use of the temporal redundancy among neighboring frames. These frames may contain motion vector information. However, the inter-code

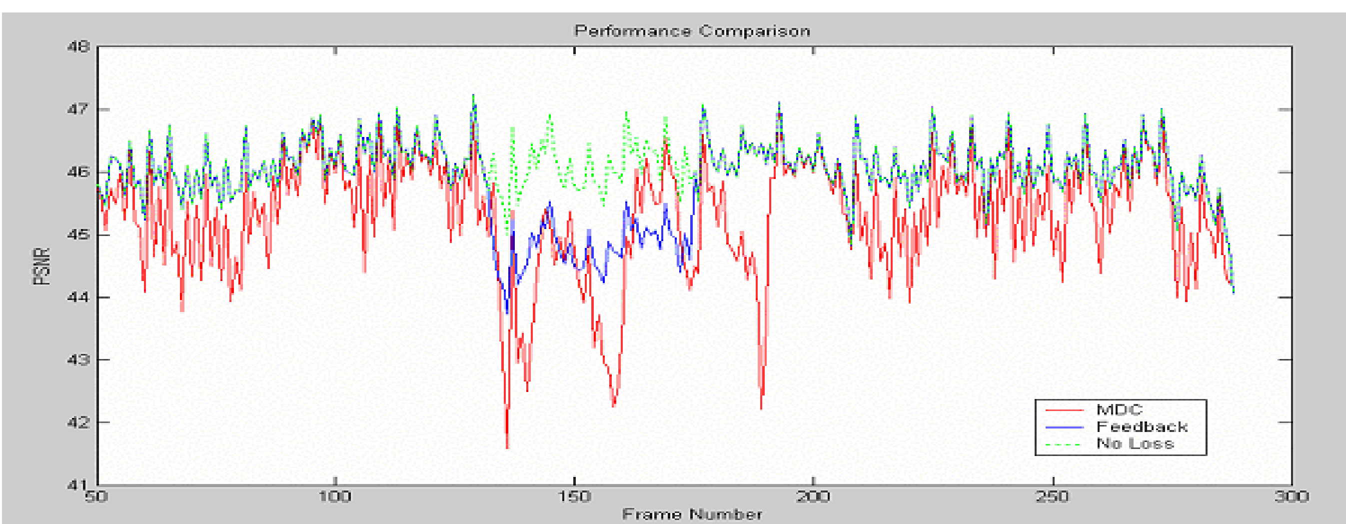

Figure 5: Performance comparison

frames are dependent on I frame and other H frames. Here, we assign weight of 200 to H frames.

(2) Sub Chunk Number: Sub chunk number indicates the packet dependency. For instance, the number of previous packets that the current packet is dependent on. Thus, the packet is less significant with higher subchunk number since the lost packet can easily

recovered by using its parent packets. Sub chunk weight is further added to the weight of the packets.

(3) Cumulative weight of a packet is compared against the high threshold and the low threshold. Packets with weight more than the high threshold are added to both descriptions of MDC. Equal number of packets with weight less than the low threshold is discarded. Packets with weight between the higher and low thresholds are evenly distributed between the two descriptions such that both the streams can be decoded independently.

5 RESULTS AND ANALYSIS

In Figure 5, we show a performance comparison if MDC-PD vs. our feedback enabled application reroute scheme. The y-axis is the PSNR value and x-axis is the frame number, which is placed according to the play order at the receiver side. The link failure is injected at frame 72, the effect starts to emerge after 2 seconds or 60 frames. Without any protection mechanism, the video quality degrades dramatically for the duration of the transient period of the link failure, as shown by the black signal in Figure 6. It is clear that both MDC-PD and our application-level reroute scheme can achieve better video performance in the event of link failure. However, under normal condition (no link failure), MDC performance is about 1dB below the performance of Feedback Method; and this is mainly due to source splitting and information redundancy of the MDC solution.

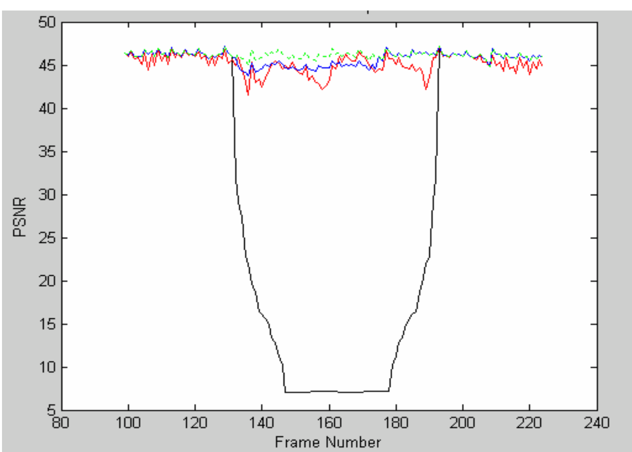

Figure 6: Link failure effect on no-protection video stream

In Figure 7, we zoom out the plot portion that indicates the effect of link failure. It is obvious that, using MDC, the receiver experiences the lower PSNR for the entire duration of the transient period for a link failure event, which can last anywhere from 2 seconds to 6 seconds although we use 2 seconds here. When applying our feedback enabled reroute method, the lower PSNR is experienced only for a short duration which is independent of the transient period. Moreover, in our feedback enabled reroute method, the quality of video during the recovery period can be tuned to fit the available backup path bandwidth and the receiver preferred buffer refill time (*refer to section 2.2.2*).

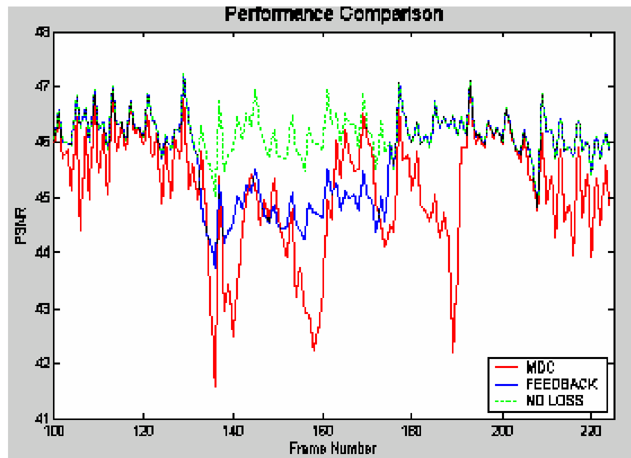

Figure 7: Zoom out version of Figure 5

6 CONCLUSIONS

To cope with packet loss during link failure events, previous research suggested MDC-PD which explicitly sends different descriptions over different network paths to achieve error resilience. However, MDC-PD could result inefficient utilization of network resource if link failure happens only infrequently. To avoid "overprotection," we propose a feedback enabled application-level reroute scheme, which sends single description during normal network conditions; when link failure occurs, the reroute scheme can dynamically switch to lower quality streams and use backup path to continually deliver the content. As we noted in the beginning of the paper, the proposed method is not intended to be the replacement of MDC-PD. Instead, we believe our reroute scheme can be used as a complementary solution to MDC-PD. In other words, our scheme is more suitable for the cases where network conditions are stable and link failure happens occasionally.

REFERENCES

1. R. Keralapura, C. N. Chuah, M. van der Schaar, C. Tillier, an B. Pesquet-Popsecu, "Adaptive Multiple Descriptions Scalable Video Coding Using Network Layer Feedback."

2. J. Apostolopoulos, T. Wong, W. Tan, and S. Wee, “On multiple description streaming with content delivery networks, “ in Porceedings of the IEEE *INFOCOM Conf*., 2002, pp. 1736-1745.
3. J. Chakareski and B. Firod, “Rate-distortion optimized packet scheduling and routing for media streaming with path diversity,” IEEE DCC 2003.
4. J. Apostolopoulos, W. Tan, S J. Wee, “Video Streaming: Concepts, Algorithms, and Systems,” HPL-2002-260, September 18th, 2002.
5. Y. Andreopoulos, A. Munteanu , J. Barbarien, M. van der Schaar, J Cornelis and Peter Schelkens, “In-Band Motion Compensated Temporal Filtering”, submitted to EURASIP – Image communication, Special Issue on Wavelet Coding, June 2003.
6. G. Iannaccone, C. N. Chuah, R. Mortier, S. Battacharrya, and C. Diot, "Analysis of Link Failures in an IP Backbone," *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseille, France, pp. 237-242, November 6-8, 2002
7. Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Christophe Diot, “Characterization of Failures in an IP Backbone”, to appear in *Proceedings of IEEE INFOCOM*, March 2004.