

# Video Streaming over IP Networks

Shivakumar Ramu

*Department of Computer Engineering, Florida Atlantic University, Florida, USA.*

[shivakmr@yahoo.com](mailto:shivakmr@yahoo.com)

## Abstract

*Transmitting a high quality, real time interactive video over the IP network is very challenging, because of limited bandwidth and high packet loss rates. The loss of packets in a video frames leads not only to reduce the quality of video but also results in the propagation of distortion to successive frames. While latency requirements do not permit retransmission of all lost data, we leverage the characteristics of MPEG-4 to selectively retransmit only the most important data in the bit stream. When latency constraints do not permit retransmission, a mechanism for recovering this data using post-processing techniques at the receiver. Quantify the effects of packet loss on the quality of MPEG-4 video, develop an analytical model to explain these effects, present a system to adaptively deliver MPEG-4 video in the face of packet loss and variable Internet conditions, and evaluate the effectiveness of the system under various network conditions. In this paper, we investigate the behavior of packet loss in Internet Protocol (IP) networks with transport protocol: User Datagram Protocol (UDP). Our analysis reveals that the packet loss behavior depends on the underlying transport protocol and that the packet loss in UDP transfers exhibits long-range dependence over the coarser time-scales.*

## 1. Introduction

Streaming is a method of making the Video, audio and other multimedia available relatively quickly via the Internet. The advantage of streaming is that it can enable easier and more responsive on-demand access to multimedia resources. Perhaps even more exciting is the possibility of integrating video and audio with other web-based resources. With streaming, the file remains on the server. The initial part is copied to a buffer on the computer and then, after a short delay, starts to play and continues as the rest of the file is

being pulled down. Streaming provides a steady method of delivery controlled by interaction between the computer and the server.

Packet loss, packet delay, and packet delay jitter are important Quality of Service parameters in video transfers. Large values for any of these parameters degrade the quality of the video application. We present here a detailed study of loss patterns that, in addition to the loss probabilities, may help characterize the loss behavior in packet networks. Streaming media is becoming increasingly prominent on the Internet. Although some progress has been made in media delivery, today's solutions (e.g., Real Player and Windows Media Player) are proprietary, inflexible, and do not provide the user with a pleasant viewing experience. The lack of an open framework hampers innovative research, particularly in the area of video delivery that adapts to changing network conditions.

While today's streaming applications are closed and proprietary, the emerging MPEG-4 standard is gaining increasing acceptance and appears to be a promising open standard for Internet video. In this paper, we describe a system that enables the adaptive unicast delivery of streaming MPEG-4 video by responding to varying network conditions.

Network delay is essentially composed by two sub-components: propagation or transmission, and queuing delay. The propagation delay depends on the network capacity and on the size of the data to transmit. Conversely, the queuing delay is unknown a-priori and it is also very variable, as data travel from source to destination along a path that is usually shared among traffic generated by other applications. Hence, it may happen that a network resource along the path is busy, causing the data to be delayed until the resource is available.

This paper primarily focuses on techniques to deal with packet losses, which are common on the Internet. Inter-frame video compression algorithms such as MPEG-4 exploit temporal correlation between frames to achieve high levels of compression by independently coding reference frames, and

representing the majority of the frames as the difference from each frame and one or more reference frames. However, these algorithms suffer from the well-known propagation of errors effect, because errors due to packet loss in a reference frame propagate to all of the dependent difference frames. The resulting stream is not even resilient to small amounts of packet loss. There is a fundamental tradeoff between and width efficiency (obtained by compression) and error resilience (obtained by coding or retransmission). Inter-frame compression schemes (such as MPEG-4) achieve significant compression of bits in comparison to other schemes that do not exploit temporal correlation, but they are also less resilient to packet loss because of the dependencies that exist between data from different frames. While many methods have been proposed to add redundancy to the bitstream to allow for more effective error correction, they also reduce much of the gains garnered from compression.

Errors in reference frames are more detrimental than those in derived frames due to propagation of errors and should therefore be given a higher level of protection than other data in the bitstream. One solution is to add redundancy to more important portions of the bitstream, or to code more important portions of the stream at a relatively higher bitrate; however, this approach reduces compression gains and in many cases does not adequately handle packet losses that occur in bursts. Prior work has gathered experimental results that describe packet loss characteristics for MPEG video and suggest the need for better error recovery and concealment techniques. Motivated by prior analysis, as well a general model we have developed to explain the effects of packet loss on MPEG video, we have developed a system that uses receiver-driven selective reliability in conjunction with receiver post-processing to efficiently recover from packet losses in reference frames. Some researchers have argued that retransmission-based error resilience is infeasible for Internet streaming because retransmission of lost data takes at least one additional round-trip time, which may be too much latency to allow for adequate interactivity. However, because of the nature of inter-frame compression, certain types of packet loss can be excessively detrimental to the quality of the received bitstream. We show that such losses can be corrected via retransmission without significantly increasing delay, using only a few frames' worth of extra buffering. In a streaming system that transports video bitstreams with inter-dependent frames; careful retransmission of lost packets provides significant benefits by alleviating the propagation of errors. This system primarily focuses on the use of

selective retransmission for packet loss recovery; we also show how the system allows selective retransmission to be used in conjunction with other error control and concealment techniques. When delay or transient loss is prohibitively high, retransmission of lost packets may not always be feasible. In these circumstances, a mechanism for recovering data in reference frames using post-processing at the receiver. To recover from packet losses, this system uses application-level framing (ALF). Because dealing with data loss is application-dependent, the application, rather than the transport layer, is most capable of handling these losses appropriately. Moreover, in the case of video, the receiver is best-equipped to make decisions with regard to packet loss recovery (e.g., whether to request a retransmission, to use post-processing and error concealment or simply to drop the frame). The ALF principle states that data must be presented to the application in units that are both meaningful to that application and independently process-able. These units, called application data units (ADUs), are also the unit of error recovery. This philosophy is backwards-compatible receiver driven selective retransmission extension to RTP called SR-RTP. This extension provides semantics for requesting the retransmission of independently processible portions of the bitstream and a means for reassembling fragmented portions of independently processible units. ALF allows the application to be notified when incomplete frames arrive and control error concealment decisions.

## 2. Problem Description.

### 2.1.1 MPEG4 Background

The MPEG-4 compression standard achieves high compression ratios by exploiting both spatial and temporal redundancy in video sequences. While spatial redundancy can be exploited by simply coding each frame separately (just as it is exploited in still images), many video sequences exhibit temporal redundancy, as two consecutive frames are often very similar. An MPEG bitstream takes advantage of this by using three types of frames.

The MPEG4 video frame structure is as shown in Figure 1. It consist of, I picture, intra coded, coded independently of other frames, P or predictive picture, predicted from the previously decoded picture and B or bi-directionally predictive picture, predicted from one previous and future picture. The arrows in the picture shows, how the frames are resulted.

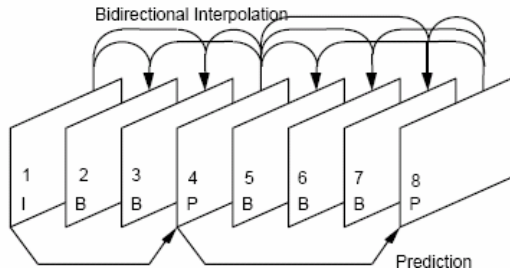


Figure 1: MPEG4 Video frame structure.

### 2.1.2 Propagation of errors.

The ability to successfully decode a compressed bitstream with inter-frame dependencies depends heavily on the receipt of reference frames (i.e., I-frames, and to a lesser degree P-frames). While the loss of one or more packets in a frame can degrade its quality, the more problematic situation is the propagation of errors to dependent frames. An example of error propagation is shown in Figures 2 and 3; the rectangular patch near the bottom of Figure 2 is the result of a single loss in an I-frame. This error spreads to neighboring frames as well, as shown in Figure 3, which depends on several preceding differentially coded frames. Figure 4 shows the evolution of frame-by-frame PSNR for the luminance component as a function of the original raw frame number for various packet loss rates. As the packet loss rate increases, the quality (in PSNR) of an increasing number of the decoded frames becomes too poor for viewing. We generalize this in Figure 4 by averaging the observed frame rate over time for a given video sequence.



Figure 2: I-Frame #48 from the "coastguard" stream with packet loss. Y PSNR: 21.995697



Figure 3: B-Frame #65 from "coastguard" stream showing propagation of errors. Y PSNR: 17.538345

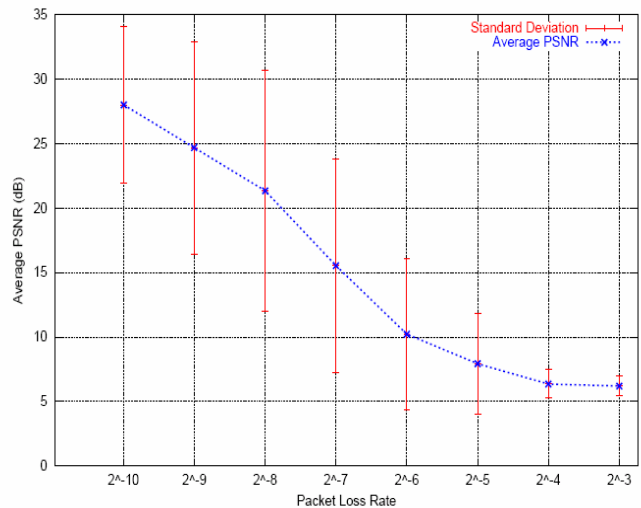


Figure 4: Average PSNR over 100 seconds of 30 fps video as a function of the packet loss rate. As the packet loss rate increases, the overall quality of the bitstream severely degrades.

### 2.1.3 Selective reliability reduces packet loss.

The preceding section shows that packet loss substantially affects the frame rate of a received video sequence and would like to somehow recover some of these packet losses. While it would be preferable to be able to recover all packets, the latency of the network, as well as bandwidth constraints, limit this possibility. Fortunately, the structure of an MPEG-4 bitstream allows us to capitalize on the notion that some data is more important than others. By recovering some of the

more important data in the bitstream, we can substantially increase the observed frame rate.

Figure 5 shows the effects of recovering lost I-frame packets via retransmissions on the effective frame rate for a given PSNR threshold of 20 dB. Recovering I-frame data can increase the effective frame rate significantly, in some cases by up to three times the frame rate without recovery. One upper curve shows the effects of recovering only I-frame data, whereas the other curve shows the effects of recovering only P-frame data. In both cases, the frame rate is significantly increased; this shows that recovering only the I-frame packets in a group of pictures results in comparable gains to recovering all P-frame data across a group of pictures. Therefore, by recovering either the I-frame data or the P-frame data via selective reliability, it is possible to significantly improve the quality of received video—there is no real need to recover all missing packets. Furthermore, recovering missing B-frame packets is not particularly useful.

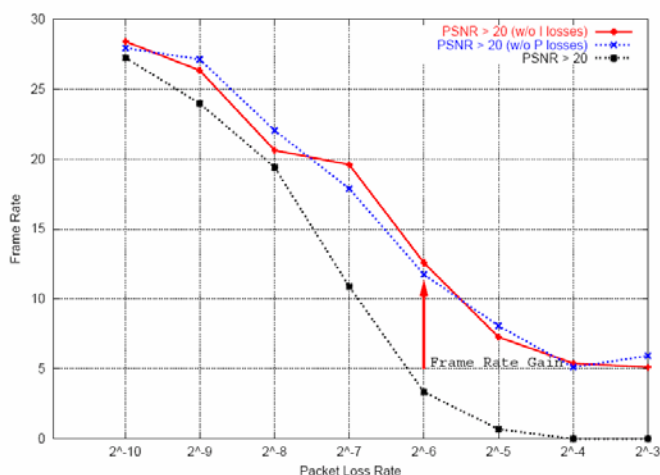


Figure 5: The effects of recovering reference frame data on frame rate. By recovering packet losses in I-frames, the frame rate for a given acceptable quality can be increased up to 3 times. This graph also shows that recovering all P-frames is roughly as effective as recovering only I-frame data.

### 3. System Architecture.

Figure 6 shows the components of the system. The server listens for requests on an RTSP port, establishes session parameters via SDP, and streams requested data to the client via RTP (over UDP) that has been extended to support application-level framing (ALF) and selective reliability. Feedback is provided to the server at the RTP layer via RTCP receiver reports, and the server adjusts the congestion window size using

the Congestion Manager (CM). The CM implements a TCP-friendly congestion control algorithm for the MPEG-4 streams and provides an API by which the server adapts to prevailing network conditions. This system supports the backwards-compatible extensions to RTP/RTCP that allow for the application-level framing of the data with Application Data Units (ADU). ADUs enable fragmentation and reassembly of independently processible units of data and also make selective recovery of application-specific data units possible at the receiver. For MPEG-4, one frame of the compressed video bitstream corresponds to one ADU. The SR-RTP layer divides the data to be sent into ADUs according to frame boundaries. The sender packetizes these ADUs, which are reassembled by the receiver and passed to the application layer for decoding once the complete frame has been received. An ADU may involve multiple packets (i.e., ADU fragments). Each packet is uniquely named by its ADU sequence number and byte offset within that ADU in order to efficiently request and perform selective retransmissions.

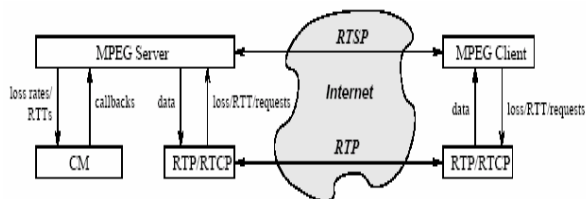


Figure 6: System architecture. Feedback is sent to the streaming application via RTCP, which is used to appropriately adjust the transmission rate.

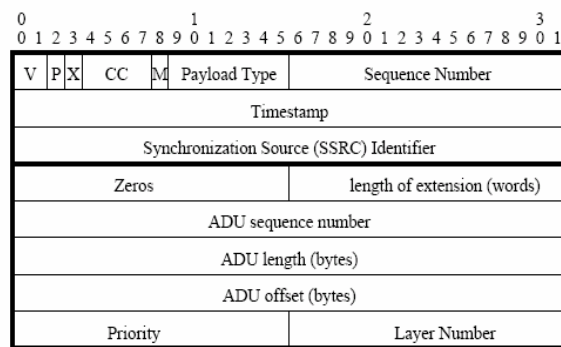


Figure 7: SR-RTP header for selective reliability.

#### 3.1 Header format.

Figure 7 shows the extended RTP header with extension to enable selective retransmission of lost packets. The first 12 bytes of the packet are identical to

the RTP header format specification. Selectively reliable RTP (SR-RTP) extension, that provides for application-level framing (ALF) as well as selective reliability. The Zeros field is a field that is required by the standard to allow multiple interoperating implementations to operate independently with different header extensions; set this to all zeros.

The ADU sequence number field uniquely identifies the ADU; in the case of MPEG-4 video, one frame corresponds to one ADU, so this is equivalent to a frame number. The ADU length field indicates the number of bytes contained within that particular ADU; this allows the transport layer to detect missing packets at the end of an ADU. The ADU offset uniquely identifies one packet within an ADU and allows for reassembly of a packet when reordering occurs. The header provides a Priority field that allows the transport layer to specify the relative importance of packets. In particular, for the purposes of our experiments, we mark MPEG-4 I-frames with a high priority so that a retransmission request is sent on an I-frame loss but not a P-frame or B-frame loss. The Layer field is used when transmitting layered video to specify the layer of video to which the packet corresponds; this feature can be used to calculate playout times, in decoding, or for caching purposes.

### 3.2 Receiver Postprocessing.

Because motion in video sequences tends to be highly correlated, however, we can make use of motion information that may be present in previous portions of the bitstream to reconstruct the lost information in the current I-frame. For scenes with higher motion, an algorithm is proposed for recovering I-frames that exploits both temporal correlation and the motion vector information in the bitstream.

A simple approach replaces the missing macro blocks with the same blocks from a previous frame. This works well in low-motion sequences, or when the loss occurs in a uniform background region. In the event of high motion, however, simple macro block replacement is not acceptable, because the missing macro block data will not correspond well with the same blocks in a previous frame. In this case, we must search for appropriate corresponding pixel values for the missing macro blocks. Fortunately, with high probability the bitstream will contain motion vectors for surrounding pictures, from which we can estimate the motion that has occurred in the region where packet loss has occurred. Figure 8 shows post processing of the high motion at the receiver side.

Figure 9 shows the data recover for texture data from previous P-frames.

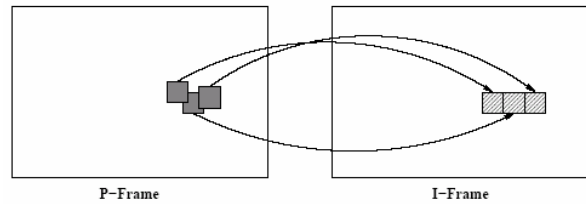


Figure 8: In High motion, replace the missing I-Frame with preceding P-Frame.

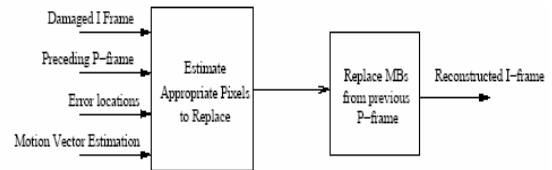


Figure 9: Block Diagram. Textual replacement.

## 4. Conclusion.

In order for video streaming to succeed on the Internet, systems must account for the anomalies of packet loss and changes in bandwidth and delay that make the delivery of real-time video on the Internet challenging. This paper analyzed the effects of packet loss on the quality of MPEG-4 video and proposed a model to explain these effects. This paper shown that, by recovery of only the most important data in the bitstream, significant performance gains can be achieved without much additional penalty in terms of latency.

## 5. References.

1. Nick Feamster and Hari Balakrishnan, Packet loss recovery for streaming video. MIT laboratory for Computer science, MA, USA.
2. Gui-liang Feng, Roberta Houston and Kuldeep Rawat, A novel packet loss recovery for video stream transmission over networks. Unviersity of Louisiana, LA.
3. Jill M. Boyce and Robeti D. Gaglianello. Packet loss effects on MPEG Video sent over Public Internet. ACM Multimedia '98. Bristol, UK.
4. Y. Pourmohammadi, K. Asrar Haghighi, A. Mohamed, H. M. Alnuweiri. Streaming MPEG-4 over IP and broadcast networks; DMIF based architectures.

5. Yi J. Liang<sup>†</sup>, John G. Apostolopoulos<sup>|</sup> and Bernd Girod<sup>†</sup>. Analysis of packet loss for compressed video; Does burst length matter?. Hewlett-packard labs, CA.
6. Fei Xue, Velibor Markovski, and Ljiljana Trajkovi, Packet loss in video transfers over IP networks. Simon fraser university, Canada.
7. Rob Koenen. Overview of the MPEG-4 standard. Technical report, ISO/IEC JTC1/SC29/WG11, December 1999.
8. W. Tan and A. Zakhor. Real-time Internet Video Using Error Resilient Scalable Compression and TCP-friendly Transport Protocol. IEEE Trans. On Multimedia, 1(2):172–186, May 1999.
9. H. Schulzrinne, A. Rao, and R. Lanphier. Real Time Streaming Protocol (RTSP). IETF, April 1998. RFC 2326.
10. N. Feamster, D. Bansal, and H. Balakrishnan. On the interactions between congestion control and layered quality adaptation for streaming video. In 11<sup>th</sup> International Packet Video Workshop, Kyongju, Korea, May 2001.