MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Adaptive Pre-stored Video Streaming with End System Multicast

Huai-Rong Shao, Chia Shen, Hao-Song Kong

TR2002-44 March 2003

Abstract

End system multicast is regarded as a promising architecture for multimedia group communications over the Internet. In this paper we discuss how to integrate adaptation capability from the application and flexible multicast functionality from End system multicast for pre-stored video streaming applications. First we propose a generalized intelligent packetization approach to differentiate information with different priorities within or between bitstreams. Then we describe the enhanced End system multicast scheme to support adaptive applications. We also present a fast and simple packet-level rate adaptation mechanism with simple logical and arithmetic operators for the proposed End system multicast. Experimental results demonstrate that our approach can achieve smooth quality streaming simultaneously to different receivers.

2002 International conference on Signal Processing

ADAPTIVE PRE-STORED VIDEO STREAMING WITH END SYSTEM MULTICAST

Huai-Rong Shao, Chia Shen, and Hao-Song Kong

Mitsubishi Electric Research Laboratories (MERL)
201 Broadway, Cambridge, MA 02139, U.S.A

ABSTRACT

End system multicast is regarded as a promising architecture for multimedia group communications over the Internet. In this paper we discuss how to integrate adaptation capability from the application and flexible multicast functionality from End system multicast for pre-stored video streaming applications. First we propose a generalized intelligent packetization approach to differentiate information with different priorities within or between bitstreams. Then we describe the enhanced End system multicast scheme to support adaptive applications. We also present a fast and simple packet-level rate adaptation mechanism with simple logical and arithmetic operators for the proposed End system multicast. Experimental results demonstrate that our approach can achieve smooth quality streaming simultaneously to different receivers.

1. INTRODUCTION

Multicast is considered as an effective communication support for multi-party applications since it can greatly save network bandwidth. However, more than ten years after its initial proposal by Deering in 1988 [1], IP Multicast is still plagued with concerns pertaining to scalability, network management, deployment and support for higher layer functionality such as error, flow and congestion control. In response to the serious scalability and deployment concerns with IP Multicast over the last decade, some researchers (summarized in [2]) recently proposed an alternate architecture, where all multicast related functionality, including group management and packet replication, is implemented at end systems. Such a kind of overlay multicast architecture is called End System Multicast. Pushing multicast functionality to edge or host of the network, End system multicast brings more flexibility to add higher layer functionalities for multimedia communications.

Rate adaptation in multimedia is regarded as a necessary mechanism to handle network and users' heterogeneity, and fluctuations of available network bandwidth. For video multicast to heterogeneous users, traditionally the sender rate is adapted to the requirement of the worst positioned receiver. The disadvantage of this approach is that the quality of the received video is degraded, except of course for the worst positioned one. This limitation can be overcome by using layered multicast mechanisms [3,4]. McCanne, Jacobson and Vetterli[4] described a receiver-driven layered multicast protocol for rate-adaptive video transmission. In their work, the source transmits each layer of its signal on a separated multicast group. Each receiver specifies its level of subscription by joining a subset of multicast groups of layered video. Li, Amar, and Paul also proposed and evaluated a Layered Video Multicast with Retransmission (LVMR) scheme [4] for distributing video using layered coding over the Internet. Conceptually, they improved the quality of reception within each video layer by re-transmitting lost packets that are given an upper bound on recovery time. However, all above layered approaches depends on IP multicast mechanism. Moreover, the layered approaches usually require multiple network sessions for one video stream. It is complicated for the network and the end-system to control and manage many network sessions for a multimedia application with several video sources, and the synchronization among multiple layers belonging to the same video source is difficult to maintain. In addition, the transmission rate cannot be adjusted in finer granularity than the difference between layers.

In this paper we investigate how to achieve rate adaptation for pre-stored video streaming in the architecture of End system multicast. Chu, Rao, Seshan and Zhang discussed video/audio conferencing support with End system multicast in [2]. But [2] mainly focused on generating and maintaining latency-bandwidth optimized overlay multicast topology at the network side. It mentioned little on adaptation at the application side. In this paper we discuss how to take advantage of both application adaptation and overlay multicast adaptation to obtain optimal end-to-end performance. A fast and simple packet-level rate adaptation mechanism was proposed to overcome the drawbacks of the layered approach. With this packet-level mechanism, only one network session is required for multiple layers of the same video and fine rate adaptation at intermediate nodes can be done easily combined with End system multicast. We also discuss how to extend the work of End system multicast in [2] to support application adaptation well.

2. BITSTREAM PACKETIZATION AT SENDER

All sender, network and receivers need to contribute to rate adaptation. One main principle of our design is to keep network rate control simple and easy to implement. If we can map the complicated semantic application adaptation information to network packet priorities, we will keep the network rate control mechanism as simple as just dropping low priority packets. So how to map the information at the compressed bitstream into different priorities packets is one key problem.

Recently several kinds of scalable coding such as layered coding, FGS (Fine Granularity Coding) and Wavelet Coding were proposed [5]. Scalable coding improves the rate adaptation capability for streaming video. However, it is not natural to map the scalable bitstream to network packets because some different priorities information are interleaved in the bitstream. On the other side, even for non-scalable coding video streams, we still can distinguish some relative priorities, for example, MPEG-1 format I frame is more important than P frame and motion vector information is more important than motion compensation information for the same P frame. Unfortunately, different types of information are also interleaved together in non-scalable coding as well as scalable coding. This leads to our bitstream re-organization before packetization for compressed video streams for fine granularity transmission rate control.

In this section we use object-based MPEG4 video as an instance to demonstrate our packetization approach though it is general enough to most compressed bitstreams. It can be seen that this packetization scheme can improve the error resilience capability and flexibility of bit rate control.

2.1 Bitstream classification

After compression, encoded video data is placed in the bitstream according to the temporal and spatial position of its content, i.e., frame by frame, macroblock by macroblock, and block by block. Information within the compressed bitstream can be divided into several semantic types such as control header, shape, motion, and texture information. These different types of information are interleaved together, although intrinsically they have different importance levels for decoding and transmission. For example, shape and motion information is more important than texture for a P frame in MPEG4. If the shape and motion information is lost during transmission, the decoder cannot reconstruct the P frame successfully. However, if partial texture information is lost without the loss of shape and motion, it is still possible to reconstruct the P frame with acceptable quality. Existing Internet video streaming and networking schemes usually don't consider this kind of differentiation within the bitstream, or just distinguish different types of frames (I, P or B) or different layers (base layer and enhancement layers). When network congestion occurs, packets are discarded with no distinction in most previous works on video transmission. Some important information may be dropped together with some less important information and the decoder cannot produce proper video sequences.

The information in the bistream of object-based video coding such as MPEG4 can be classified into the following categories without considering the enhancement layers:

Control information, such as Video Object Header, Video Object Layer Header, and Video Object Plane Header.
Shape information of I frame.
Texture DC information of I frame.
Texture AC information of I frame.
Shape information of P frame.
Motion information of P frame.
Texture information of P frame.
Shape information of B frame.
Motion information of B frame.
Texture information of B frame.

2.2 Prioritization and packetization

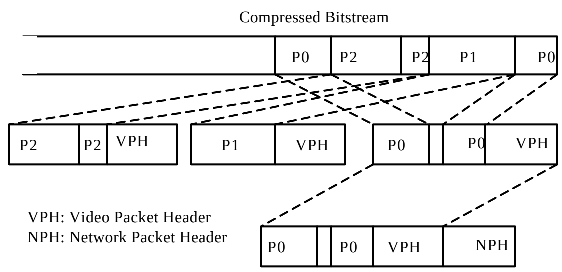

Figure 1: Intelligent packetization.

Different types of information can be assigned to different importance levels. Besides information type, the importance level of the information is decided by the following two factors: the distortion reduction of the information and the dependency relationship between different parts of information. When transmitted, information with the same priority is aggregated and different priorities of information are packetized into different classes of packets (Figure 1). For example, we classify the single layer or base layer of multi-layer compressed information into the following classes:

Priority0 Class:
Control information;
Shape information of I frame (base layer);
Texture DC information of I frame (base layer).
Priority1 Class: Texture AC information of I frame (base layer).
Priority2 Class:
Shape information of P frame (base layer);
Motion information of P frame(base layer).
Priority3 Class: Texture information of P frame (base layer);
Priority4 Class:
Shape information of B frame (base layer);
Motion information of B frame (base layer);
Texture information of B frame (base layer).

In the case that network resources cannot satisfy the rate requirement of a video object flow, the packets with lower priorities will be discarded by the sender or intermediate nodes earlier than those with higher priorities if needed. In addition, different error control mechanisms can be implemented on different priority packets to enhance the error resilience capability. To maintain compliance with the MPEG4 syntax, we use an index table for each video object rather than define a new syntax in the video bitstream. The index table includes several items such as index number, information category, priority level, starting position (relative), and length. This table is used to index different types of information in the compressed bitstream and is generated as a individual file along with the compressed bitstream when encoding a video object. So the index table actually is a virtual table which acts only as a reference for extracting different parts of information and does not constitute part of the bitstream.

We define a new application level packet (ALP) format for object-based video. The ALP size is limited to no larger than the MTU of the network. We also avoid using small ALPs in order to achieve high transmission efficiency. Several parts of the information with the same priority will be multiplexed at the ALP level, though probably they are from different VPs. However, multiplexing will arise a new problem. If one packet that contains a lot of consecutive motion information is lost, severe quality degradation may be brought to video playback. Hence, we use two methods to reduce this kind of impairment. One is to limit the number of video packets in an ALP. The other is to place video packets interleavly. For example, shape and motion information of video packet with 0, 2, 4 are placed into ALP0, and information of video packet 1,3,5 are placed into ALP1, respectively. Our rule for multiplexing of the shape and motion information or texture information of VPs into one ALP is that we interleave until one of the following constraints is met: number of VPs in current ALP reaches a certain threshold; the size of one ALP reaches a certain maximum size; or a different vop-coding-type is found.

3. ENHANCED END SYSTEM MULTICAST FOR RATE ADAPTATION

Our work at the network side is based on the construction of bandwidth-delay optimized overlay spanning tree for data delivery in [2]. Application adaptation factor is added into the End system multicast in our approach.

3.1 Construction of the Spanning tree

Two basic methods have emerged for the construction of overlay spanning trees. One approach is to construct the tree directly -that is, members explicitly select their parents from among the members they know. An alternate approach is to construct trees in a two-step process. First they construct a richer connected graph termed a mesh. Second, they construct (reverse) shortest path spanning trees of the mesh, each tree rooted at the corresponding source using well-known routing algorithms. While in the extreme case, a mesh could consist of all possible N *(N -1) overlay links in a group consisting of N members, typically protocols try to keep the meshes sparse in order to keep the overhead of routing low. Given that the final spanning trees are constructed from among the overlay links present in the mesh, it becomes important to construct a good quality mesh in the first place.

Based on Narada [6], the authors in [2] choose multiple routing metrics in the distance vector protocol running on the mesh -the available bandwidth and the latency of the overlay link. The routing protocol uses a variant of the shortest widest path algorithm presented in [7]. Every member tries to pick the widest (highest and width) path to every other member. If there are multiple paths with the same bandwidth, the member picks the shortest (lowest latency) path among all these.

To improve the stability of the tree, exponential smoothing is used for the raw estimates of latency and available bandwidth on overlay links. And also the discrete bandwidth levels are defined to reduce the instability.

We further introduce the application adaptation to handle the instability problem. When overlay links change dynamically, the spanning tree will not change until the available bandwidth cannot satisfy the minimal requirement from the application.

3.2 Dynamic packet-level rate control

Authors in [2] adopt the combination of active probing and passive monitoring to obtain the status of overlay links in order to adjust spanning tree dynamically. This information can also used by us to dynamically adapt the transmission rate to available bandwidth at intermediate nodes.

Our scheme supports adaptive rate control by discarding some packets with lower priorities at the sender or/and intermediate nodes according to the network available bandwidth. We assume there are N priority levels, $Pi\,(0 \le i < N)$, for a video object, and each level has the original bit rate $r_i(0 \le i < N)$, and the original rate of the video flow is R. Obviously, we have

$$R = \sum_{i=0}^{N-1} r_i \quad . \tag{1}$$

During transmission, if the network congestion occurs or the receivers require lower bit rates through some user-interactivity behaviors, the bit rate of the video object needs to be reduced to R'. In order to do that, we first need to find the index k ($0 \le k < N$) satisfying

$$\sum_{i=0}^{k-1} r_i \le R' < \sum_{i=0}^{k} r_i \quad . \tag{2}$$

All packets with priority Pj (k<j<N) will be discarded at the sender level. Then, if

$$R^{'} = \sum_{i=0}^{k-1} r_i \, , \tag{3}$$

all packets with priority Pk will also be discarded at the sender level. Otherwise, some fine bit rate adjustment will be implemented, meaning some packets with priority P_k will be selectively discarded at the sender and/or the intermediate nodes. For example, if some B frames need to be dropped, we can employ the scheme proposed in [10] to discard B frames selectively. For another example, if texture information within the P frame needs to be partially discarded, we will adopt the FGS scheme and discard A% texture information in P frame wherein

$$A = \frac{\sum_{i=0}^{k} r_i - R^{'}}{r_k} \times 100\% \tag{4}$$

It can be seen that our rate control scheme supports both temporal and quality scalability.

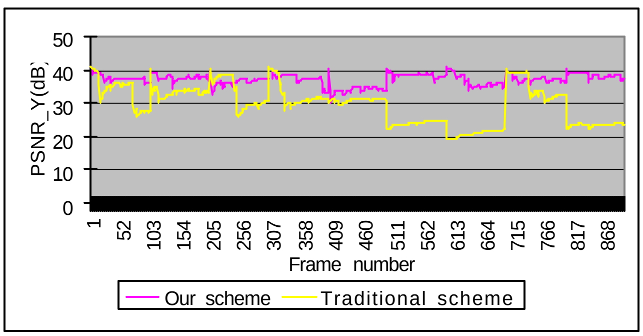

Figure 2: Quality comparisons (bit rate=240kbps).

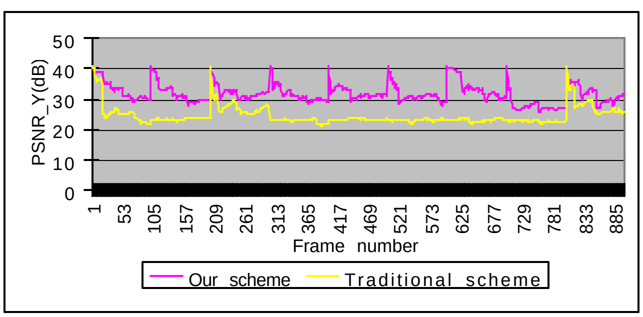

Figure 3: Quality comparisons (rate=180kbps).

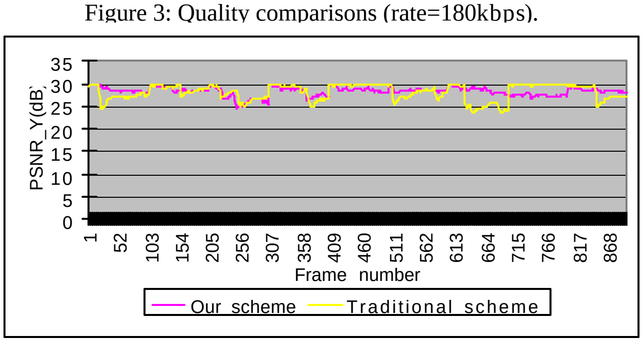

Figure 4: Quality comparisons (rate=56kbps).

4. PERFORMANCE EVALUATION

We did several simulation experiments to evaluate the network performance for our enhanced End system multicast scheme and obtained similar results with those in [2]. We also did experiments to evaluate the performance of adaptive video streaming applications within the End system multicast architecture. We implemented a simple MPEG4 video streaming system and run it on a NS2 simulated End system multicast environment. Our streaming system is based on the Microsoft's MPEG4 video encoding/decoding source codes and Microsoft research's IPv6 protocol stack source codes. Besides our scalable transmission approach, for the sake of comparison we also implemented the traditional approach in which the bitstream is packetized with no information re-organization/prioritization and all packets have a fixed size (600bytes). Figure 2~4 demonstrate the video quality curves of Akiyo at three receivers for both our approach and traditional approach. It can be see that our approach achieves relatively smooth video quality for different receivers with different bandwidth capacities. And also our packetization scheme can achieve better video quality than traditional approaches.

5. CONCLUSIONS AND FUTURE WORK

In this paper we present how to integrate adaptation capability from the application and flexible multicast functionality from End system multicast for pre-stored video streaming applications. Because End system multicast is more appropriate for small group applications, our approach proposed also aims to applications with video streaming to small group receivers. Many collaborative applications need such streaming functionality. For example, some people in a remote video conference procedure may want to watch a clip of news movie simultaneously during their discussion. Our future work includes how to support video streaming to hundreds of thousands users such as Internet TV.

6. REFERENCES

[1]S. Deering, "Multicast Routing in Internetworks and Extended Lans", In Proceedings of ACM SIGCOMM 1988.

[2]Y. Chu, S. Rao, S. Seshan and H. Zhang, "Enabling Conferencing Applications on the Internet using an Overlay Multicast Architecture", In proceedings of Sigcomm, Sep. 2001.

[3]S. McCanne, "Receiver-driven layered multicast", SIGCOMM Symposium on Communications Architectures and Protocols, Palo Alto, California, Aug. 1996.

[4]X. Li, M.H. Ammar, and S. Paul, "Layered video multicast with retransmission (LVMR): Evaluation of error recovery", Proceedings of NOSSDAV, May 1997.

[5]D. Wu, T. Hou, Y. Zhang, " Scalable video coding and transport over broadband wireless networks", Proceedings of the IEEE, Vol. 89, Issue 1, Jan. 2001.
[6]Y. Chu, S. Rao,and H. Zhang, "A Case for End System Multicast", In Proceedings of ACM Sigmetrics, June 2000.
[7]Z. Wang and J. Crowcroft, "Bandwidth-delay based routing algorithms", In IEEE GlobeCom , November 1995.