Real-time Video over the Internet: A Big Picture

Dapeng Wu* Y. Thomas Hou† Jason Yao† H. Jonathan Chao*

*Polytechnic University, Brooklyn, NY, USA.

†Fujitsu Laboratories of America, Sunnyvale, CA, USA.

Abstract

Real-time video over the Internet is becoming an important component of many multimedia applications. Due to both the requirements from video applications and the nature of the current Internet (e.g., delay, loss, and heterogeneity), there are many challenging issues for transporting real-time video over the Internet. This paper takes a holistic approach to address key challenges from both transport and compression perspectives. In particular, we outline a framework for real-time video over the Internet, which includes the following four components: congestion control, error control, error-resilient mechanisms and packetization. For each of the four components, we discuss existing proposed solutions. We point out that there is a need of synergy from both transport and compression perspectives in designing protocols and algorithms for real-time video over the Internet.

1 Introduction

Real-time video is becoming an important component of many multimedia applications. Video applications typically have the following requirements, which may not be well supported by the current Internet.

- *Bandwidth requirements:* To achieve acceptable presentation quality, real-time video typically has minimum bandwidth requirements. However, the current Internet does not provide such bandwidth guarantees.
- *Delay requirements:* In contrast to data transfer, which does not have strict delay constraints, real-time video is delay sensitive. But Internet congestion may bring excessive delay to video traffic.
- *Loss requirements:* Since packet loss makes the presentation displeasing to human eyes, video applications typically have loss requirements. But as packet loss is unavoidable in the current Internet, video presentation quality may be severely degraded.

For multicast video, there are additional challenging issues for real-time Internet video as follows.

- *Heterogeneity:* Heterogeneity exists in both the Internet and the receivers. Network heterogeneity refers to unevenly distributed resources (e.g., processing, bandwidth, storage and control policies) among the subnetworks in the Internet. Receiver heterogeneity refers to the difference in requirements (e.g., latency, visual quality) and processing capabilities among the receivers.
- *Scalability problem:* Certain control algorithms work well when they are executed by one receiver. But if it is applied to a large number of receivers, congestion may occur. This is referred to scalability problem. Scalability problem is critical and must be solved for multicast real-time video.

To support the above requirements from real-time video, there are two general approaches, namely the *network-centric approach,* and the *end system-based approach.* In the network-centric approach, the routers must be configured to support quality of service (QoS) such as bounded delay and packet loss for video applications. On the other hand, in the end system-based approach, end systems employ control techniques to maximize the video quality even without any QoS support from the routers. Such end system-based mechanisms are particularly significant since they are independent of the evolution of the underlying network technologies.

This paper focuses on the end system-based approach and takes a holistic approach to present solutions from both transport and compression perspectives. We outline a framework for real-time Internet video, which includes four key components: *congestion control, error control, error-resilient mechanisms,* and *packetization.*

Congestion control includes *rate control, rate adaptive video encoding,* and *rate shaping.* Rate control is from the transport perspective; rate adaptive video encoding is from the compression perspective; rate shaping takes both transport and compression into considerations.

Error control includes forward error correction (FEC) and retransmission, both of which are aimed at maximizing video presentation quality in the presence of packet loss. FEC can be achieved from the transport perspective, or the compression perspective, or both. Retransmission is from the transport perspective.

Error-resilient mechanisms attempt to maximize the video presentation quality in the presence of packet loss. It includes *error-resilient encoding* and *error concealment,* both of which are from compression perspective.

Packetization refers to converting a compressed video bit-stream into packets for transport over a packet-switched network (i.e., Internet). An appropriate packetization algorithm is essential for efficient and robust delivery of video over the Internet.

For the reminder of this paper, we elaborate on the above four key components of our framework for real-time video over the Internet. Section 2 presents the approaches on congestion control. In Section 3, we describe the mechanisms for error control. Section 4 discusses error-resilient mechanisms. In Section 5, we present packetization schemes. Section 6 summarizes this paper.

2 Congestion Control

Congestion control includes *rate control, rate adaptive video encoding,* and *rate shaping.*

2.1 Rate Control

Rate control takes the transport perspective. Since a window-based congestion control such as TCP introduces intolerable delays during packet retransmission, a rate-based congestion control (or rate control) is typically employed for transporting real-time video with UDP [21]. Existing rate control schemes for real-time video can be classified into three categories: (1) *source-based rate control,* (2) *receiver-based rate control,* and (3) *hybrid rate control.*

2.1.1 Source-based Rate Control

Under source-based rate control, the sender is responsible for adapting the transmission rate of the video stream. Source-based rate control attempts to minimize the amount of packet loss by matching the rate of the video stream to the available network bandwidth. Typically, feedback is employed by source-based rate control mechanisms in order for the source to keep track of the dynamic nature of the Internet. Source-based rate control can be applied to both unicast [21] and multicast scenarios [2].

For unicast video applications, existing source-based rate control mechanisms can be classified into two approaches, namely, *the probe-based approach* and *the model-based approach.*

The probe-based approach

The probe-based approach is based on probing experiments. Specifically, the source probes for available network bandwidth by adjusting the sending rate so that some QoS parameter can be satisfied (e.g., the packet loss ratio p below a certain threshold P_{th} [21]). The sending rate at the source can be adjusted through additive increase/multiplicative decrease [21] or multiplicative increase/multiplicative decrease [15].

The model-based approach

Since the probe-based approach implicitly estimates the available network bandwidth based on packet loss ratio, it may unfairly compete network bandwidth with TCP flows. To address this issue, a model-based approach has been proposed to calculate a share of network bandwidth explicitly so that a source can share network bandwidth with TCP flows in a fair (or *"friendly"*) manner [6].

The model-based approach is based on a throughput model of a TCP connection. Specifically, the throughput of a TCP connection, say λ, can be characterized as follows [6]:

$$\lambda = \frac{1.22 \times MTU}{RTT \times \sqrt{p}}, \quad (1)$$

where MTU (or Maximum Transit Unit) is the maximum packet size, RTT is the round trip time, and p is the packet loss ratio experienced by the flow. The MTU can be found through the mechanism proposed by Mogul and Deering [12]. The parameter RTT can be obtained through feedback of timing information. Finally, the receiver can periodically send the parameter p to the source in the time scale of round trip time. Upon the receipt of the parameter p, the source estimates the sending rate λ and adjusts its sending rate.

When Eq. (1) is used to determine the sending rate of the video stream, a video stream can share the network bandwidth with other TCP flows in a fair manner. For this reason, the model-based rate control is also called "TCP-friendly" rate control [6].

Multicasting

For multicast under the source-based rate control, the sender uses a single channel to transport the video stream to a group of receivers. Such a multicast is called *single-channel multicast.* For single-channel multicast, only the probe-based rate control is employed and is best illustrated by IVS (INRIA Video-conference System) [2].

In IVS [2], each receiver estimates its packet loss ratio and determines the network status to be in one of the following three states: UNLOADED, LOADED, or CONGESTED. The source uses randomly polling the receivers to solicit the network status information so as to avoid feedback implosion. Based on the percentage of UNLOADED and CONGESTED receivers, the source adjusts its sending rate.

Single-channel multicast has good bandwidth efficiency since all the receivers share one channel. But single-channel multicast is unable to provide flexibility and service differentiations to different receivers with diverse access link capacities, processing capabilities and interests.

On the other hand, the multicast video, delivered through individual unicast streams, can offer differentiated services to receivers since each receiver can negotiate the parameters of the service individually with the source. The problem with the unicast-based multicast video is bandwidth inefficiency.

To achieve good trade-off between bandwidth efficiency and service flexibility for multicasting video, two mechanisms, namely, *receiver-based rate control* and *hybrid rate control,* are proposed, which we discuss as follows.

2.1.2 Receiver-based Rate Control

Under receiver-based rate control, the receivers control the receiving rate of video streams by adding or dropping channels; the sender does not participate in rate control. Receiver-based rate control is typically applied to the layered multicast since the heterogeneity problem under multicast can be readily solved by receiver-based rate control.

For the layered multicast, at the sender side, a raw

video sequence is compressed into a base layer and one or more enhancement layers. The base layer can be independently decoded and it provides basic video quality. The enhancement layers can only be decoded together with the base layer and are used to further refine the quality of presentation. After compression, each video layer is sent to a separate IP multicast group. At the receiver side, each receiver subscribes to a certain set of video layers by joining the corresponding IP multicast groups. Each receiver tries to achieve the highest subscription level of video layers without incurring congestion.

Similar to source-based rate control, existing receiver-based rate control mechanisms can be put into two categories, namely, *the probe-based approach* and *the model-based approach.*

The probe-based approach was first employed in Receiver-driven Layered Multicast (RLM) [11]. Basically, the probe-based rate control works as follows. When no congestion is detected, a receiver probes for available bandwidth by joining a layer, resulting an increase of its receiving rate. If no congestion is detected after the joining, the join-experiment is successful. Otherwise, the receiver drops the newly added layer. When congestion is detected, a receiver drops a layer, causing an reduction of its receiving rate.

The RLM [11] has a potential scalability problem when the number of receivers becomes large. If each receiver carries out the above control independently, the aggregate frequency of join-experiments increases with the number of receivers. Since a failed join-experiment could incur congestion to the network, an increase of the number of join-experiments could aggravate network congestion. To minimize the frequency of join-experiments, a shared learning algorithm was proposed [11]. The essence of the shared learning algorithm is to let a receiver multicast its intent to the group before it starts a join-experiment. Each receiver can learn from other receivers' failed join-experiments, resulting in decrease of the frequency of failed join-experiments.

However, the shared learning algorithm in [11] requires each receiver to maintain a comprehensive group knowledge base containing the results of all the join-experiments for the multicast group. Furthermore, the use of multicasting to update the comprehensive group knowledge base may decrease usable bandwidth on low-speed links and lead to lower quality for receivers on these links. To reduce message processing overhead at each receiver and to decrease bandwidth usage of the shared learning algorithm, a hierarchical rate control mechanism was proposed in Layered Video Multicast with Retransmissions (LVMR) [10].

Unlike the probe-based approach which implicitly estimates the available network bandwidth through probing experiments, the model-based approach uses explicit estimation for available network bandwidth [17]. The model-based approach is based on the throughput model of a TCP connection, which is described in Section 2.1.1. Thus, the model-based rate control is also "TCP-friendly". Figure 1 shows the flow chart of the basic model-based rate control executed by each receiver, where γ_i is the transmission rate of layer i. In the algorithm, it is assumed that each receiver knows the transmission rate of all the layers.

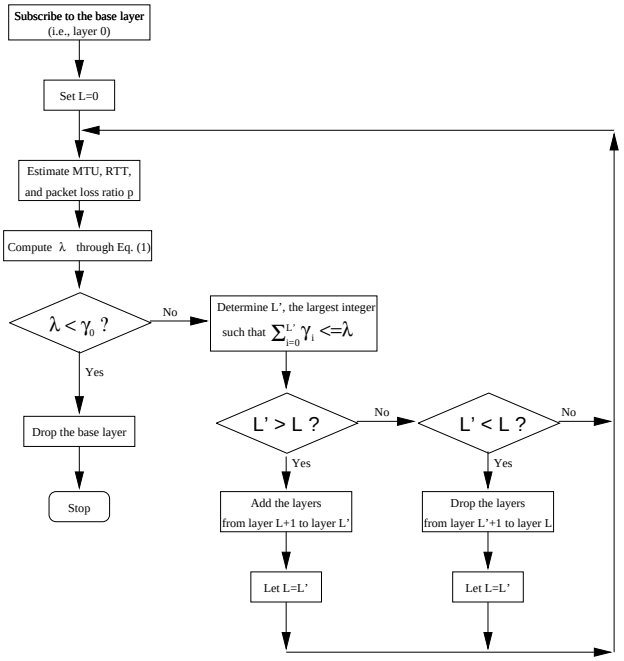

Figure 1: Flow chart of model-based rate control at a receiver.

2.1.3 Hybrid Rate Control

Under hybrid rate control, both the receivers and the sender can participate in rate control. That is, the receivers can regulate the receiving rate of video streams by adding or dropping channels while the sender can also adjust the transmission rate of each channel based on feedback information from the receivers.

2.2 Rate-adaptive Video Encoding

Rate adaptive video encoding follows the compression perspective. The objective of a rate-adaptive encoding algorithm is to maximize the perceptual quality under a given encoding rate. This is achieved by altering the encoder's quantization parameter (QP) and/or the video frame rate.

Traditional video encoders such as H.261 and MPEG-1/2 typically rely on the alteration of the QP of the encoder to achieve rate adaptation. These encoding schemes must perform coding with constant frame rate. This is because even a slight reduction in frame rate can substantially degrade the perceptual quality at the receiver, especially during a dynamic scene change. Since altering the QP does not suffice to achieve very low bit-rate, these encoding schemes may not be suitable for very low bit-rate video applications.

On the other hand, video encoders such as MPEG-4 and H.263 are suitable for very low bit-rate video applications since they allow the alteration of the frame rate. This is achieved by frame-skip, meaning that the frame is not encoded. Specifically, when the encoder buffer is going to overflow (i.e., the bit budget is over-used by the previous frame), a complete frame will be skipped at the encoder. This will allow the bits for the previous frames to be transmitted within the period of this frame, and thus reducing the buffer level.

For an object-based MPEG-4 video encoder, each individual video object is classified into a video object plane (VOP) and each VOP is encoded separately. Such isolation of video objects offers great flexibility to perform adaptive encoding. For example, we can dynamically adjust target bit-rate distribution among video objects, in addition to the alteration of QP on each VOP [21]. This can improve the perceptual quality for the regions of interest (e.g., head and shoulder) while lowering the quality for other regions (e.g., background).

2.3 Rate Shaping

Rate shaping could follow either the transport perspective or the compression perspective. A rate shaper is an interface (or filter) between the encoder and the network. The objective of rate shaping is to adapt the rate of a compressed video bit-stream to the target rate constraint. Since rate shaping does not require interactions with the encoder, rate shaping is applicable to any video coding scheme and is applicable to both live and stored video.

A representative rate shaping mechanism from the transport perspective is *server selective frame discard* [23]. As the loss of frames/packets is unavoidable in the Internet, the *selective frame discard* preemptively drops frames at the server in an intelligent manner by taking into consideration of the available network bandwidth and client QoS requirements.

A representative rate shaping mechanism from the compression perspective is *dynamic rate shaping* [5], which is based on rate-distortion (R-D) theory. More specifically, the dynamic rate shaper selectively discards the Discrete Cosine Transform (DCT) coefficients of the high frequencies so that the target rate can be achieved. Since human eyes are not sensitive to high frequencies, the dynamic rate shaper selects the highest frequencies and discards the DCT coefficients of these frequencies until the target rate can be met.

The fact that packet loss is unavoidable in the Internet and may have significant impact on perceptual quality prompts the need to design mechanisms to maximize the video presentation quality in presence of packet loss. In the following two sections, we discuss error control and error-resilient mechanisms, both of which are two effective means to enhance the video quality under error-prone environment.

3 Error Control

We organize this section as follows. In Section 3.1, we survey the approaches for FEC. Section 3.2 describes various mechanisms based on delay-constrained retransmission.

3.1 FEC

The principle of FEC is to add redundant information so that original messages can be reconstructed in the presence of packet loss. Depending on the particular redundant information being added, we classify existing FEC schemes into three categories: (1) *channel coding,* (2) *source coding-based FEC,* and (3) *joint source/channel coding.*

3.1.1 Channel Coding

For Internet applications, channel coding is typically done with block codes. A video stream is first chopped into segments, each of which is packetized into k packets; then for each segment, a block code (e.g., Tornado code [1]) is applied to the k packets to generate a n-packet block, where $n > k$. To perfectly recover a segment, a user only needs to receive any k packets in the n-packet block.

Since recovery is carried out entirely at the receiver, the channel coding approach can scale to arbitrary number of receivers in a large multicast group (i.e., no scalability issue). Furthermore, due to its ability to recover from any k out of n packets regardless of which particular packet is lost, it allows the network and receivers to discard some of the packets which cannot be handled due to limited bandwidth or processing power. Thus, it is also applicable to heterogeneous networks and receivers with different capabilities.

However, there are also some disadvantages associated with channel coding. First of all, channel coding may increase transmission rate. This is because channel coding adds $n - k$ redundant packets to every k original packets, which increases the rate by a factor of n/k. In addition, the higher the loss rate, the higher the transmission rate would be required to recover from the loss (i.e., the more redundant packets are required). The higher the transmission rate, the more congested the network would become, resulting in even higher packet loss rate. This makes channel coding vulnerable for short-term congestion. Second, channel coding may increase delay. This is because (1) a channel encoder at the sender must wait for all k packets in a segment before it can generate the n-packet block; and (2) the receiver must wait for at least k packets of a n-packet block to arrive before it can playback the video segment. In addition, recovery from bursty loss requires the use of either longer blocks (i.e., both k and n are increased) or techniques like interleaving. In either case, delay will be increased.

3.1.2 Source Coding-based FEC

Source coding-based FEC (SFEC) [3], a variant of FEC, was proposed recently for Internet video. Like channel coding, SFEC also adds redundant information to recover from loss. For example, an SFEC scheme can have the n-th packet contains the n-th GOB plus redundant information about the $(n-1)$-th GOB. If the $(n-1)$-th packet is lost and the n-th packet is received, the receiver can reconstruct the $(n-1)$-th GOB from the redundant information about the $(n-1)$-th GOB contained in the n-th packet. However, the reconstructed $(n-1)$-th GOB has coarser quality because the redundant information about the $(n-1)$-th GOB contained in the n-th packet is a compressed version of the $(n-1)$-th GOB with a larger quantization parameter.

The main difference between SFEC and channel coding is the kind of redundant information being added to a compressed video stream. Specifically, channel coding adds redundant information according to a block code (irrelevant to the video) while the redundant information added by SFEC is more compressed versions of the raw video. As a result, when there is packet loss, chan-

nel coding could achieve perfect recovery while SFEC recovers the video with reduced quality.

One advantage of SFEC over channel coding is smaller delay. This is because each packet can be decoded independently in SFEC, while, under channel coding, both the channel encoder and the channel decoder have to wait for at least k packets of a segment.

3.1.3 Joint Source/Channel Coding

Since channel coding is data-independent, which may not achieve optimal performance for a specific data type, to achieve better performance, joint source/channel coding may be employed. An example joint source/channel coding scheme, introduced by Davis and Danskin in [4], can be employed to transport images over the Internet. In this scheme, source and channel coding bits are allocated in a way that can minimize an expected distortion measure. As a result, more perceptually important low frequency sub-bands of images are shielded heavily using channel codes while higher frequencies are shielded lightly. This unequal error protection reduces channel coding overhead, which is most pronounced on bursty channels where uniform application of channel codes can be quite expensive.

3.2 Delay-constrained Retransmission

A conventional retransmission scheme such as automatic repeat request (ARQ) relies on the receiver to send feedback to the source when packets are lost. Upon receiving such feedback, the source retransmits the lost packets. The conventional ARQ is usually dismissed as a method for transporting real-time video since a retransmitted packet arrives at least 3 one-way trip times after the original packet, which might exceed the maximum allowable delay. However, if the one-way trip time is short with respect to the maximum allowable delay, a retransmission-based approach (also called delay-constrained retransmission) may be a viable option for error control [14].

In the following, we present various delay-constrained retransmission schemes for unicast and multicast.

3.2.1 Unicast

Delay-constrained retransmission mechanisms can be receiver-based, sender-based, or hybrid sender/receiver-based control.

The receiver-based control

The objective of the receiver-based control is to minimize retransmission requests that cannot meet delay constraint. The following shows an example of the receiver-based control.

> When the receiver detects the loss of packet N:
> if $(T_c + RTT + D_s < T_d(N))$
> send the request for retransmission of packet N to the sender

where T_c is the current time, RTT is an estimated round trip time, D_s is a slack term, and $T_d(N)$ is the time when packet N is scheduled for display. The slack term D_s is used to tolerate errors in estimating RTT, the sender's response time to a request, and/or the receiver's processing delay (e.g., decoding).

The sender-based control

Similar to the receiver-based control, the objective of the sender-based control is to minimize retransmission of packets that cannot meet their delay constraints. The following is an example of the sender-based control:

> When the sender receives a request for retransmission of packet N:
> if $(T_c + T_o + D_s < T'_d(N))$
> retransmit packet N to the receiver;

where T_o is the estimated one-way trip time (from the sender to the receiver), and $T'_d(N)$ is an estimate of $T_d(N)$. To obtain $T'_d(N)$, the receiver has to send $T_d(N)$ to the sender. Then, based on the differences between the sender's system time and the receiver's system time, the sender can derive $T'_d(N)$. The slack term D_s is used to tolerate errors in estimating T_o, tolerance of error in estimating $T'_d(N)$, and/or the receiver's processing delay (e.g., decoding).

The hybrid control

The hybrid control is a combination of both the sender-based control and the receiver-based control. The hybrid control may achieve better performance at the cost of higher complexity.

3.2.2 Multicast

In the multicast case, retransmissions must be restricted within closely located multicast members. This is because one-way trip times between these members tend to be small, making retransmissions effective for timely recovery. There is another problem associated with multicast, i.e., feedback implosion of retransmission requests, which must be addressed. Therefore, methods for delay-constrained retransmission for multicast typically attempt to limit the number or scope of retransmission requests.

A logical tree can be configured to limit the number/scope of retransmission requests and to achieve local recovery among closely located multicast members [9, 22]. The logical tree can be constructed by statically assigning Designated Receivers (DRs) at each level of the tree to help with retransmission of lost packets [9], or it can be dynamically constructed through the protocol used in STructure-Oriented Resilient Multicast (STORM) [22]. By adapting the tree structure to changing network traffic conditions and group membership, the system could achieve higher probability of receiving timely retransmissions.

Similar to the receiver-based control for unicast, receivers in a multicast group can decide whether or not to send retransmission requests. By minimizing the number of requests for retransmissions of those packets that cannot meet their time constraints, bandwidth efficiency can be improved [9].

To address heterogeneity problem, a receiver-initiated mechanism for error recovery can be adopted as in STORM [22]. Through such a mechanism, each receiver

can dynamically select the best possible DR to achieve good trade-off between desired latency and the degree of reliability.

4 Error-resilient Mechanisms

Error-resilient mechanisms address loss recovery purely from the compression perspective. Existing error-resilient mechanisms include *error-resilient encoding,* and *error concealment.* Error-resilient encoding is executed at the source to prevent error propagation should loss occur while error concealment is executed at the receiver when loss occurs.

4.1 Error-resilient Encoding

There are several standard error-resilient tools, including re-synchronization marking, data partitioning, and data recovery (e.g., reversible variable length codes (RVLC)) [7]. However, re-synchronization marking, data partitioning, and data recovery are targeted at error-prone environment like wireless channel and may not be applicable to Internet environment. For video transmission over the Internet, the boundary of a packet already provides a synchronization point in the variable-length coded bit-stream at the receiver side. Furthermore, since a packet loss may cause the loss of all the motion data and its associated shape/texture data, mechanisms such as re-synchronization marking, data partitioning, and data recovery may not be useful for Internet video applications. Therefore, we do not intend to discuss these standard error-resilient tools. Instead, we will focus on two techniques that are promising for robust Internet video transmission, namely, *optimal mode selection* and *multiple description coding.*

4.1.1 Optimal Mode Selection

High-compression coding algorithms usually employ inter-coding (i.e., prediction) to achieve efficiency. With these coding algorithms, loss of a packet may degrade video quality over a large number of frames, until the next intra-coded frame is received. Intra-coding can effectively stop error propagation at the expense of efficiency while inter-coding can achieve compression efficiency at the risk of error propagation. Therefore, a good mode selection between intra mode and inter mode should be in place to enhance the robustness of the compressed video.

A coding algorithm such as H.263 or MPEG-4 [7] usually employs rate control to match the output rate to the available bandwidth. The objective of rate-controlled compression algorithms is to maximize the video quality under the constraint of a given bit budget. This can be achieved by choosing a mode that minimizes the quantization distortion between the original frame or macroblock (MB) and the reconstructed one under a given bit budget [13], which is the so-called R-D optimized mode selection. We refer such R-D optimized mode selection as the classical approach. The classical approach is not able to achieve global optimality under the error-prone environment since it does not consider the network congestion status and the receiver behavior. To address this problem, an end-to-end approach has been proposed to optimize R-D mode selection [20], which takes into consideration of the source behavior, the path characteristics, and the receiver behavior. It has been shown that such an end-to-end approach is capable of offering superior performance over the classical approach for Internet video [20].

4.1.2 Multiple Description Coding

Multiple description coding (MDC) is another way of making trade-off between compression efficiency and robustness under packet loss [18]. With MDC, a raw video sequence is compressed into multiple streams (referred as descriptions). Each description provides acceptable visual quality while multiple descriptions combined can provide a better visual quality. The advantages of MDC are: (1) robustness: even if a receiver gets only one description, it can still reconstruct video with acceptable quality; (2) enhanced quality: if a receiver gets multiple descriptions, it can combine them together to produce a better presentation.

However, the above advantages come at a price. To make each description provide acceptable visual quality, each description must carry sufficient information about the original video. This will reduce the compression efficiency compared to conventional single description coding (SDC).

4.2 Error Concealment

Since human eyes can tolerate a certain degree of distortion in video signals, when a packet loss is detected, the receiver can employ error concealment to conceal the lost data and make the presentation as less displeasing to human eyes as possible [19].

There are two basic approaches for error concealment, namely, *spatial interpolation* and *temporal interpolation.* In spatial interpolation, missing pixel values are reconstructed using neighboring spatial information, while in temporal interpolation, lost data is reconstructed from data in the previous frames. Typically, spatial interpolation is used to reconstruct the missing data in intra-coded frames while temporal interpolation is used to reconstruct the missing data in inter-coded frames.

In recent years, numerous error-concealment schemes have been proposed in the literature (refer to [19] for a good survey). However, most error concealment techniques discussed in [19] are only applicable to either asynchronous transfer mode (ATM) or wireless environments, and require substantial computational complexity, which is applicable to decoding still images but may not be acceptable in decoding real-time video. In the following, we describe several simple error concealment schemes that are applicable to Internet video communication.

Scheme EC-1: The receiver replaces the whole frame (in which some blocks are corrupted due to packet losses) with the previous reconstructed frame.

Scheme EC-2: The receiver replaces a corrupted block with the block at the same location from the previous frame.

Scheme EC-3: The receiver replaces the corrupted block with the block from the previous frame pointed by a motion vector. The motion vector is

copied from its neighboring block when available, otherwise the motion vector is set to zero.

EC-1 and EC-2 are special cases of EC-3. If the motion vector of the corrupted block is available, EC-3 can achieve better performance than EC-1 and EC-2 while EC-1 and EC-2 have less complexity than that of EC-3.

5 Packetization

A packetization mechanism is an essential component for transporting compressed video over the Internet. The choice of a packetization algorithm may affect both the efficiency and robustness of video delivery [21]. It is clear that the use of large packet size will reduce packetization overhead, as long as packet size is upper bounded by the path MTU. But robustness also needs to be considered for packetization of video bit-streams. In the following, we summarize several popular packetization schemes for video transport.

PKT-1: Each generated packet has the same fixed packet size (e.g., [8]). Although this packetization scheme is very simple, an MB may be split into two packets, resulting in dependency between two packets.

PKT-2: Each generated packet solely contains a single MB (e.g., [16]). Under this scheme, no MB will be split and loss of a packet only corrupts one MB. For this reason, this packetization scheme is recommended by Internet Engineering Task Force (IETF) [16].

PKT-3: Each generated packet solely contains a single GOB (e.g., [24]). Under such scheme, no GOB will be split and loss of a packet only corrupts one GOB. This packetization scheme is also recommended by IETF [24].

PKT-4: This packetization is targeted at MPEG-4 and works as follows [21]. If a complete VOP in MPEG-4 fits into a packet, packetized such VOP with a single packet; otherwise, try to packetize as many MBs as possible into a packet without crossing over into the next VOP. This scheme takes into consideration of the VOP concept in MPEG-4. Loss of one packet only corrupts one VOP. Since VOP is larger than GOB and MB, PKT-4 achieves higher efficiency than PKT-2 and PKT-3. Also, PKT-4 removes dependency between packets, which is a problem for PKT-1.

6 Summary

Video communication is becoming an important component of Internet multimedia applications. There are many challenging issues in real-time video delivery over the Internet. This paper offers a big picture or framework to address these issues from both transport and compression perspectives at an end system. Our framework consists of four key components for real-time video over the Internet, namely, congestion control, error control, error-resilient mechanisms and packetization. For each component, we discussed existing approaches and schemes.

Table 1 summarizes this paper and shows a design space along two perspectives: transport and compression. For the four main rows in Table 1, we stress that each of the four component discussed in this paper is critical and any design that overlooks any one of the components would degrade overall performance. For the horizontal two main columns in Table 1, we find that a conventional mechanism from one perspective may be substituted or complemented by a new mechanism from another perspective. For example, channel coding (transport) can be substituted by source coding-based FEC (compression). There is much room to be explored between the transport and compression perspectives so as to meet the particular design objective and performance criterion. Recently, there have been extensive efforts on the combined transport and compression approaches [4, 20]. We expect that the synergy of transport and compression could provide better solutions to the problems encountered in the design of video delivery systems.

References

[1] A. Albanese, J. Blömer, J. Edmonds, M. Luby and M. Sudan, "Priority encoding transmission," *IEEE Trans. on Information Theory,* vol. 42, no. 6, Nov. 1996.

[2] J-C. Bolot, T. Turletti and I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet," *Proc. ACM SIGCOMM'94,* pp. 58–67, London, UK, Sept. 1994.

[3] J-C. Bolot and T. Turletti, "Adaptive error control for packet video in the Internet," *Proc. IEEE Int. Conf. on Image Processing (ICIP'96),* Lausanne, Sept. 1996.

[4] G. Davis and J. Danskin, "Joint source and channel coding for Internet image transmission," *Proc. SPIE Conference on Wavelet Applications of Digital Image Processing XIX,* Denver, CO, Aug. 1996.

[5] A. Eleftheriadis and D. Anastassiou, "Meeting arbitrary QoS constraints using dynamic rate shaping of coded digital video," *Proc. 5th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95),* pp. 95–106, April 1995.

[6] S. Floyd, and K. Fall , "Promoting the use of end-to-end congestion control in the Internet," *IEEE/ACM Trans. on Networking,* vol. 7, no. 4, pp. 458–472, Aug. 1999.

[7] ISO/IEC JTC 1/SC 29/WG 11, "Information technology – coding of audio-visual objects, part 1: systems, part 2: visual, part 3: audio," *FCD 14496,* Dec. 1998.

[8] F. Le Leannec and C. M. Guillemot, "Error resilient video transmission over the Internet," *SPIE Proc. Visual Communications and Image Processing (VCIP'99),* Jan. 1999.

[9] X. Li, S. Paul, P. Pancha and M. H. Ammar, "Layered video multicast with retransmissions (LVMR): evaluation of error recovery schemes," *Proc. IEEE NOSSDAV'97,* May 1997.

Table 1: Design space for real-time video over the Internet.

		Transport perspective	Compression perspective
Congestion control	Rate control	*Source-based*	
		Receiver-based	
		Hybrid	
	Rate adaptive encoding		*Altering quantizer*
			Altering frame rate
	Rate shaping	*Selective frame discard*	*Dynamic rate shaping*
Error control	FEC	*Channel coding*	*SFEC*
		Joint channel/source coding	
	Delay-constrained retransmission	*Sender-based control*	
		Receiver-based control	
		Hybrid control	
Error resilient mechanisms	Error resilient encoding		*Optimal mode selection*
			Multiple description coding
	Error concealment		*EC*-1/2/3
Packetization		*PKT*-1	*PKT*-2/3
		PKT-4	

[10] X. Li, S. Paul and M. H. Ammar, "Layered video multicast with retransmissions (LVMR): evaluation of hierarchical rate control," *Proc. IEEE INFOCOM'98,* March 1998.

[11] S. McCanne, V. Jacobson and M. Vetterli, "Receiver-driven layered multicast," *Proc. ACM SIGCOMM'96,* pp. 117–130, Aug. 1996.

[12] J. Mogul and S. Deering, "Path MTU discovery," *RFC 1191,* Internet Engineering Task Force, Nov. 1990.

[13] A. Ortega and K. Ramchandran, "Rate-distortion methods for image and video compression," *IEEE Signal Processing Magazine,* pp. 74–90, Nov. 1998.

[14] M. Podolsky, M. Vetterli and S. McCanne, "Limited retransmission of real-time layered multimedia," *Proc. IEEE Workshop on Multimedia Signal Processing,* Dec. 1998.

[15] T. Turletti and C. Huitema, "Videoconferencing on the Internet," *IEEE/ACM Trans. on Networking,* vol. 4, no. 3, pp. 340–351, June 1996.

[16] T. Turletti and C. Huitema, "RTP payload format for H.261 video streams," *RFC 2032,* Internet Engineering Task Force, Oct. 1996.

[17] T. Turletti, S. Parisis and J. Bolot, "Experiments with a layered transmission scheme over the Internet," *INRIA Technical Report,* `http://www.inria.fr/RRRT/RR-3296.html`, Nov. 1997.

[18] Y. Wang, M. T. Orchard and A. R. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients," *Proc. IEEE Workshop on Multimedia Signal Processing,* June 1997.

[19] Y. Wang and Q.-F. Zhu, "Error control and concealment for video communication: a review," *Proceedings of the IEEE,* vol. 86, no. 5, pp. 974–997, May 1998.

[20] D. Wu, Y. T. Hou, B. Li, W. Zhu, Y.-Q. Zhang and H. J. Chao, "An end-to-end approach for optimal mode selection in Internet video communication: theory and application," to appear in *IEEE J. on Selected Areas in Communications,* 2000.

[21] D. Wu, Y. T. Hou, W. Zhu, H.-J. Lee, T. Chiang, Y.-Q. Zhang and H. J. Chao, "On end-to-end architecture for transporting MPEG-4 video over the Internet," to appear in *IEEE Trans. on Circuits and Systems for Video Technology,* 2000.

[22] X. R. Xu, A. C. Myers, H. Zhang and R. Yavatkar, "Resilient multicast support for continuous-media applications," *Proc. IEEE NOSSDAV'97.*

[23] Z.-L. Zhang, S. Nelakuditi, R. Aggarwa and R. P. Tsang, "Efficient server selective frame discard algorithms for stored video delivery over resource constrained networks," *Proc. IEEE INFOCOM'99,* March 1999.

[24] C. Zhu, "RTP payload format for H.263 video streams," *RFC 2190,* Internet Engineering Task Force, Sept. 1997.