An End-to-End Architecture for MPEG-4 Video Streaming over the Internet

Y. Thomas Hou* Dapeng Wu† Wenwu Zhu‡ Hung-Ju Lee§
Tihao Chiang¶ Ya-Qin Zhang‖

Abstract

It is a challenging problem to design an efficient MPEG-4 video delivery system that can maximize the perceptual quality while achieving high resource utilization. This paper addresses this problem by presenting an architecture of transporting MPEG-4 video over the Internet, which includes an end-to-end feedback control algorithm and a source encoding rate control algorithm. Our feedback control algorithm is capable of estimating the available bandwidth in the network based on the feedback information from the receiver, while our source encoding rate control algorithm is able to adjust the encoding rate of MPEG-4 video to the desired rate. Simulation results demonstrate that our architecture achieves good perceptual picture quality under low bit-rate and varying network conditions while efficiently utilizing network resources.

1 Introduction

With the success of Internet and the emerging of multimedia communication era, the new international standard, MPEG-4, is poised to become the enabling technology for multimedia communications in the next millennium [3]. MPEG-4 builds on elements from several successful technologies such as digital video, computer graphics, and the World Wide Web with the aim of providing powerful tools in the production, distribution, and display of multimedia contents with unprecedented new features and functions. With the extreme flexibility and efficiency provided by coding a new form of visual data called visual object (VO), it is foreseen that MPEG-4 will be capable of addressing emerging truly interactive content-based video services as well as conventional storage and transmission of video.

*Y. T. Hou is with Fujitsu Laboratories of America, Sunnyvale, CA, USA.

†D. Wu is with Polytechnic University, Brooklyn, NY, USA.

‡W. Zhu is with Bell Labs, Lucent Technologies, Murray Hill, NJ, USA.

§H.-J. Lee is with Sarnoff Corporation, Princeton, NJ, USA.

¶T. Chiang is with Sarnoff Corporation, Princeton, NJ, USA.

‖Y.-Q. Zhang is with Microsoft Research, Beijing, China.

Recently, Internet video streaming has brought enormous interests from both industry and academia. Video streaming typically has delay and loss requirements.[1] However, delay and loss in the current Internet are drastically varying from time to time, which is very detrimental to video streaming. Thus, it is a major challenge to design an efficient video delivery system that can maximize users' perceived quality of service (QoS) while achieving efficient network resource utilization.

This paper presents an end-to-end transport architecture for MPEG-4 video over the Internet. The objective of our architecture is to achieve good perceptual quality at the application level while being adaptive to network condition and utilizing network resource efficiently. More specifically, our framework consists of two key components: (1) an end-to-end feedback control algorithm; and (2) an adaptive video encoding rate control algorithm. Simulation results show that our MPEG-4 transport architecture is capable of transporting MPEG-4 video over the network with good perceptual quality under low bit-rate and varying network conditions and utilize network resource efficiently.

Previous work on feedback control for Internet video include [6]. Since this algorithm employs multiplicative rate increase, there is frequent and large rate fluctuation, which leads to large packet loss ratio. This paper extends the feedback control technique in [6] for MPEG-4 video. In particular, we design an end-to-end feedback control algorithm for MPEG-4 video by employing the RTCP feedback mechanism and using packet loss as congestion indication.

Our source encoding rate control in this paper is based on the following concepts and techniques: (1) an accurate second-order rate distortion model for the target bit-rate estimation, (2) a sliding window method for smoothing the impact of scene change, (3) an adaptive data-point selection criterion for model

[1] Streaming video implies that the content need not be downloaded in full before it begins playing, but is played out even as it is being received and decoded.

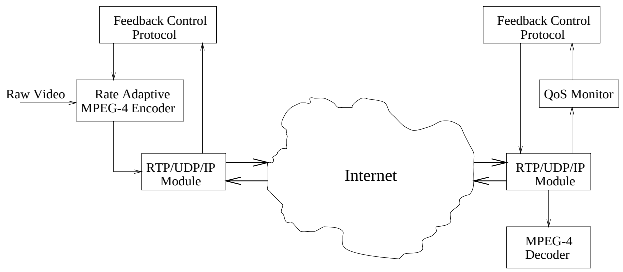

Figure 1: An architecture of transporting MPEG-4 video over the Internet.

updating process, (4) an adaptive threshold shape control for optimal use of bit budget, and (5) a dynamic bit-rate allocation among video objects with different coding complexities. This algorithm has been accepted in the international standard for MPEG-4 [3].

The remainder of this paper is organized as follows. Section 2 gives an overview of our MPEG-4 transport architecture. In Section 3, we design an end-to-end feedback control algorithm for MPEG-4 video streaming. Section 4 describes the video encoding rate control algorithm. In Section 5, we use simulation results to demonstrate the performance of MPEG-4 video streaming under our transport architecture. Section 6 concludes this paper.

2 Transporting MPEG-4 Video Streams over the Internet

Figure 1 gives an overview of our architecture for transporting MPEG-4 video over the Internet. In essence, the MPEG-4 video encoder is controlled by a feedback control algorithm and performs adaptive encoding based on feedback so that it can respond gracefully to network congestion.

Under our architecture, we design an end system (sender and receiver) based feedback control for MPEG-4 video using the RTCP flow control mechanism. At the receiver side, the receiver monitors the packet loss ratio of the arriving packet stream and periodically conveys packet loss information to the source with RTCP feedback control packets. At the source side, there is a rate calculation algorithm that estimates the the available network bandwidth based on packet loss information in the returning RTCP packets. Section 3 presents the details of our feedback control algorithm.

Based on the estimated rate by the feedback control algorithm, the source encoder performs compression so that the output rate matches the estimated rate. A good source encoding rate control should not only be able to produce the desired output rate, but also maximize the perceptual quality. Under MPEG-4 environment, adaptive encoding may be achieved by alteration of both the encoder's quantization parameters and the video frame rate. In Section 4, we will describe our source encoding rate control algorithm for MPEG-4. Our algorithm is capable of achieving the desired output rate with excellent perceptual quality and has been accepted in the international standard for MPEG-4 [3].

3 A Feedback Control Algorithm

Under our architecture, we let the MPEG-4 video source gradually increase its transmission rate to probe available network bandwidth. Such rate increase will first have the source's rate reach the available network bandwidth. Then the source rate will overshoot the available network bandwidth and fall into the congestion region. Congestion is detected by the receiver through packet loss in the received packets. The receiver sends feedback RTCP packets to the source to inform it about congestion status. Once source receives such feedback, it decreases it transmission rate.

In consistent with the RTP/RTCP standard [4], we let the source periodically send one RTCP control packet for every N_s MPEG-4 video packets. The receiver sends a feedback RTCP control packet to the source upon receiving N_r packets or at least once every 5 seconds. The returning RTCP packet contains the packet loss ratio P_{loss} the receiver observed during the N_r packet time interval since the previous feedback RTCP packet.

Algorithm 1 Feedback Control Algorithm

Sender Behavior

- The sender starts to transmit at the output rate $r :=$ InitRate, which is greater than or equal to its minimum rate MinRate;
- For every N_s transmitted RTP data packets, the sender sends a forward RTCP control packet;
- Upon the receipt of a backward RTCP with the packet loss ratio P_{loss} from the receiver, the output rate r at the source is adjusted according to the following rule:

$$\begin{array}{l} \text{if } (P_{loss} \leq P_{threshold}) \\ \quad r := \min\{(r + \text{AddRate}), \text{MaxRate}\}; \\ \text{else} \\ \quad r := \max\{(\alpha \times r), \text{MinRate}\} \end{array}$$

Receiver Behavior

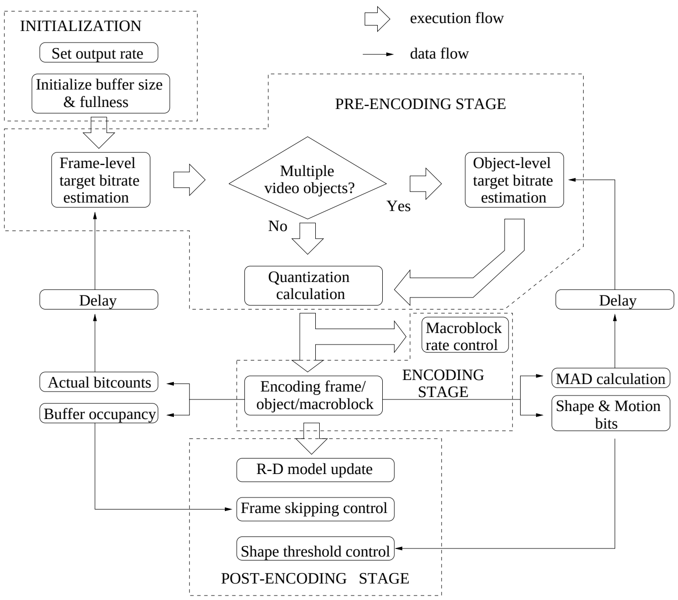

Figure 2: Source encoding rate control.

- The receiver keeps track of the sequence number in the RTP header of the arriving packets;
- Upon receiving N_r packets or at most 5 seconds, the receiver sends a feedback RTCP packet to the source containing packet loss rate P_{loss} it observes during this time interval. □

4 Source Encoding Rate Control

A good source encoding rate control should not only be able to produce the desired output rate, but also maximize the perceptual quality. Figure 2 is a block diagram of our source encoding rate control algorithm.

Our source encoding rate control is performed in four stages as follows.

1. Initialization stage: the major tasks the encoder has to complete include
   - initializing the buffer size based on the latency requirement
   - subtracting the bit counts for the first I-frame from the total bit counts
   - initializing the buffer occupancy in the middle level.
2. Pre-encoding stage: the tasks of the rate control scheme include
   - estimation of the target bits
   - further adjustment of the target bits based on the buffer status for each VO
   - quantization parameter calculation.
3. Encoding stage: the major tasks the encoder has to complete include
   - encoding the video frame (object) and recording all actual bit-rate
   - activating the macroblock-layer rate control if desired.
4. Post-encoding stage: the encoder needs to complete the following tasks
   - updating the corresponding quadratic rate-distortion model for the entire frame or an individual VO
   - performing the shape threshold control to balance the bit usage between shape information and texture information
   - performing the frame skipping control to prevent the potential buffer overflow or underflow.

Our source encoding rate control achieves more accurate target bit rate allocation under the constraints of low latency and limited buffer size [1, 2]. In addition to the frame-level rate control, the encoding scheme is also applicable to the macro-block-level rate control for finer bit allocation and buffer control, and multiple video objects (VOs) rate control for better video object presentation when feedback indicates more network bandwidth is available.

5 Simulation Results

In this section, we implement our proposed architecture and algorithms on our network simulator and perform a simulation study on MPEG-4 video transport over the network. The purpose of this section is to demonstrate that our architecture and algorithms can (1) transport MPEG-4 video streams over the network with good perceptual picture quality under both low bit rate and varying network conditions, and (2) adapt to available network bandwidth and utilize it efficiently.

We employ a parking lot network configuration (Fig. 3), where path G1 consists of multiple flows and traverse from the first switch (SW1) to the last switch (SW5), path G2 starts from SW2 and terminates at the last switch (SW5), and so forth. Clearly, Link45 is the potential bottleneck link for all flows.

In our simulations, path G1 consists of one MPEG-4 source and five TCP connections while paths G2,

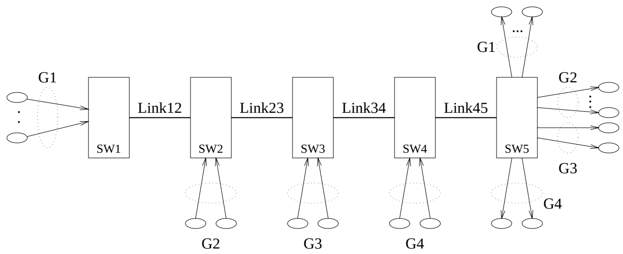

Figure 3: A parking lot network.

Table 1: Simulation parameters.

End System (MPEG-4)	MaxPL		4208 bits
	IR		10 Kbps
	AIR		0.5 Kbps
	PR		200 Kbps
	MR		5 Kbps
	α		0.95
	N_s		79
	N_r		25
	$P_{threshold}$		5%
	Buffer size		1 Mbytes
End System (TCP)	Packet size		576 bytes
	Default timeout		500 ms
	TCP version		Reno
Switch	Buffer size		10 Kbytes
	Packet processing delay		4 μs
	Buffer management		Tail Dropping
Link	End system to switch	Speed	10 Mbps
		Distance	1 km
	Inter-switch	Speed	400 Kbps
		Distance	1000 km

G3, and G4 all consist of five TCP connections, respectively. All the TCP sources are persistent during the whole simulation run.

At the source side, we use the standard raw video sequence "Akiyo" in QCIF format for the MPEG-4 video encoder. The encoder performs MPEG-4 compression and adaptively adjusts its rate under our feedback control algorithm (Algorithm 1). The encoded bit stream is packetized with RTP/UDP/IP protocol overhead and sent to the network. Packets may be dropped due to congestion in the network. For arriving packets, the receiver extracts the packet content to form the bit stream for the MPEG-4 decoder. For a lost packet, the video object plane (VOP) associated with the lost packet will be discarded and a previous VOP will be copied over.

Table 1 lists the parameters used in our simulation. We use 576 bytes for the path MTU. Therefore, the

Figure 4: Video objects VO1 (left) and VO2 (right) in "Akiyo" MPEG-4 video sequence.

maximum payload length, MaxPL, is 526 bytes (576 bytes minus 50 bytes of overhead) [5].

We run our simulation for 450 seconds. Since there is only 300 continuous frames in "Akiyo" sequence available, we repeat the video sequence cyclically during the 450-second simulation run. In the simulations, we identify two VOs as VO1 (background) and VO2 (foreground) (see Fig. 4) and encode them separately.

Figure 5 shows source rate behavior during the 450 second simulation run. We find that the source is able to adjust its rate to keep track of the varying network available bandwidth. Figure 6 shows the link utilization and packet loss ratio during the same simulation run. The oscillation in source rate (Fig. 5) and network utilization (Fig. 6) are due to the propagation delay of the links and the binary nature of our feedback control algorithm. The source performs additive rate increase until it reaches the available link bandwidth. After that the source rate overshoots and results in congestion and packet loss. Packet loss is detected at the receiver and such information is conveyed to the source. Upon receiving such feedback, the source decrease its rate. Despite the oscillations, the bottleneck link (Link45) is 100% utilized. Furthermore, we find that the average packet loss ratio is only 1.6%, which demonstrates the effectiveness of our feedback control algorithm.

Figure 7 shows the PSNR for the Y component of each VOs in the MPEG-4 video sequence at the receiver for the same simulation run. To examine the perceptual quality of the MPEG-4 video, we play out the decoded video sequence at the receiver. We observe that the reconstructed video have good perceptual quality. Figure 8 shows a sample video frame at the receiver.

In summary, based on the simulation results in this section, we conclude that our architecture and algorithms can (1) transport MPEG-4 video streams over the network with good perceptual picture quality un-

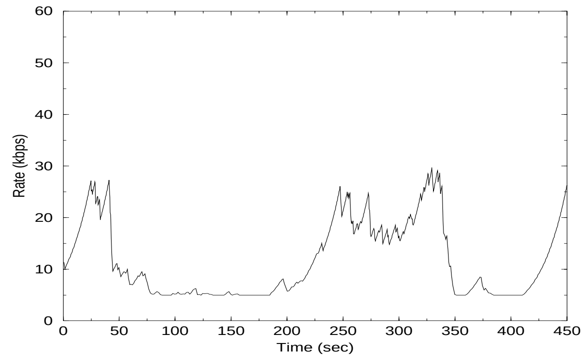

Figure 5: The output rate of the MPEG-4 video source.

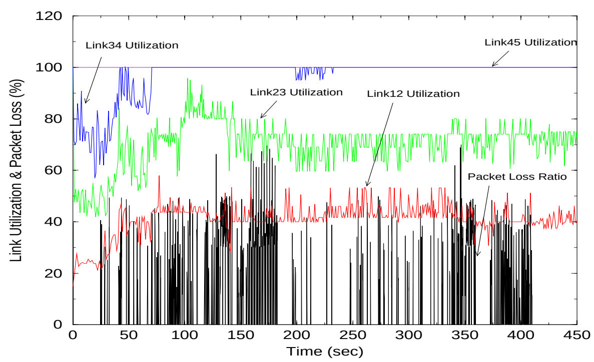

Figure 6: Link utilization and packet loss ratio.

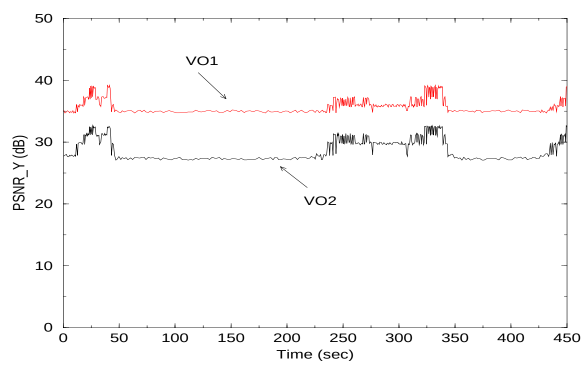

Figure 7: PSNR of each VO of MPEG-4 video at the receiver.

Figure 8: A sample frame during the simulation run.

der both low bit rate and varying network conditions; and (2) adapt to available network bandwidth and utilize it efficiently.

6 Concluding Remarks

This paper presents an end-to-end transport architecture for MPEG-4 video over the Internet. The proposed end-to-end transport architecture includes an end-to-end feedback control algorithm and a source encoding rate control algorithm. Simulation results show that our MPEG-4 transport architecture is capable of transporting MPEG-4 video over the network with good perceptual quality under low bit-rate and varying network conditions and utilize network resource efficiently.

References

[1] 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.

[2] H.-J. Lee, T. Chiang and Y.-Q. Zhang, "Scalable rate control for very low bit rate (VLBR) video," *Proc. IEEE ICIP'97,* pp. 768–771, Oct. 1997.

[3] 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.

[4] 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.

[5] H. Schulzrinne, D. Hoffman, M. Speer, R. Civanlar, A. Basso, V. Balabanian and C. Herpel, "RTP payload format for MPEG-4 elementary streams," *Internet Draft,* Internet Engineering Task Force, March 1998.

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