Transporting Real-time Video over the Internet: Challenges and Approaches

Dapeng Wu, *Student Member, IEEE,* Yiwei Thomas Hou, *Member, IEEE,* and Ya-Qin Zhang *Fellow, IEEE*

Abstract— **Delivering real-time video over the Internet is an important component of many Internet multimedia applications. Transmission of real-time video has bandwidth, delay and loss requirements. However, the current Internet does not offer any quality of service (QoS) guarantees to video transmission over the Internet. In addition, the heterogeneity of the networks and end-systems makes it difficult to multicast Internet video in an efficient and flexible way. Thus, designing protocols and mechanisms for Internet video transmission poses many challenges. In this paper, we take a holistic approach to these challenges and present solutions from both transport and compression perspectives. With the holistic approach, we design a framework for transporting real-time Internet video, which includes two components, namely, congestion control and error control. Specifically, congestion control consists of rate control, rate adaptive encoding, and rate shaping; error control consists of forward error correction (FEC), retransmission, error-resilience and error concealment. For the design of each component in the framework, we classify approaches and summarize representative research work. We point out there exists a design space which can be explored by video application designers, and suggest that the synergy of both transport and compression could provide good solutions.**

Keywords— **Internet, real-time video, congestion control, error control.**

I. Introduction

UNICAST and multicast delivery of real-time video are important building blocks of many Internet multimedia applications, such as Internet television (see Fig. 1), video conferencing, distance learning, digital libraries, telepresence, and video-on-demand. Transmission of real-time video has bandwidth, delay and loss requirements. However, there is no quality of service (QoS) guarantee for video transmission over the current Internet. In addition, for video multicast, the heterogeneity of the networks and receivers makes it difficult to achieve bandwidth efficiency and service flexibility. Therefore, there are many challenging issues that need to be addressed in designing protocols and mechanisms for Internet video transmission.

We list the challenging QoS issues as follows.

1. **Bandwidth.** To achieve acceptable presentation quality, transmission of real-time video typically has minimum bandwidth requirement (say, 28 Kbps). However, the current Internet does not provide bandwidth reservation to meet such a requirement. Furthermore, since traditional routers typically do not actively participate in congestion control [7], excessive traffic can cause congestion collapse, which can further degrade the throughput of real-time video.
2. **Delay.** In contrast to data transmission, which are usually not subject to strict delay constraints, real-time video requires bounded end-to-end delay (say, 1 second). That is, every video packet must arrive at the destination in time to be decoded and displayed. This is because real-time video must be played out continuously. If the video packet does not arrive timely, the playout process will pause, which is annoying to human eyes. In other words, the video packet that arrives beyond a time constraint is useless and can be considered lost. Although real-time video requires timely delivery, the current Internet does not offer such a delay guarantee. In particular, the congestion in the Internet could incur excessive delay, which exceeds the delay requirement of real-time video.
3. **Loss.** Loss of packets can potentially make the presentation displeasing to human eyes, or, in some cases, make the presentation impossible. Thus, video applications typically impose some packet loss requirements. Specifically, the packet loss ratio is required to be kept below a threshold (say, 1%) to achieve acceptable visual quality. Although real-time video has a loss requirement, the current Internet does not provide any loss guarantee. In particular, the packet loss ratio could be very high during network congestion, causing severe degradation of video quality.

Besides the above QoS problems, for video multicast applications, there is another challenge coming from the *heterogeneity* problem. Before addressing the heterogeneity problem, we first describe the advantages and disadvantages of unicast and multicast. The unicast delivery of real-time video uses point-to-point transmission, where only one sender and one receiver are involved. In contrast, the multicast delivery of real-time video uses point-to-multipoint transmission, where one sender and multiple receivers are involved. For applications such as video conferencing and Internet television, delivery using multicast can achieve high bandwidth efficiency since the receivers can share links. On the other hand, unicast delivery of such applications is inefficient in terms of bandwidth utilization. An example is give in Fig. 2, where, for unicast, five copies of the video content flow across Link 1 and three copies flow across Link 2 as shown in Fig. 2(a). In contrast, the multicast removes this replication. That is, there is

Manuscript received February 2, 2000; revised August 3, 2000. This paper was recommended by Editor Jim Calder.

D. Wu is with Carnegie Mellon University, Dept. of Electrical & Computer Engineering, 5000 Forbes Ave., Pittsburgh, PA 15213, USA.

Y.T. Hou is with Fujitsu Laboratories of America, 595 Lawrence Expressway, Sunnyvale, CA 94085, USA.

Y.-Q. Zhang is with Microsoft Research China, 5F, Beijing Sigma Center, No. 49 Zhichun Road, Haidian District, Beijing 100080, China.

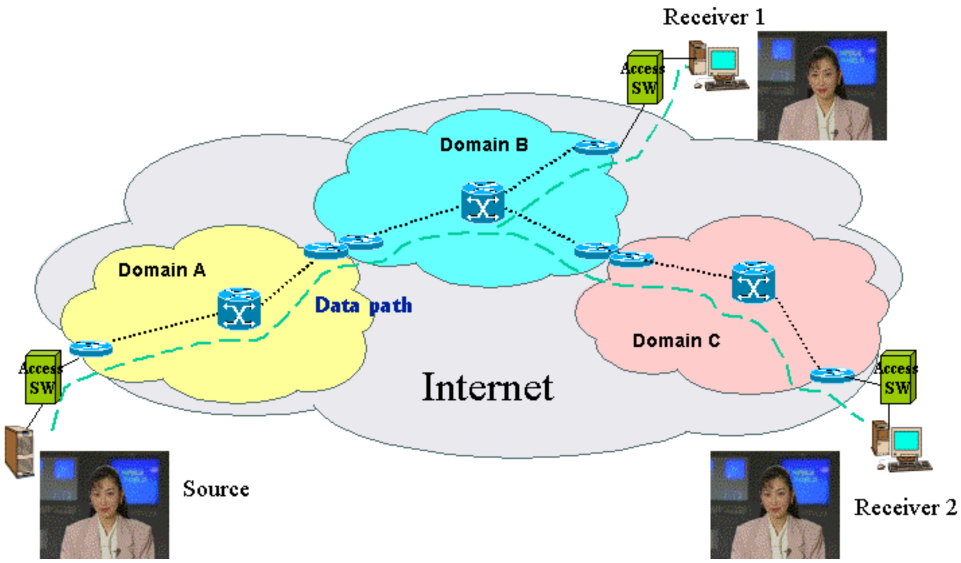

Fig. 1. Internet television uses multicast (point-to-multipoint communication) instead of unicast (point-to-point communication) to deliver real-time video so that users can share the common links to reduce bandwidth usage in the network.

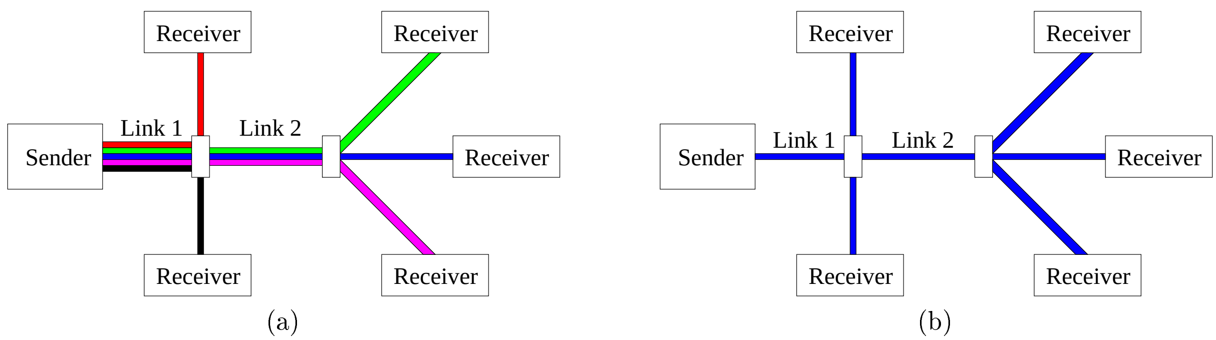

Fig. 2. (a) Unicast video distribution using multiple point-to-point connections. (b) Multicast video distribution using point-to-multipoint transmission.

only one copy of the video content traversing any link in the network (Fig. 2(b)), resulting in substantial bandwidth savings. However, the efficiency of multicast is achieved at the cost of losing the service flexibility of unicast (i.e., in unicast, each receiver can individually negotiate service parameters with the source). Such lack of flexibility in multicast can be problematic in a heterogeneous network environment, which we elaborate as follows.

Heterogeneity. There are two kinds of heterogeneity, namely, *network heterogeneity* and *receiver heterogeneity.* Network heterogeneity refers to the sub-networks in the Internet having unevenly distributed resources (e.g., processing, bandwidth, storage and congestion control policies). Network heterogeneity could make different user experience different packet loss/delay characteristics. Receiver heterogeneity means that receivers have different or even varying latency requirements, visual quality requirement, and/or processing capability. For example, in live multicast of a lecture, participants who want to ask questions and interact with the lecturer desire stringent real-time constraints on the video while passive listeners may be willing to sacrifice latency for higher video quality.

The sharing nature of multicast and the heterogeneity of networks and receivers sometimes present a conflicting dilemma. For example, the receivers in Fig. 2(b) may attempt to request for different video quality with different bandwidth. But only one copy of the video content is sent out from the source. As a result, all the receivers have to receive the same video content with the same quality. It is thus a challenge to design a multicast mechanism that not only achieves efficiency in network bandwidth but also meets the various requirements of the receivers.

To address the above technical issues, two general approaches have been proposed. The first approach is *network-centric.* That is, the routers/switches in the network are required to provide QoS support to guarantee bandwidth, bounded delay, delay jitter, and packet loss for video applications (e.g, integrated services [6], [11], [42], [65] or differentiated services [2], [27], [35]). The second approach is solely *end system-based* and does not impose any requirements on the network. In particular, the end systems employ control techniques to maximize the video quality without any QoS support from the transport network. In this paper, we focus on the end system-based approach. Such an approach is of particular significance since it does not require the participation of the networks and is applicable to both the current and future Internet.

Extensive research based on the end system-based approach has been conducted and various solutions have been proposed. This paper aims at giving the reader a big picture of this challenging area and identifying a design space that can be explored by video application designers. We take a holistic approach to present solutions from both transport and compression perspectives. By transport perspective, we refer to the use of control/processing techniques without regard of the specific video semantics. In other words, these control/processing techniques are applicable to generic data. By compression perspective, we mean employing signal processing techniques with consideration of the video semantics on the compression layer. With the holistic approach, we design a framework, which consists of two components, namely, *congestion control* and *error control.*

1. **Congestion control.** Bursty loss and excessive delay have devastating effect on video presentation quality and they are usually caused by network congestion. Thus, congestion control is required to reduce packet loss and delay. One congestion control mechanism is *rate control* [5]. Rate control attempts to minimize network congestion and the amount of packet loss by matching the rate of the video stream to the available network bandwidth. In contrast, without rate control, the traffic exceeding the available bandwidth would be discarded in the network. To force the source to send the video stream at the rate dictated by the rate control algorithm, rate adaptive video encoding [63] or rate shaping [18] is required. Note that rate control is from the transport perspective, while rate adaptive video encoding is from the compression perspective; rate shaping is in both transport and compression domain.
2. **Error control.** The purpose of congestion control is to prevent packet loss. However, packet loss is unavoidable in the Internet and may have significant impact on perceptual quality. Thus, other mechanisms must be in place to maximize video presentation quality in presence of packet loss. Such mechanisms include error control mechanisms, which can be classified into four types, namely, *forward error correction (FEC), retransmission, error-resilience,* and *error concealment.* The principle of FEC is to add extra (redundant) information to a compressed video bit-stream so that the original video can be reconstructed in presence of packet loss. There are three kinds of FEC: (1) channel coding, (2) source coding-based FEC, and (3) joint source/channel coding. The use of FEC is primarily because of its advantage of small transmission delay [14]. But FEC could be ineffective when bursty packet loss occurs and such loss exceeds the recovery capability of the FEC codes. Conventional retransmission-based schemes such as automatic repeat request (ARQ) are usually dismissed as a means for transporting real-time video since the delay requirement may not be met. However, if the one-way trip time is short with respect to the maximum allowable delay, a retransmission-based approach (called delay-constrained retransmission) is a viable option for error control [38]. Error-resilient schemes deal with packet loss on the compression layer. Unlike traditional FEC (i.e., channel coding), which directly corrects bit errors or packet losses, error-resilient schemes consider the semantic meaning of the compression layer and attempt to limit the scope of damage (caused by packet loss) on the compression layer. As a result, error-resilient schemes could reconstruct the video picture with gracefully degraded quality. Error concealment is a post-processing technique used by the decoder. When uncorrectable bit errors occur, the decoder uses error concealment to hide the glitch from the viewer so that a more visually pleasing rendition of the decoded video can be obtained. Note that channel coding and retransmission recover packet loss from the transport perspective, while source coding-based FEC, error-resilience, and error concealment deal with packet loss from the compression perspective; joint source/channel coding falls in both transport and compression domain.

The remainder of this paper is organized as follows. Section II presents the approaches for congestion control. In Section III, we describe the mechanisms for error control. Section IV summarizes this paper and points out future research directions.

II. Congestion Control

There are three mechanisms for congestion control: rate control, rate adaptive video encoding, and rate shaping. Rate control follows the transport approach; rate adaptive video encoding follows the compression approach; rate shaping could follow either the transport approach or the compression approach.

For the purpose of illustration, we present an architecture including the three congestion control mechanisms in Fig. 3, where the rate control is a source-based one (i.e., the source is responsible for adapting the rate). Although the architecture in Fig. 3 is targeted at transporting live video, this architecture is also applicable to stored video if the rate adaptive encoding is excluded. At the sender side, the compression layer compresses the live video based on a rate adaptive encoding algorithm. After this stage, the compressed video bit-stream is first filtered by the rate shaper and then passed through the RTP/UDP/IP layers before entering the Internet, where RTP is Real-time Transport Protocol [41]. Packets may be dropped inside the Internet (due to congestion) or at the destination (due to excess delay). For packets that are successfully delivered to the destination, they first pass through the IP/UDP/RTP layers before being decoded at the video decoder.

Under our architecture, a QoS monitor is maintained at the receiver side to infer network congestion status based on the behavior of the arriving packets, e.g., packet loss and delay. Such information is used in the feedback control protocol, which sends information back to the video source. Based on such feedback information, the rate control module estimates the available network bandwidth and conveys the estimated network bandwidth to the rate adaptive encoder or the rate shaper. Then, the rate adaptive encoder or the rate shaper regulates the output rate of the video stream according to the estimated network bandwidth. It

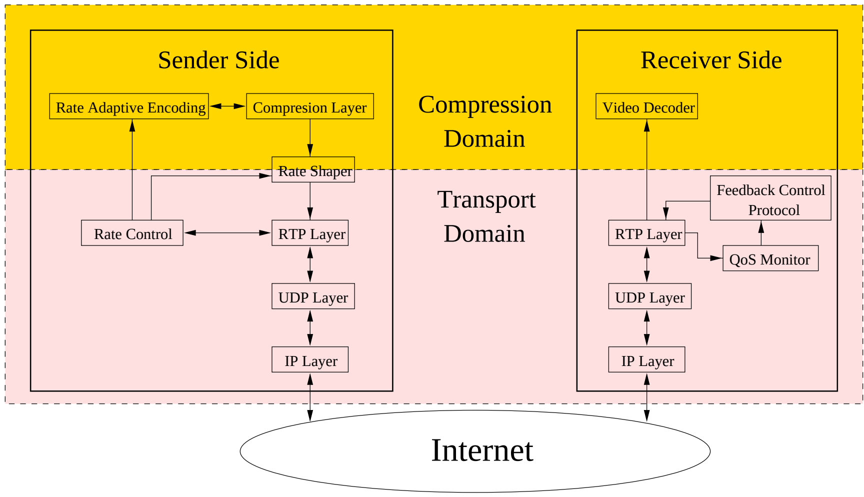

Fig. 3. A layered architecture for transporting real-time video.

is clear that the source-based congestion control must include: (1) rate control, and (2) rate adaptive video encoding or rate shaping.

We organize the rest of this section as follows. In Section II-A, we survey the approaches for rate control. Section II-B describes basic methods for rate-adaptive video encoding. In Section II-C, we classify methodologies for rate shaping and summarize representative schemes.

A. Rate Control: A Transport Approach

Since TCP retransmission introduces delays that may not be acceptable for real-time video applications, UDP is usually employed as the transport protocol for real-time video streams [63]. However, UDP is not able to provide congestion control and overcome the lack of service guarantees in the Internet. Therefore, it is necessary to implement a control mechanism on the upper layer (higher than UDP) to prevent congestion.

There are two types of control for congestion prevention: one is *window-based* [26] and the other is *rate-based* [52]. The window-based control such as TCP works as follows: it probes for the available network bandwidth by slowly increasing a congestion window (used to control how much data is outstanding in the network); when congestion is detected (indicated by the loss of one or more packets), the protocol reduces the congestion window greatly (see Fig. 4). The rapid reduction of the window size in response to congestion is essential to avoid network collapse. On the other hand, the rate-based control sets the sending rate based on the estimated available bandwidth in the network; if the estimation of the available network bandwidth is relatively accurate, the rate-based control could also prevent network collapse.

Since the window-based control like TCP typically couples retransmission which can introduce intolerable delays, the rate-based control (i.e., rate control) is usually employed for transporting real-time video [63]. Existing rate control schemes for real-time video can be classified into three categories, namely, source-based, receiver-based, and hybrid rate control, which are described in Sections II-A.1 to II-A.3, respectively.

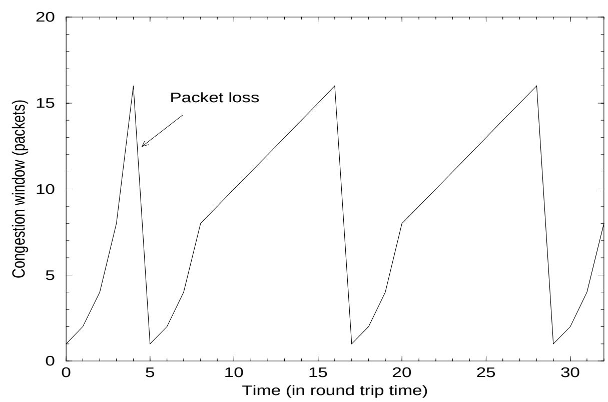

Fig. 4. Congestion window behavior under window-based control.

A.1 Source-based Rate Control

Under the source-based rate control, the sender is responsible for adapting the transmission rate of the video stream. The source-based rate control can minimize the amount of packet loss by matching the rate of the video stream to the available network bandwidth. In contrast, without rate control, the traffic exceeding the available bandwidth could be discarded in the network.

Typically, feedback is employed by source-based rate control mechanisms to convey the changing status of the Internet. Based upon the feedback information about the network, the sender could regulate the rate of the video stream. The source-based rate control can be applied to both unicast [63] and multicast [3].

For unicast video, the existing source-based rate control mechanisms can be classified into two approaches, namely, the probe-based approach and the model-based approach, which are presented as follows.

Probe-based approach. Such an approach is based on probing experiments. Specifically, the source probes for the available network bandwidth by adjusting the sending rate

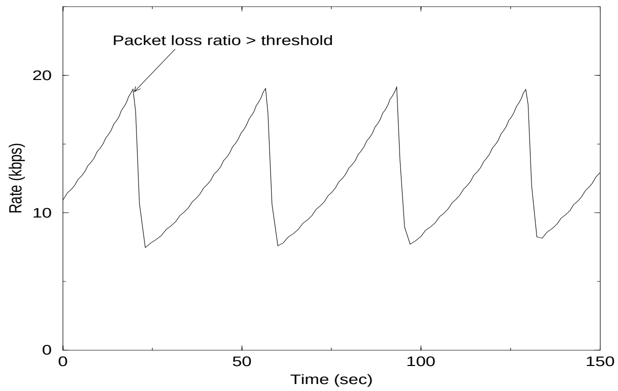

Fig. 5. Source rate behavior under the AIMD rate control.

so that some QoS requirements are met, e.g., packet loss ratio p is below a certain threshold P_{th} [63]. The value of P_{th} is determined according to the minimum video perceptual quality required by the receiver. There are two ways to adjust the sending rate: Additive Increase and Multiplicative Decrease (AIMD) [63], and Multiplicative Increase and Multiplicative Decrease (MIMD) [52]. The probe-based rate control could avoid congestion since it always tries to adapt to the congestion status, e.g., keep the packet loss at an acceptable level.

For the purpose of illustration, we briefly describe the source-based rate control based on additive increase and multiplicative decrease. The AIMD rate control algorithm is shown as follows [63].

if ($p \leq P_{th}$)
  $r := \min\{(r + \text{AIR}), \text{MaxR}\}$;
else
  $r := \max\{(\alpha \times r), \text{MinR}\}$.

where p is the packet loss ratio; P_{th} is the threshold for the packet loss ratio; r is the sending rate at the source; AIR is the additive increase rate; MaxR and MinR are the maximum rate and the minimum rate of the sender, respectively; and α is the multiplicative decrease factor.

Packet loss ratio p is measured by the receiver and conveyed back to the sender. An example source rate behavior under the AIMD rate control is illustrated in Fig. 5.

Model-based approach. Different from the probe-based approach, which implicitly estimates the available network bandwidth, the model-based approach attempts to estimate the available network bandwidth explicitly. This can be achieved by using a throughput model of a TCP connection, which is characterized by the following formula [19]:

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

where λ is the throughput of a TCP connection, MTU (Maximum Transit Unit) is the maximum packet size used by the connection, RTT is the round trip time for the connection, p is the packet loss ratio experienced by the connection. Under the model-based rate control, Eq. (1) can be used to determine the sending rate of the video stream. That is, the rate-controlled video flow gets its bandwidth share like a TCP connection. As a result, the rate-controlled video flow could avoid congestion in a way similar to that of TCP, and can co-exist with TCP flows in a "friendly" manner. Hence, the model-based rate control is also called "TCP friendly" rate control [57]. In contrast to this TCP friendliness, a flow without rate control can get much more bandwidth than a TCP flow when the network is congested. This may lead to possible starvation of competing TCP flows due to the rapid reduction of the TCP window size in response to congestion.

To compute the sending rate λ in Eq. (1), it is necessary for the source to obtain the MTU, RTT, and packet loss ratio p. The MTU can be found through the mechanism proposed by Mogul and Deering [34]. In the case when the MTU information is not available, the default MTU, i.e., 576 bytes, will be used. The parameter RTT can be obtained through feedback of timing information. In addition, the receiver can periodically send the parameter p to the source in the time scale of the round trip time. Upon the receipt of the parameter p, the source estimates the sending rate λ and then a rate control action may be taken.

Single-channel multicast vs. unicast. For multicast under the source-based rate control, the sender uses a single channel or one IP multicast group to transport the video stream to the receivers. Thus, such multicast is called *single-channel multicast*.

For single-channel multicast, only the probe-based rate control can be employed [3]. A representative work is the IVS (INRIA Video-conference System) [3]. The rate control in IVS is based on additive increase and multiplicative decrease, which is summarized as follows. Each receiver estimates its packet loss ratio, based on which, each receiver can determine the network status to be in one of the three states: UNLOADED, LOADED, and CONGESTED. The source solicits the network status information from the receivers through probabilistic polling, which helps to avoid feedback implosion.[1] In this way, the fraction of UNLOADED and CONGESTED receivers can be estimated. Then, the source adjusts the sending rate according to the following algorithm.

if ($F_{con} > T_{con}$)
  $r := \max(r/2, \text{MinR})$;
else if ($F_{un} == 100\%$)
  $r := \min\{(r + \text{AIR}), \text{MaxR}\}$;

where F_{con}, F_{un}, and T_{con} are fraction of CONGESTED receivers, fraction of UNLOADED receivers, and a preset threshold, respectively; r, MaxR, MinR, and AIR are the sending rate, the maximum rate, the minimum rate, and additive increase rate, respectively.

Single-channel multicast has good bandwidth efficiency since all the receivers share one channel (e.g., the IP multicast group in Fig. 2(b)). But single-channel multicast is unable to provide service flexibility and differentiation to

[1] Feedback implosion means that there are too many feedback messages for the source to handle.

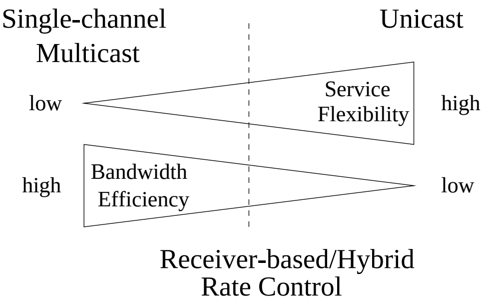

Fig. 6. Trade-off between bandwidth efficiency and service flexibility.

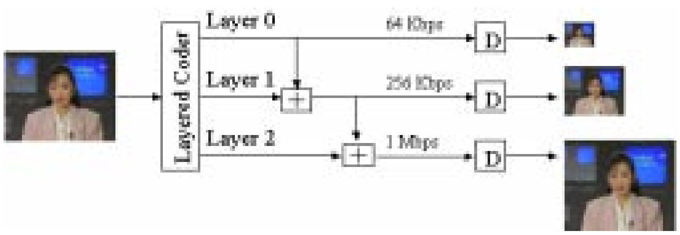

Fig. 7. Layered video encoding/decoding. D denotes the decoder.

different receivers with diverse access link capacities, processing capabilities and interests.

On the other hand, multicast video, delivered through individual unicast streams (see Fig. 2(a)), can offer differentiated services to receivers since each receiver can individually negotiate the service parameters with the source. But the problem with unicast-based multicast video is bandwidth inefficiency.

Single-channel multicast and unicast-based multicast are two extreme cases shown in Fig. 6. To achieve good trade-off between bandwidth efficiency and service flexibility for multicast video, two mechanisms, namely, receiver-based and hybrid rate control, have been proposed, which we discuss as follows.

A.2 Receiver-based Rate Control

Under the receiver-based rate control, the receivers regulate the receiving rate of video streams by adding/dropping channels. In contrast to the sender-based rate control, the sender does not participate in rate control here. Typically, the receiver-based rate control is applied to layered multicast video rather than unicast video. This is primarily because the source-based rate control works reasonably well for unicast video and the receiver-based rate control is targeted at solving heterogeneity problem in the multicast case.

Before we address the receiver-based rate control, we first briefly describe layered multicast video as follows. At the sender side, a raw video sequence is compressed into multiple layers: a base layer (i.e., Layer 0) and one or more enhancement layers (e.g., Layers 1 and 2 in Fig. 7). 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 they further refine the quality of the base layer. This is illustrated in Fig. 7. The base layer consumes the least bandwidth (e.g., 64 Kbps in Fig. 7); the higher the layer is, the more bandwidth the layer consumes (see Fig. 7). 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 group. In addition, each receiver tries to achieve the highest subscription level of video layers without incurring congestion. In the example shown in Fig. 8, each layer has a separate IP multicast group. Receiver 1 joins all three IP multicast groups. As a result, it consumes 1 Mbps and receives all the three layers. Receiver 2 joins the two IP multicast groups for Layer 0 and Layer 1 with bandwidth usage of 256 Kbps. Receiver 3 only joins the IP multicast group for Layer 0 with bandwidth consumption of 64 Kbps.

Like the source-based rate control, we classify existing receiver-based rate control mechanisms into two approaches, namely, the probe-based approach and the model-based approach, which are presented as follows.

Probe-based approach. This approach was first employed in Receiver driven Layered Multicast (RLM) [33]. Basically, the probe-based rate control consists of two parts:

1. When no congestion is detected, a receiver probes for the available bandwidth by joining a layer, which leads to an increase of its receiving rate. If no congestion is detected after the joining, the join-experiment is considered "successful". Otherwise, the receiver drops the newly added layer.
2. When congestion is detected, the receiver drops a layer, resulting in reduction of its receiving rate.

The above control has a potential problem when the number of receivers becomes large. If each receiver carries out the above join-experiment independently, the aggregate frequency of such experiments increases with the number of receivers. Since a failed join-experiment could incur congestion to the network, an increase of join-experiments could aggravate network congestion.

To minimize the frequency of join-experiments, a shared learning algorithm was proposed in [33]. The essence of the shared learning algorithm is to have a receiver multicast its intent to the group before it starts a join-experiment. In this way, each receiver can learn from other receivers' failed join-experiments, resulting in a decrease of the number of failed join-experiments.

The shared learning algorithm in [33] requires each receiver to maintain a comprehensive group knowledge base, which contains the results of all the join-experiments for the multicast group. In addition, 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 called Layered Video Multicast with Retransmissions (LVMR) [30] was proposed. The methodology of the hierarchical rate control is to partition the comprehensive group knowledge base, organize the partitions in a hierarchical way and distribute relevant

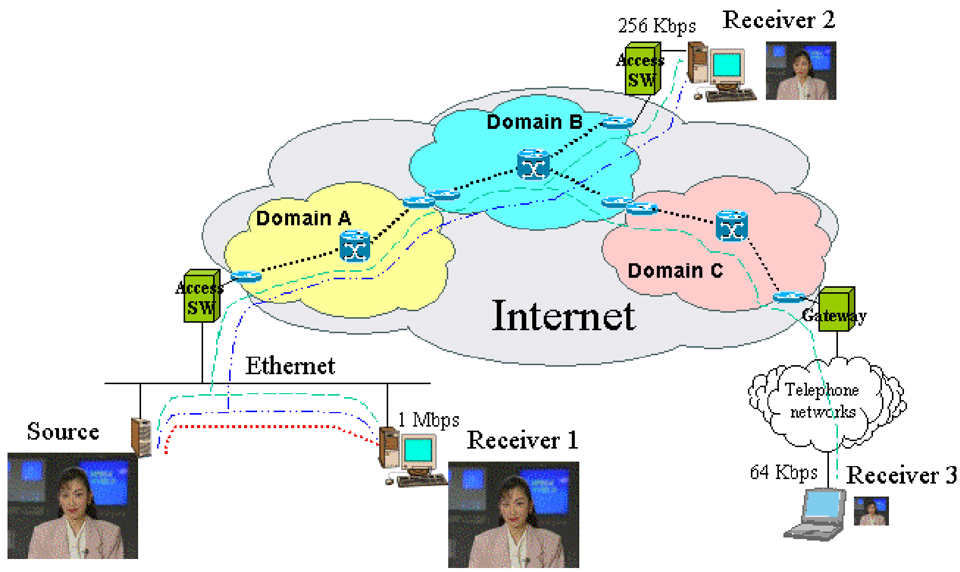

Fig. 8. IP multicast for layered video.

information (rather than all the information) to the receivers. In addition, the partitioning of the comprehensive group knowledge base allows multiple experiments to be conducted simultaneously, making it faster for the rate to converge to the stable state. Although the hierarchical rate control could reduce control protocol traffic, it requires installing agents in the network so that the comprehensive group knowledge base can be partitioned and organized in a hierarchical way.

Model-based approach. Unlike the probe-based approach which implicitly estimates the available network bandwidth through probing experiments, the model-based approach attempts to explicitly estimate the available network bandwidth. The model-based approach is based on the throughput model of a TCP connection, which was described in Section II-A.1.

Fig. 9 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. For the ease of description, we divide the algorithm into the following steps.

Initialization: A receiver starts with subscribing the base layer (i.e., Layer 0) and initializes the variable L to 0. The variable L represents the highest layer currently subscribed.

Step 1: Receiver estimates MTU, RTT, and packet loss ratio p for a given period. The MTU can be found through the mechanism proposed by Mogul and Deering [34]. Packet loss ratio p can be easily obtained. However, the RTT cannot be measured through a simple feedback mechanism due to feedback implosion problem. A mechanism [53], based on RTCP protocol, has been proposed to estimate the RTT.

Step 2: Upon obtaining MTU, RTT, and p for a given period, the target rate λ can be computed through Eq. (1).

Step 3: Upon obtaining λ, a rate control action can be taken. If $\lambda < \gamma_0$, drop the base layer and stop receiving video (the network cannot deliver even the base layer due to congestion); otherwise, determine L', the largest integer such that $\sum_{i=0}^{L'} \gamma_i \leq \lambda$. If $L' > L$, add the layers from Layer $L+1$ to Layer L', and Layer L' becomes the highest layer currently subscribed (let $L = L'$); if $L' < L$, drop layers from Layer $L'+1$ to Layer L, and Layer L' becomes the highest layer currently subscribed (let $L = L'$). Return to Step 1.

The above algorithm has a potential problem when the number of receivers becomes large. If each receiver carries out the rate 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 join-experiments could aggravate network congestion. To coordinate the joining/leaving actions of the receivers, a scheme based on synchronization points [55] was proposed. With small protocol overhead, the proposed scheme in [55] helps to reduce the frequency and duration of join-experiments, resulting in a smaller possibility of congestion.

A.3 Hybrid Rate Control

Under the hybrid rate control, the receivers regulate the receiving rate of video streams by adding/dropping channels while the sender also adjusts the transmission rate of each channel based on feedback information from the receivers. Since the hybrid rate control consists of rate control at both the sender and a receiver, previous approaches described in Sections II-A.1 and II-A.2 can be employed.

The hybrid rate control is targeted at multicast video and is applicable to both layered video [44] and non-layered video [8]. Different from the source-based rate control framework where the sender uses a single channel, the hybrid rate control framework uses multiple channels. On the other hand, different from the receiver-based rate control framework where the rate for each channel is constant, the hybrid rate control enables the sender to dynamically

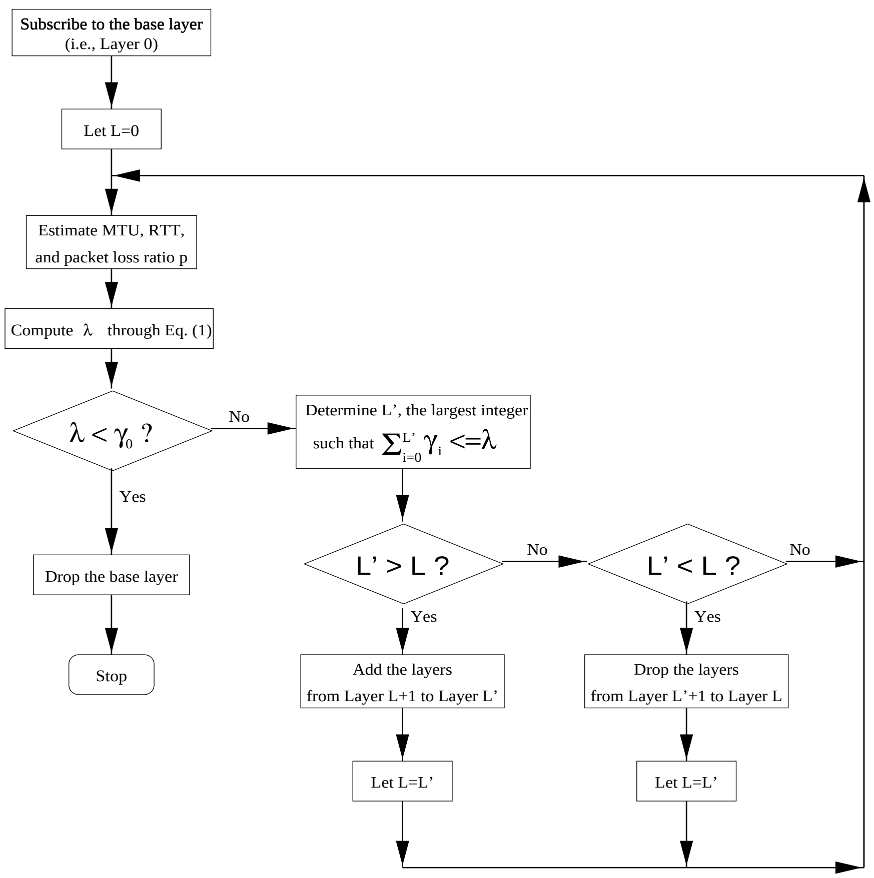

Fig. 9. Flow chart of the basic model-based rate control for a receiver.

change the rate for each channel based on congestion status.

One representative work using hybrid rate control is destination set grouping (DSG) protocol [8]. Before we present the DSG protocol, we first briefly describe the architecture associated with DSG. At the sender side, a raw video sequence is compressed into multiple streams (called replicated adaptive streams), which carry the same video information with different rate and quality. Different from layered video, each stream in DSG can be decoded independently. After compression, each video stream is sent to a separate IP multicast group. At the receiver side, each receiver can choose a multicast group to join by taking into account of its capability and congestion status. The receivers also send feedback to the source, and the source uses this feedback to adjust the transmission rate for each stream.

The DSG protocol consists of two main components:

1. **Rate control at the source.** For each stream, the rate control at the source is essentially the same as that used in IVS (see Section II-A.1). But the feedback control for each stream works independently.
2. **Rate control at a receiver.** A receiver can change its subscription and join a higher or lower quality stream based on network status, i.e., the fraction of UNLOADED, LOADED and CONGESTED receivers. The mechanism to obtain the fraction of UNLOADED, LOADED and CONGESTED receivers is similar to that used in IVS. The rate control at a receiver takes the probe-based approach as presented in Section II-A.2.

B. *Rate-adaptive Video Encoding: A Compression Approach*

Rate-adaptive video encoding has been studied extensively for various standards and applications, such as video conferencing with H.261 and H.263 [31], [61], storage media with MPEG-1 and MPEG-2 [16], [28], [48], real-time transmission with MPEG-1 and MPEG-2 [17], [23], and the recent object-based coding with MPEG-4 [54], [63]. The objective of a rate-adaptive encoding algorithm is to maximize the perceptual quality under a given encoding rate.[2] Such adaptive encoding can be achieved by the alteration of the encoder's quantization parameter (QP) and/or the alteration of the video frame rate.

Traditional video encoders (e.g., H.261, MPEG-1/2) typically rely on altering the QP of the encoder to achieve rate adaptation. These encoding schemes must perform coding with constant frame rates. 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 is not enough to achieve very low bit-rate, these encoding schemes may

[2]The given encoding rate can be either fixed or dynamically changing based on the network congestion status.

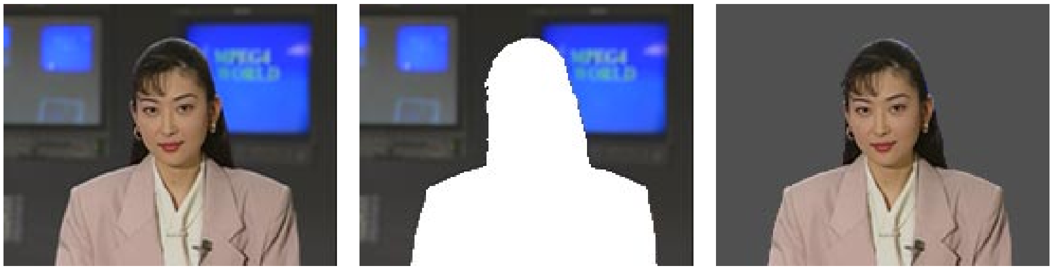

Fig. 10. An example of video object (VO) concept in MPEG-4 video. A video frame (left) is segmented into two VO planes where VO1 (middle) is the background and VO2 (right) is the foreground.

not be suitable for very low bit-rate video applications.

On the contrary, MPEG-4 and H.263 coding schemes are suitable for very low bit-rate video applications since they allow the alteration of the frame rate. In fact, the alteration of the frame rate is achieved by frame-skip.[3] Specifically, if the encoder buffer is in danger of overflow (i.e., the bit budget is over-used by the previous frame), a complete frame can be skipped at the encoder. This will allow the coded bits of the previous frames to be transmitted during the time period of this frame, therefore reducing the buffer level (i.e., keeping the encoded bits within the budget).

In addition, MPEG-4 is the first international standard addressing the coding of video objects (VO's) (see Fig. 10) [24]. With the flexibility and efficiency provided by coding video objects, MPEG-4 is capable of addressing interactive content-based video services as well as conventional stored and live video [39]. In MPEG-4, a frame of a video object is called a video object plane (VOP), which is encoded separately. Such isolation of video objects provides us with much greater flexibility to perform adaptive encoding. In particular, we can dynamically adjust target bit-rate distribution among video objects, in addition to the alteration of QP on each VOP (such a scheme is proposed in [63]). This can upgrade the perceptual quality for the regions of interest (e.g., head and shoulder) while lowering the quality for other regions (e.g., background).

For all the video coding algorithms, a fundamental problem is how to determine a suitable QP to achieve the target bit-rate. The rate-distortion (R-D) theory is a powerful tool to solve this problem. Under the R-D framework, there are two approaches for encoding rate control in the literature: the model-based approach and the operational R-D based approach. The model-based approach assumes various input distributions and quantizer characteristics [9], [63]. Under this approach, closed-form solutions can be obtained by using continuous optimization theory. On the other hand, the operational R-D based approach considers practical coding environments where only a finite set of quantizers is admissible [23], [28], [48], [61]. Under the operational R-D based approach, the admissible quantizers are used by the rate control algorithm to determine the optimal strategy to minimize the distortion under the constraint of a given bit budget. The optimal discrete solutions can be found through applying integer programming theory.

[3]Skipping a frame means that the frame is not encoded.

C. Rate Shaping

Rate shaping is a technique to adapt the rate of compressed video bit-streams to the target rate constraint. A rate shaper is an interface (or filter) between the encoder and the network, with which the encoder's output rate can be matched to the available network bandwidth. Since rate shaping does not require interaction with the encoder, rate shaping is applicable to any video coding scheme and is applicable to both live and stored video. Rate shaping can be achieved through two approaches: one is from the transport perspective [22], [45], [67] and the other is from the compression perspective [18].

A representative mechanism from the transport perspective is *server selective frame discard* [67]. The server selective frame discard is motivated by the following fact. Usually, a server transmits each frame without any awareness of the available network bandwidth and the client buffer size. As a result, the network may drop packets if the available bandwidth is less than required, which leads to frame losses. In addition, the client may also drop packets that arrive too late for playback. This causes wastage of network bandwidth and client buffer resources. To address this problem, the selective frame discard scheme preemptively drops frames at the server in an intelligent manner by considering available network bandwidth and client QoS requirements. The selective frame discard has two major advantages. First, by taking the network bandwidth and client buffer constraints into account, the server can make the best use of network resources by selectively discarding frames in order to minimize the likelihood of future frames being discarded, thereby increasing the overall quality of the video delivered. Second, unlike frame dropping in the network or at the client, the server can also take advantage of application-specific information such as regions of interest and group of pictures (GOP) structure, in its decision in discarding frames. As a result, the server optimizes the perceived quality at the client while maintaining efficient utilization of the network resources.

A representative mechanism from the compression perspective is dynamic rate shaping [18]. Based on the R-D theory, 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 less sensitive to higher frequencies, the dy-

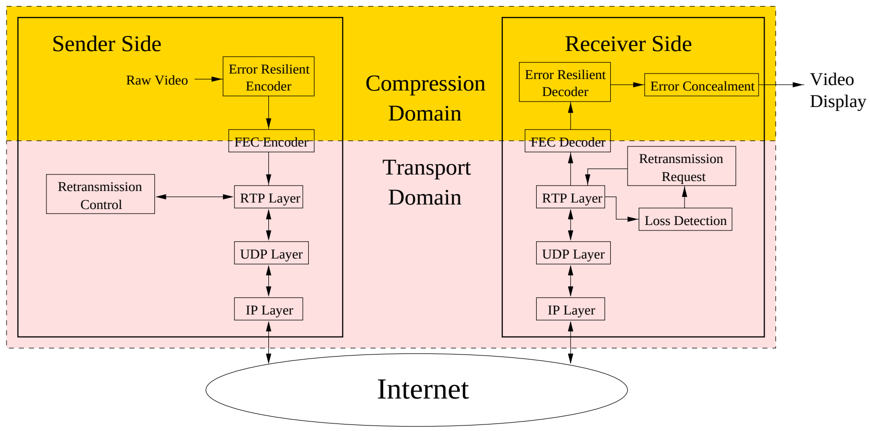

Fig. 11. An architecture for error control mechanisms.

namic rate shaper selects the highest frequencies and discards the DCT coefficients of these frequencies until the target rate is met.

Congestion control attempts to prevent packet loss by matching the rate of video streams to the available bandwidth in the network. However, packet loss is unavoidable in the Internet and may have significant impact on perceptual quality. Therefore, we need other mechanisms to maximize the video presentation quality in presence of packet loss. Such mechanisms include error control mechanisms, which are presented in the next section.

III. Error Control

In the Internet, packets may be dropped due to congestion at routers, they may be mis-routed, or they may reach the destination with such a long delay as to be considered useless or lost. Packet loss may severely degrade the visual presentation quality. To enhance the video quality in presence of packet loss, error control mechanisms have been proposed.

For certain types of data (such as text), packet loss is intolerable while delay is acceptable. When a packet is lost, there are two ways to recover the packet: the corrupted data must be corrected by traditional FEC (i.e., channel coding), or the packet must be retransmitted. On the other hand, for real-time video, some visual quality degradation is often acceptable while delay must be bounded. This feature of real-time video introduces many new error control mechanisms, which are applicable to video applications but not applicable to traditional data such as text. Basically, the error control mechanisms for video applications can be classified into four types, namely, FEC, retransmission, error-resilience, and error concealment. FEC, retransmission, and error-resilience are performed at both the source and the receiver side, while error concealment is carried out only at the receiver side. Fig. 11 shows the location of each error control mechanism in a layered architecture. As shown in Fig. 11, retransmission recovers packet loss from the transport perspective; error-resilience and error concealment deal with packet loss from the compression perspective; and FEC falls in both transport and compression domains. For the rest of this section, we present FEC, retransmission, error-resilience, and error concealment, respectively.

A. FEC

The use of FEC is primarily because of its advantage of small transmission delay, compared with TCP [14]. The principle of FEC is to add extra (redundant) information to a compressed video bit-stream so that the original video can be reconstructed in presence of packet loss. Based on the kind of redundant information to be added, the existing FEC schemes can be classified into three categories: (1) channel coding, (2) source coding-based FEC, and (3) joint source/channel coding, which will be presented in Sections III-A.1 to III-A.3, respectively.

A.1 Channel Coding

For Internet applications, channel coding is typically used in terms of block codes. Specifically, 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 $n > k$. Specifically, the channel encoder places the k packets into a group and then creates additional packets from them so that the total number of packets in the group becomes n, where $n > k$ (shown in Fig. 12). This group of packets is transmitted to the receiver, which receives K packets. To perfectly recover a segment, a user must receive K ($K \geq k$) packets in the n-packet block (see Fig. 12). In other words, a user only needs to receive any k packets in the n-packet block so that it can reconstruct all the original k packets.

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. In addition, due to its ability to recover from any k out of n packets regardless of which packets are 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

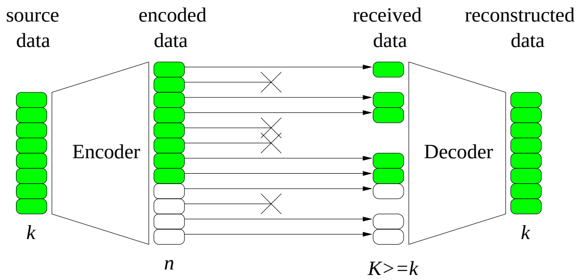

Fig. 12. Channel coding/decoding operation.

disadvantages associated with channel coding as follows.

1. It increases the 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 is, the higher the transmission rate is required to recover from the loss. The higher the transmission rate is, the more congested the network gets, which leads to an even higher loss rate. This makes channel coding vulnerable for short-term congestion. However, efficiency may be improved by using unequal error protection [1].
2. It increases delay. This is because (1) a channel encoder must wait for all k packets in a segment before it can generate the $n-k$ redundant packets; and (2) the receiver must wait for at least k packets of a block before it can playback the video segment. In addition, recovery from bursty loss requires the use of either longer blocks (i.e., larger k and n) or techniques like interleaving. In either case, delay will be further increased. But for video streaming applications, which can tolerate relatively large delay, the increase in delay may not be an issue.
3. It is not adaptive to varying loss characteristics and it works best only when the packet loss rate is stable. If more than $n-k$ packets of a block are lost, channel coding cannot recover any portion of the original segment. This makes channel coding useless when the short-term loss rate exceeds the recovery capability of the code. On the other hand, if the loss rate is well below the code's recovery capability, the redundant information is more than necessary (a smaller ratio n/k would be more appropriate). To improve the adaptive capability of channel coding, feedback can be used. That is, if the receiver conveys the loss characteristics to the source, the channel encoder can adapt the redundancy accordingly. Note that this requires a closed loop rather than an open loop in the original channel coding design.

A significant portion of previous research on channel coding for video transmission has involved *equal error protection* (EEP), in which all the bits of the compressed video stream are treated equally, and given an equal amount of redundancy. However, the compressed video stream typically does not consist of bits of equal significance. For example, in MPEG, an I-frame is more important than a P-frame while a P-frame is more important than a B-frame. Current research is heavily weighted towards *unequal error protection* (UEP) schemes, in which the more significant information bits are given more protection. A representative work of UEP is the Priority Encoding Transmission (PET) [1]. A key feature of the PET scheme is to allow a user to set different levels (priorities) of error protection for different segments of the video stream. This unequal protection makes PET efficient (less redundancy) and suitable for transporting MPEG video which has an inherent priority hierarchy (i.e., I-, P-, and B-frames).

To provide error recovery in layered multicast video, Tan and Zakhor proposed a receiver-driven hierarchical FEC (HFEC) [50]. In HFEC, additional streams with only FEC redundant information are generated along with the video layers. Each of the FEC streams is used for recovery of a different video layer, and each of the FEC streams is sent to a different multicast group. Subscribing to more FEC groups corresponds to higher level of protection. Like other receiver-driven schemes, HFEC also achieves good trade-off between flexibility of providing recovery and bandwidth efficiency, that is:

Flexibility of providing recovery: Each receiver can independently adjust the desired level of protection based on past reception statistics and the application's delay tolerance.

Bandwidth efficiency: Each receiver will subscribe to only as many redundancy layers as necessary, reducing overall bandwidth utilization.

A.2 Source Coding-based FEC

Source coding-based FEC (SFEC) is a recently devised variant of FEC for Internet video [4]. Like channel coding, SFEC also adds redundant information to recover from loss. For example, SFEC could add redundant information as follows: the nth packet contains the nth GOB (Group of Blocks) and redundant information about the $(n-1)$th GOB. If the $(n-1)$th packet is lost but the nth packet is received, the receiver can still reconstruct the $(n-1)$th GOB from the redundant information about the $(n-1)$th GOB, which is contained in the nth packet. However, the reconstructed $(n-1)$th GOB has a coarser quality. This is because the redundant information about the $(n-1)$th GOB is a compressed version of the $(n-1)$th GOB with a larger quantizer, resulting in less redundancy added to the nth packet.

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

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

Similar to channel coding, the disadvantages of SFEC

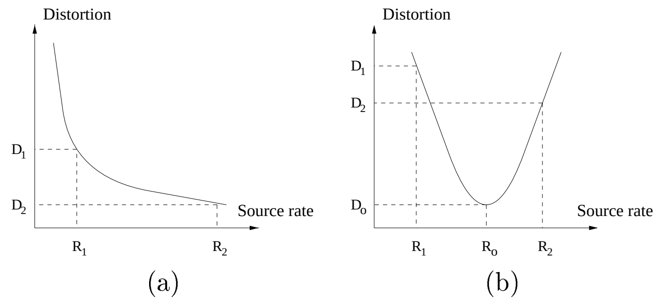

Fig. 13. (a) Rate-distortion relation for source coding. (b) Rate-distortion relation for the case of joint source/channel coding.

are: (1) an increase in the transmission rate, and (2) inflexible to varying loss characteristics. However, such inflexibility to varying loss characteristics can also be improved through feedback [4]. That is, if the receiver conveys the loss characteristics to the source, the SFEC encoder can adjust the redundancy accordingly. Note that this requires a close loop rather than an open loop in the original SFEC coding scheme.

A.3 Joint Source/Channel Coding

Due to Shannon's separation theorem [43], the coding world was generally divided into two camps: source coding and channel coding. The camp of source coding was concerned with developing efficient source coding techniques while the camp of channel coding was concerned with developing robust channel coding techniques [21]. In other words, the camp of source coding did not take channel coding into account and the camp of channel coding did not consider source coding. However, Shannon's separation theorem is not strictly applicable when the delay is bounded, which is the case for such real-time services as video over the Internet [10]. The motivation of joint source/channel coding for video comes from the following observations:

- **Case A:** According to the rate-distortion theory (shown in Fig. 13(a)) [13], the lower the source-encoding rate R for a video unit, the larger the distortion D of the video unit. That is, $R \downarrow \Rightarrow D \uparrow$.
- **Case B:** Suppose that the total rate (i.e., the source-encoding rate R plus the channel-coding redundancy rate R') is fixed and channel loss characteristics do not change. The higher the source-encoding rate for a video unit is, the lower the channel-coding redundancy rate would be. This leads to a higher probability P_c of the event that the video unit gets corrupted, which translates into a larger distortion of the video unit. That is, $R \uparrow \Rightarrow R' \downarrow \Rightarrow P_c \uparrow \Rightarrow D \uparrow$.

Combining Cases A and B, it can be argued that there exists an optimal source-encoding rate R_o that achieves the minimum distortion D_o (see Fig. 13(b)), given a constant total rate. As illustrated in Fig. 13(b), the left part of the curve shows Case A while the right part of the curve shows Case B. The two parts meet at the optimal point (R_o, D_o).

The objective of joint source/channel coding is to find the optimal point shown in Fig. 13(b) and design source/channel coding schemes to achieve the optimal point. In other words, finding an optimal point in joint source/channel coding is to make an optimal rate allocation between source coding and channel coding.

Basically, joint source/channel coding is accomplished by three tasks:

- **Task 1:** finding an optimal rate allocation between source coding and channel coding for a given channel loss characteristic,
- **Task 2:** designing a source coding scheme (including specifying the quantizer) to achieve its target rate,
- **Task 3:** designing/choosing channel codes to match the channel loss characteristic and achieve the required robustness.

For the purpose of illustration, Fig. 14 shows an architecture for joint source/channel coding. Under the architecture, a QoS monitor is kept at the receiver side to infer the channel loss characteristics. Such information is conveyed back to the source side through the feedback control protocol. Based on such feedback information, the joint source/channel optimizer makes an optimal rate allocation between the source coding and the channel coding (Task 1) and conveys the optimal rate allocation to the source encoder and the channel encoder. Then the source encoder chooses an appropriate quantizer to achieve its target rate (Task 2) and the channel encoder chooses a suitable channel code to match the channel loss characteristic (Task 3).

An example of joint source/channel coding is the scheme introduced by Davis and Danskin [14] for transmitting 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 a uniform application of channel codes is expensive.

B. Delay-constrained Retransmission: A Transport Approach

A conventional retransmission scheme, ARQ, works as follows: when packets are lost, the receiver sends feedback to notify the source; then 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 time after the transmission of the original packet, which might exceed the delay required by the application. However, if the one-way trip time is short with respect to the maximum allowable delay, a retransmission-based approach, called delay-constrained retransmission, is a viable option for error control [37], [38].

Typically, one-way trip time is relatively small within the same local area network (LAN). Thus, even delay sensitive interactive video applications could employ delay-

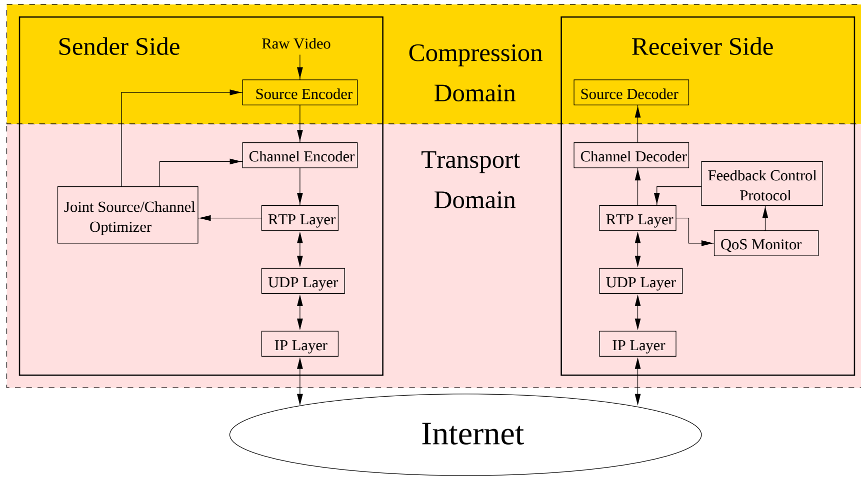

Fig. 14. An architecture for joint source/channel coding.

constrained retransmission for loss recovery in an LAN environment [15]. Delay-constrained retransmission may also be applicable to streaming video, which can tolerate relatively large delay due to a large receiver buffer and relatively long delay for display. As a result, even in wide area network (WAN), streaming video applications may have sufficient time to recover from lost packets through retransmission and thereby avoid unnecessary degradation in reconstructed video quality.

In the following, we present various delay-constrained retransmission schemes for unicast (Section III-B.1) and multicast (Section III-B.2), respectively.

B.1 Unicast

Based on who determines whether to send and/or respond to a retransmission request, we design three delay-constrained retransmission mechanisms for unicast, namely, receiver-based, sender-based, and hybrid control.

Receiver-based control. The objective of the receiver-based control is to minimize the requests of retransmission that will not arrive timely for display. Under the receiver-based control, the receiver executes the following algorithm.

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 could include tolerance of error in estimating RTT, the sender's response time to a request, and/or the receiver's processing delay (e.g., decoding). If $T_c + RTT + D_s < T_d(N)$ holds, it is expected that the retransmitted packet will arrive timely for display. The timing diagram for receiver-based control is shown in Fig. 15.

Sender-based control. The objective of the sender-based control is to suppress retransmission of packets that will

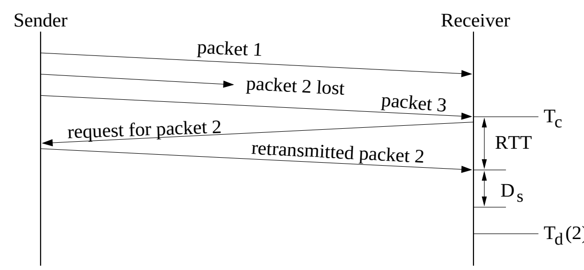

Fig. 15. Timing diagram for receiver-based control.

miss their display time at the receiver. Under the sender-based control, the sender executes the following algorithm.

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 feedback $T_d(N)$ 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 may include error terms in estimating T_o and $T'_d(N)$, as well as tolerance in the receiver's processing delay (e.g., decoding). If $T_c + RTT + D_s < T'_d(N)$ holds, it can be expected that retransmitted packet will reach the receiver in time for display. The timing diagram for sender-based control is shown in Fig. 16.

Hybrid control. The objective of the hybrid control is to minimize the request of retransmissions that will not arrive for timely display, and to suppress retransmission of the packets that will miss their display time at the receiver. The hybrid control is a simple combination of the sender-based control and the receiver-based control. Specifically, the receiver makes decisions on whether to send retransmission requests while the sender makes decisions on whether to disregard requests for retransmission. The hybrid control could achieve better performance at the cost of higher

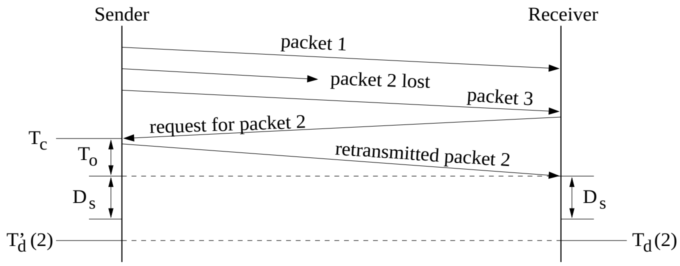

Fig. 16. Timing diagram for sender-based control.

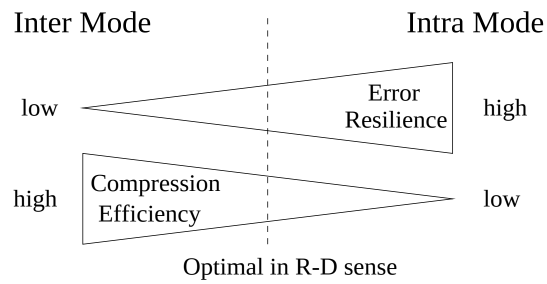

Fig. 17. Illustration of optimal mode selection.

complexity.

B.2 Multicast

In the multicast case, retransmission has to be restricted within closely located multicast members. This is because one-way trip times between these members tend to be small, making retransmissions effective in timely recovery. In addition, feedback implosion of retransmission requests is a problem that must be addressed under the retransmission-based approach. Thus, methods are required to limit the number or scope of retransmission requests.

Typically, a logical tree is configured to limit the number/scope of retransmission requests and to achieve local recovery among closely located multicast members [29], [32], [64]. 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 [29]. Or it can be dynamically constructed through the protocol used in STructure-Oriented Resilient Multicast (STORM) [64]. By adapting the tree structure to changing network traffic conditions and group memberships, the system could achieve higher probability of receiving retransmission timely.

Similar to the receiver-based control for unicast, receivers in a multicast group can make decisions on whether to send retransmission requests. By suppressing the requests for retransmission of those packets that cannot be recovered in time, bandwidth efficiency can be improved [29]. Besides, using a receiving buffer with appropriate size could not only absorb the jitter but also increase the likelihood of receiving retransmitted packets before their display time [29].

To address heterogeneity problem, a receiver-initiated mechanism for error recovery can be adopted as done in STORM [64]. Under this mechanism, each receiver can dynamically select the best possible DR to achieve good trade-off between desired latency and the degree of reliability.

C. Error-resilience: A Compression Approach

Error-resilient schemes address loss recovery from the compression perspective. Specifically, they attempt to prevent error propagation or limit the scope of the damage (caused by packet losses) on the compression layer. The standardized error-resilient tools include re-synchronization marking, data partitioning, and data recovery (e.g., reversible variable length codes (RVLC)) [24], [49]. However, re-synchronization marking, data partitioning, and data recovery are targeted at error-prone environments like wireless channels and may not be applicable to the Internet. For Internet video, the boundary of a packet already provides a synchronization point in the variable-length coded bit-stream at the receiver side. On the other hand, 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 present the standardized error-resilient tools. Instead, we present two techniques which are promising for robust Internet video transmission, namely, *optimal mode selection* and *multiple description coding*.

C.1 Optimal Mode Selection

In many video coding schemes, a block, which is a video unit, is coded by reference to a previously coded block so that only the difference between the two blocks needs to be coded, resulting in high coding efficiency. This is called *inter mode*. Constantly referring to previously coded blocks has the danger of error propagation. By occasionally turning off this inter mode, error propagation can be limited. But it will be more costly in bits to code a block all by itself, without any reference to a previously coded block. Such a coding mode is called *intra mode*. Intra-coding can effectively stop error propagation at the cost of compression efficiency while inter-coding can achieve compression efficiency at the risk of error propagation. Therefore, there is a trade-off in selecting a coding mode for each block (see Fig. 17). How to optimally make these choices is the subject of many research investigations [12], [62], [66].

For video communication over a network, a block-based coding algorithm such as H.263 or MPEG-4 [24] usually employs rate control to match the output rate to the available network 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 block and the reconstructed one under a given bit budget [36], [46], 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 opti-

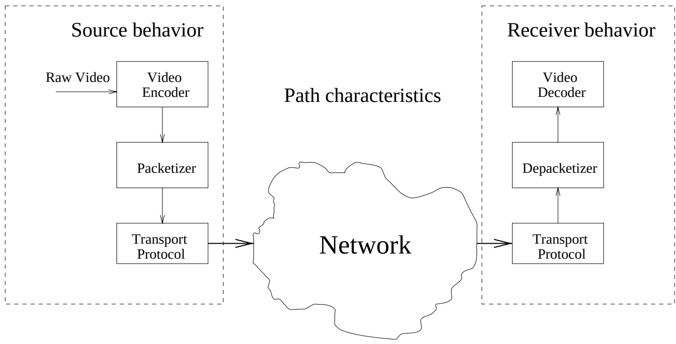

Fig. 18. Factors that have impact on the video presentation quality: source behavior, path characteristics, and receiver behavior.

mality 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 to R-D optimized mode selection was proposed [62]. Under the end-to-end approach, three factors were identified to have impact on the video presentation quality at the receiver: (1) the source behavior, e.g., quantization and packetization, (2) the path characteristics, and (3) the receiver behavior, e.g., error concealment (see Fig. 18). Based on the characterizations, a theory [62] for globally optimal mode selection was developed. By taking into consideration of the network congestion status and the receiver behavior, the end-to-end approach is shown to be capable of offering superior performance over the classical approach for Internet video applications [62].

C.2 Multiple Description Coding

Multiple description coding (MDC) is another way to achieve trade-off between compression efficiency and robustness to packet loss [59]. With MDC, a raw video sequence is compressed into multiple streams (referred to as descriptions). Each description provides acceptable visual quality; more combined descriptions provide a better visual quality. The advantages of MDC are: (1) *robustness to loss:* even if a receiver gets only one description (other descriptions being lost), it can still reconstruct video with acceptable quality; and (2) *enhanced quality:* if a receiver gets multiple descriptions, it can combine them together to produce a better reconstruction than that produced from any single description.

However, the advantages do not come for free. 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). In addition, although more combined descriptions provide a better visual quality, a certain degree of correlation between the multiple descriptions has to be embedded in each description, resulting in further reduction of the compression efficiency. Current research effort is to find a good trade-off between the compression efficiency and the reconstruction quality from one description.

D. Error Concealment: A Compression Approach

When packet loss is detected, the receiver can employ error concealment to conceal the lost data and make the presentation more pleasing to human eyes. Since human eyes can tolerate a certain degree of distortion in video signals, error concealment is a viable technique to handle packet loss [60].

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

In recent years, numerous error-concealment schemes have been proposed in the literature (refer to [60] for a good survey). Examples include maximally smooth recovery [58], projection onto convex sets [47], and various motion vector and coding mode recovery methods such as motion compensated temporal prediction [20]. However, most error concealment techniques discussed in [60] are only applicable to either ATM or wireless environments, and require substantial additional computation complexity, which is acceptable for decoding still images but not tolerable in decoding real-time video. Therefore, we only describe simple error concealment schemes that are applicable to Internet video communication.

We describe three simple error concealment (EC) schemes as follows.

EC-1: The receiver replaces the whole frame (where some blocks are corrupted due to packet loss) with the previous reconstructed frame.

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

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.

IV. Summary

Transporting video over the Internet is an important component of many multimedia applications. Lack of QoS support in the current Internet, and the heterogeneity of the networks and end-systems pose many challenging problems for designing video delivery systems. In this paper, we identified four problems for video delivery systems: bandwidth, delay, loss, and heterogeneity. There are two general approaches that address these problems: the network-centric approach and the end system-based approach. We are concerned with the mechanisms that follow the end system-based approach.

TABLE I
TAXONOMY OF THE DESIGN SPACE

		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 resilience		Optimal mode selection
			Multiple description coding
	Error concealment		*EC*-1, *EC*-2, *EC*-3

TABLE II
RATE CONTROL

		Model-based approach	Probe-based approach
Rate control	Source-based	Unicast	Unicast/Multicast
	Receiver-based	Multicast	Multicast
	Hybrid		Multicast

Over the past several years, extensive research based on the end system-based approach has been conducted and various solutions have been proposed. To depict a big picture, we took a holistic approach from both transport and compression perspectives. With the holistic approach, we presented a framework for transporting real-time Internet video, which consisted of two components: congestion control and error control. We have described various approaches and schemes for the two components. All the possible approaches/schemes for the two components can form a design space. As shown in Table I, the approaches/schemes in the design space can be classified along two dimensions: the transport perspective and the compression perspective.

To give the reader a clear picture of this design space, we summarize the advantages and disadvantages of the approaches and schemes as follows.

1. Congestion control. There are three mechanisms for congestion control: rate control, rate adaptive video encoding, and rate shaping. Rate control schemes can be classified into three categories: source-based, receiver-based, and hybrid. As shown in Table II, rate control schemes can follow either the model-based approach or the probe-based approach. Source-based rate control is primarily targeted at unicast and can follow either the model-based approach or the probe-based approach. If applied in multicast, source-based rate control can only follow the probe-based approach. Source-based rate control needs another component to enforce the rate on the video stream. This component could be either rate adaptive video encoding or rate shaping. Examples of combining source-based rate control with rate adaptive video encoding can be found in [51], [63]. Examples of combining source-based rate control with rate shaping include [25]. Receiver-based and hybrid rate control were proposed to address the heterogeneity problem in multicast video. The advantage of receiver-based control over sender-based control is that the burden of adaptation is moved from the sender to the receivers, resulting in enhanced service flexibility and scalability. Receiver-based rate control can follow either the model-based approach or the probe-based approach. Hybrid rate control combines some of the best features of receiver-based and sender-based control in terms of service flexibility and bandwidth efficiency. But it can only follow the probe-based approach. For video multicast, one advantage of the model-based approach over the probe-based approach is that it does not require exchange of information among the group as is done under the probe-based approach. Therefore, it eliminates processing at each receiver and the bandwidth usage associated with information exchange.

2. Error control. It takes the form of FEC, delay-constrained retransmission, error-resilience or error concealment. There are three kinds of FEC: channel coding, source coding-based FEC, and joint source/channel coding. The advantage of all FEC schemes over TCP is reduction in video transmission latency. Source coding-based FEC can achieve lower delay than channel cod-

ing while joint source/channel coding could achieve optimal performance in a rate-distortion sense. The disadvantages of all FEC schemes are: increase in the transmission rate, and inflexibility to varying loss characteristics. A feedback mechanism can be used to improve FEC's inflexibility. Unlike FEC, which adds redundancy to recover from loss that might not occur, a retransmission-based scheme only re-sends the packets that are lost. Thus, a retransmission-based scheme is adaptive to varying loss characteristics, resulting in efficient use of network resources. But delay-constrained retransmission-based schemes may become useless when the round trip time is too large. Optimal mode selection and multiple description coding are two error-resilient mechanisms recently proposed. Optimal mode selection achieves the best trade-off between compression efficiency and error resilience in an R-D sense. The cost of optimal mode selection is its complexity, which is similar to that of motion compensation algorithms. Multiple description coding is another way of trading off compression efficiency with robustness to packet loss. The advantage of MDC is its robustness to loss and enhanced quality. The cost of MDC is reduction in compression efficiency. Finally, as the last stage of a video delivery system, error concealment can be used in conjunction with any other techniques (i.e., congestion control, FEC, retransmission, and error-resilience).

The reasons why we divide the design space along two dimensions (transport and compression) lie in that: (1) We find that a conventional mechanism from one perspective can 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); ARQ (pure transport) is substituted by delay-constrained retransmission (considering characteristics of compression layer); traditional error recovery mechanisms (channel coding and ARQ) are pure transport techniques while new mechanisms (e.g., error-resilient mechanisms) try to address error recovery from the compression perspective. (2) There are much room in the design space from both transport and compression perspective. For example, joint source/channel coding combines the best features of both transport and compression techniques; periodic temporal dependency distance (PTDD) [40] is capable of preventing error propagation on the compression layer (compression) with retransmissions (transport); conveying back the address of erroneous blocks to the source (transport) could help the encoder prevent error propagation (compression) [56].

As shown in the paper, a framework for transporting real-time video over the Internet includes two components: congestion control and error control. We stress that overlook of any of the two components would degrade the overall performance. We also have discussed the design of each component, which can be achieved by either a transport approach or a compression approach. Recently, there have been extensive efforts on the combined approaches [14], [40], [56], [62]. We expect that the synergy of transport and compression could provide better solutions in the design of video delivery systems.

A promising future research direction is to combine the end system-based control techniques discussed in this paper with QoS support from the network. The motivation is as follows. Different from the case in circuit-switched networks, in packet-switched networks, flows are statistically multiplexed onto physical links and no flow is isolated. To achieve high statistical multiplexing gain or high resource utilization in the network, occasional violations of hard QoS guarantees (called statistical QoS) are allowed. For example, the delay of 95% packets is within the delay bound while 5% packets are not guaranteed to have bounded delays. The percentage (e.g., 95%) is in an average sense. In other words, a certain flow may have only 10% packets arriving within the delay bound while the average for all flows is 95%. The statistical QoS service only guarantees the average performance, rather than the performance for each flow. In this case, if the end system-based control is employed for each video stream, higher presentation quality can be achieved since the end system-based control is capable of adapting to the short-term violations.

As a final note, we would like to point out that each scheme has a trade-off between cost/complexity and performance. We have identified a design space that can be explored by video application designers and have provided insights on the trade-offs of each mechanism in the design space. Designers can choose a scheme in the design space that meets the specific cost/performance objectives.

Acknowledgments

The authors would like to thank the anonymous reviewers for their constructive comments that helped to improve the presentation of this paper.

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, pp. 1737–1744, Nov. 1996.
[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," *RFC 2475,* Internet Engineering Task Force, Dec. 1998.
[3] J-C. Bolot, T. Turletti, and I. Wakeman, "Scalable feedback control for multicast video distribution in the Internet," in *Proc. ACM SIGCOMM'94,* pp. 58–67, London, UK, Sept. 1994.
[4] J-C. Bolot and T. Turletti, "Adaptive error control for packet video in the Internet," in *Proc. IEEE Int. Conf. on Image Processing (ICIP'96),* pp. 25–28, Lausanne, Switzerland, Sept. 1996.
[5] J-C. Bolot and T. Turletti, "Experience with control mechanisms for packet video in the Internet," *ACM Computer Communication Review,* vol. 28, no. 1, Jan. 1998.
[6] R. Braden, D. Clark, and S. Shenker, "Integrated services in the Internet architecture: An overview," *RFC 1633,* Internet Engineering Task Force, July 1994.
[7] B. Braden, D. Black, J. Crowcroft, B. Davie, S. Deering, D. Estrin, S. Floyd, V. Jacobson, G. Minshall, C. Partridge, L. Peterson, K. Ramakrishnan, S. Shenker, J. Wroclawski, and L. Zhang, "Recommendations on queue management and congestion avoidance in the Internet," *RFC 2309,* Internet Engineering Task Force, April 1998.
[8] S. Y. Cheung, M. Ammar, and X. Li, "On the use of destination set grouping to improve fairness in multicast video distribution," in *Proc. IEEE INFOCOM'96,* pp. 553–560, San Francisco, CA, March 1996.

[9] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 7, no. 1, pp. 246–250, Feb. 1997.

[10] P. A. Chou, "Joint source/channel coding: a position paper," in *Proc. NSF Workshop on Source-Channel Coding,* San Diego, CA, USA, Oct. 1999.

[11] D. Clark, S. Shenker, and L. Zhang, "Supporting real-time applications in an integrated services packet network: architecture and mechanisms," in *Proc. ACM SIGCOMM'92,* Baltimore, MD, Aug. 1992.

[12] G. Cote and F. Kossentini, "Optimal intra coding of blocks for robust video communication over the Internet," to appear in *EUROSIP Image Communication Special Issue on Real-time Video over the Internet.*

[13] T. Cover and J. Thomas, *Elements of information theory,* John Wiley & Sons, New York, 1991.

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

[15] B. J. Dempsey, J. Liebeherr, and A. C. Weaver, "On retransmission-based error control for continuous media traffic in packet-switching networks," *Computer Networks and ISDN Systems,* vol. 28, no. 5, pp. 719–736, March 1996.

[16] W. Ding and B. Liu, "Rate control of MPEG video coding and recording by rate-quantization modeling," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 6, pp. 12–20, Feb. 1996.

[17] W. Ding, "Joint encoder and channel rate control of VBR video over ATM networks," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 7, pp. 266–278, April 1997.

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

[19] 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.

[20] M. Ghanbari, "Cell-loss concealment in ATM video codes," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 3, pp. 238–247, June 1993.

[21] J. Hagenauer, "Source-controlled channel decoding," *IEEE Trans. on Communications,* vol. 43, no. 9, pp. 2449–2457, Sept. 1995.

[22] M. Hemy, U. Hengartner, P. Steenkiste, and T. Gross, "MPEG system streams in best-effort networks," in *Proc. IEEE Packet Video'99,* New York, April 26–27, 1999.

[23] C. Y. Hsu, A. Ortega, and A. R. Reibman, "Joint selection of source and channel rate for VBR transmission under ATM policing constraints," *IEEE Journal on Selected Areas in Communications,* vol. 15, pp. 1016–1028, Aug. 1997.

[24] 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.

[25] S. Jacobs and A. Eleftheriadis, "Streaming video using TCP flow control and dynamic rate shaping," *Journal of Visual Communication and Image Representation,* vol. 9, no. 3, pp. 211–222, Sept. 1998.

[26] V. Jacobson, "Congestion avoidance and control," in *Proc. ACM SIGCOMM'88,* pp. 314–329, Aug. 1988.

[27] V. Jacobson, K. Nichols, and K. Poduri, "An expedited forwarding PHB," *RFC 2598,* Internet Engineering Task Force, June 1999.

[28] J. Lee and B.W. Dickenson, "Rate-distortion optimized frame type selection for MPEG encoding," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 7, pp. 501–510, June 1997.

[29] X. Li, S. Paul, P. Pancha, and M. H. Ammar, "Layered video multicast with retransmissions (LVMR): evaluation of error recovery schemes," in *Proc. IEEE Int. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97),* pp. 161–172, May 1997.

[30] X. Li, S. Paul, and M. H. Ammar, "Layered video multicast with retransmissions (LVMR): evaluation of hierarchical rate control," in *Proc. IEEE INFOCOM'98,* vol. 3, pp. 1062–1072, March 1998.

[31] F. C. Martins, W. Ding, and E. Feig, "Joint control of spatial quantization and temporal sampling for very low bit-rate video," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP'96),* vol. 4, pp. 2072–2075, May 1996.

[32] N. Maxemchuk, K. Padmanabhan, and S. Lo, "A cooperative packet recovery protocol for multicast video," in *Proc. IEEE Int. Conference on Network Protocols (ICNP'97),* pp. 259–266, Oct. 1997.

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

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

[35] K. Nichols, V. Jacobson, and L. Zhang, "A two-bit differentiated services architecture for the Internet," *RFC 2638,* Internet Engineering Task Force, July 1999.

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

[37] M. Podolsky, M. Vetterli, and S. McCanne, "Limited retransmission of real-time layered multimedia," in *Proc. IEEE Workshop on Multimedia Signal Processing,* pp. 591–596, Dec. 1998.

[38] M. Podolsky, K. Yano, and S. McCanne, "A RTCP-based retransmission protocol for unicast RTP streaming multimedia," *Internet draft,* Internet Engineering Task Force, Oct. 1999, work in progress.

[39] K. R. Rao, "MPEG-4 — the emerging multimedia standard," in *Proc. IEEE International Caracas Conference on Devices, Circuits and Systems,* pp. 153–158, March 1998.

[40] I. Rhee, "Error control techniques for interactive low-bit-rate video transmission over the Internet," in *Proc. ACM SIGCOMM'98,* Vancouver, Canada, Aug. 1998.

[41] H. Schulzrinne, S. Casner, R. Frederick, and V. Jacobson, "RTP: a transport protocol for real-time applications," *RFC 1889,* Internet Engineering Task Force, Jan. 1996.

[42] S. Shenker, C. Partridge, and R. Guerin, "Specification of guaranteed quality of service," *RFC 2212,* Internet Engineering Task Force, Sept. 1997.

[43] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal,* vol. 27, 1948.

[44] H. M. Smith, M. W. Mutka, and E. Torng, "Bandwidth allocation for layered multicasted video," in *Proc. IEEE Int. Conference on Multimedia Computing and Systems,* vol. 1, pp. 232–237, June 1999.

[45] K. Sripanidkulchai and T. Chen, "Network-adaptive video coding and transmission," in *SPIE Proc. Visual Communications and Image Processing (VCIP'99),* San Jose, CA, Jan. 1999.

[46] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine,* pp. 74–90, Nov. 1998.

[47] H. Sun and W. Kwok, "Concealment of damaged block transform coded images using projection onto convex sets," *IEEE Trans. on Image Processing,* vol. 4, pp. 470–477, Apr. 1995.

[48] H. Sun, W. Kwok, M. Chien, and C. H. J. Ju, "MPEG coding performance improvement by jointly optimizing coding mode decision and rate control," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 7, pp. 449–458, June 1997.

[49] R. Talluri, "Error-resilience video coding in the ISO MPEG-4 standard," *IEEE Communications Magazine,* pp. 112–119, June 1998.

[50] W. Tan and A. Zakhor, "Multicast transmission of scalable video using receiver-driven hierarchical FEC," in *Proc. Packet Video Workshop 1999,* New York, April 1999.

[51] W. Tan and A. Zakhor, "Real-time Internet video using error resilient scalable compression and TCP-friendly transport protocol," *IEEE Trans. on Multimedia,* vol. 1, no. 2, pp. 172–186, June 1999.

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

[53] 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.
[54] A. Vetro, H. Sun, and Y. Wang, "MPEG-4 rate control for multiple video objects," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 9, no. 1, pp. 186–199, Feb. 1999.
[55] L. Vicisano, L. Rizzo, and J. Crowcroft, "TCP-like congestion control for layered multicast data transfer," in *Proc. IEEE INFOCOM'98,* vol. 3, pp. 996–1003, March 1998.
[56] J.-T. Wang and P.-C. Chang, "Error-propagation prevention technique for real-time video transmission over ATM networks," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 9, no. 3, April 1999.
[57] X. Wang and H. Schulzrinne, "Comparison of adaptive Internet multimedia applications," *IEICE Trans. on Communications,* vol. E82-B, no. 6, pp. 806–818, June 1999.
[58] Y. Wang, Q.-F. Zhu, and L. Shaw, "Maximally smooth image recovery in transform coding," *IEEE Trans. on Communications,* vol. 41, no. 10, pp. 1544–1551, Oct. 1993.
[59] Y. Wang, M. T. Orchard, and A. R. Reibman, "Multiple description image coding for noisy channels by pairing transform coefficients," in *Proc. IEEE Workshop on Multimedia Signal Processing,* pp. 419–424, June 1997.
[60] 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.
[61] T. Weigand, M. Lightstone, D. Mukherjee, T. G. Campbell, and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit-rate video coding and the emerging H.263 standard," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 6, pp. 182–190, April 1996.
[62] 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," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, pp. 977–995, June 2000.
[63] 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," *IEEE Trans. on Circuits and Systems for Video Technology,* vol. 10, no. 6, Sept. 2000.
[64] X. R. Xu, A. C. Myers, H. Zhang, and R. Yavatkar, "Resilient multicast support for continuous-media applications," in *Proc. IEEE Int. Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'97),* pp. 183–194, May 1997.
[65] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A new resource ReSerVation Protocol," *IEEE Network Magazine,* vol. 7, no, 5, pp. 8–18, Sept. 1993.
[66] R. Zhang, S. L. Regunathan, and K. Rose, "Video coding with optimal inter/intra-mode switching for packet loss resilience," *IEEE Journal on Selected Areas in Communications,* vol. 18, no. 6, pp. 966–976, June 2000.
[67] 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," in *Proc. IEEE INFOCOM'99,* pp. 472–479, New York, March 1999.

Dapeng Wu (S'98) received the B.E degree from Huazhong University of Science and Technology, Wuhan, China, and the M.E. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 1990 and 1997 respectively, both in Electrical Engineering. From July 1997 to December 1999, he conducted graduate research at Polytechnic University, Brooklyn, New York. Since January 2000, he has been working towards his Ph.D. degree in Electrical Engineering at Carnegie Mellon University, Pittsburgh, PA. During the summers of 1998, 1999 and 2000, he conducted research at Fujitsu Laboratories of America, Sunnyvale, California, on architectures and traffic management algorithms in integrated services (IntServ) networks and differentiated services (DiffServ) Internet for multimedia applications. His current interests are in the areas of rate control and error control for video communications over the Internet and wireless networks, and next generation Internet architecture, protocols, implementations for integrated and differentiated services. He is a student member of the IEEE and the ACM.

Yiwei Thomas Hou (S'91–M'98) obtained his B.E. degree (*Summa Cum Laude*) from the City College of New York in 1991, the M.S. degree from Columbia University in 1993, and the Ph.D. degree from Polytechnic University, Brooklyn, New York, in 1997, all in Electrical Engineering. He was awarded a five-year National Science Foundation Graduate Research Traineeship for pursuing Ph.D. degree in high speed networking, and was recipient of the Alexander Hessel award for outstanding Ph.D. dissertation (1997–1998 academic year) from Polytechnic University. While a graduate student, he worked at AT&T Bell Labs, Murray Hill, New Jersey, during the summers of 1994 and 1995, on internetworking of IP and ATM networks; he conducted research at Bell Labs, Lucent Technologies, Holmdel, New Jersey, during the summer of 1996, on fundamental problems on network traffic management.

Since September 1997, Dr. Hou has been a Researcher Scientist at Fujitsu Laboratories of America, Sunnyvale, California. He received several awards from Fujitsu Laboratories of America for intellectual property contributions. His current research interests are in the areas of scalable architecture, protocols, and implementations for differentiated services Internet; terabit switching; and quality of service (QoS) support for multimedia over IP networks. He has authored or co-authored over 50 refered papers in the above areas, including over 20 papers in major international archival journals. Dr. Hou is a member of the IEEE, ACM, Sigma Xi, and New York Academy of Sciences.

Ya-Qin Zhang (S'87–M'90–SM'93–F'97) is currently the Managing Director of Microsoft Research China. He was previously the Director of Multimedia Technology Laboratory at Sarnoff Corporation in Princeton, New Jersey (formerly David Sarnoff Research Center and RCA Laboratories). His laboratory is a world leader in MPEG2/DTV, MPEG4/VLBR, and multimedia information technologies. He was with GTE Laboratories Inc. in Waltham, Massachusetts, and Contel Technology Center in Chantilly, Virginia from 1989 to 1994. He has authored and co-authored over 150 refereed papers and 30 US patents granted or pending in digital video, Internet multimedia, wireless and satellite communications. Many of the technologies he and his team developed have become the basis for start-up ventures, commercial products, and international standards.

Dr. Zhang was Editor-In-Chief for the *IEEE Transactions on Circuits and Systems for Video Technology* from July 1997 to July 1999. He was a Guest Editor for the special issue on "Advances in Image and Video Compression" for the *Proceedings of the IEEE* (February 1995). He serves on the editorial boards of seven other professional journals and over a dozen conference committees. He has been an active contributor to the ISO/MPEG and ITU standardization efforts in digital video and multimedia.

Dr. Zhang is a Fellow of IEEE. He received numerous awards, including several industry technical achievement awards and IEEE awards. He was awarded as the "Research Engineer of the Year" in 1998 by the Central Jersey Engineering Council for his "leadership and invention in communications technology, which has enabled dramatic advances in digital video compression and manipulation for broadcast and interactive television and networking applications."