A Platform for Real-Time Content Adaptive Video Transmission Over Heterogeneous Networks

Hao-Song Kong, Anthony Vetro, Hari Kalva, Dongdong Fu, Ximin Zhang, Jianlin Guo and Huifang Sun

Mitsubishi Electric Research Laboratories, Murray Hill, NJ, USA

ABSTRACT

In this paper, we present a real-time adaptive streaming video platform. This platform is fully compliant with the Internet Streaming Media Alliance Implementation Specification. It has been used for experiments of Real-time video streaming and transcoding via unicast and multicast over heterogeneous networks. One of the examples of streaming video over a lossy channel is given, and a simple and efficient scheme for the packet loss recovery is presented.

Keywords: Video streaming, ISMA, RTP, RTSP, SDP, Packet loss recovery, QoS

1. INTRODUCTION

In the recent years, there has been an explosive growth in the development and deployment of distributed applications that transmit and receive multimedia content over the Internet. However, the real multimedia applications, such as visual communications (audio and video) over the Internet have not been widely and successfully used in our daily life, because the current best-effort Internet does not offer any quality of service (QoS) guarantees. The service requirements of the multimedia applications differ significantly from those of traditional data-oriented applications, like HTTP based streaming, email and FTP etc. Multimedia applications are highly sensitive to end-to-end delay and delay variation. In addition, for multicast, it is difficult to efficiently support multicast media streaming while providing service flexibility to meet a wide range of QoS requirements from different users. Therefore, multimedia streaming over the Internet is very challenging. Apple, Microsoft and RealNetworks have proposed some solutions for multimedia streaming. Unfortunately, those solutions use different transport schemes and media types, incompatible to each other. The recently released Internet Streaming Media Alliance Implementation Specification (ISMA) [1] offers an open standard for media streaming with the goal to use the existing open standards that people can follow to build interoperable audio and video systems for use on IP networks and the Internet.

In this paper, we briefly introduce the ISMA specification and its requirements in section 2. Based on the ISMA document, we propose a framework for real-time streaming video over heterogeneous networks and describe the system architecture in section 3. In section 4, we give an example of video streaming over a lossy channel. We present a simple and efficient scheme for packet loss recovery and show an experimental result. Finally, we conclude and discuss future work in section 5.

2. ISMA IMPLEMENTATION SPECIFICATION VERSION 1.0

The fulfillment of Internet media streaming has been delayed due to a number of technical issues. One significant impediment has been the lack of a set of widely adopted technical standards for the transmission of audio/video on the Internet. In order to meet the development needs and to achieve rapid deployment for the Internet media streaming, the Internet Streaming Media Alliance (ISMA) has proposed the implementation specification version 1.0. This specification emphasizes to use existing open standards for the development of media streaming system. It has three criteria for selecting the technologies in this implementation specification.

The first criterion is that the technology must be an open standard. The specifications must be available to people who wish to participate.

The second criterion is that the technology should be interoperable. It also argues that the profile should aim for simplicity; seeking a solution that addresses the core needs of the Internet audio/video applications, and not the specialized needs of every conceivable application.

The third criterion is that the technology be forward-looking, and provide for new network media technologies and emerging video-enabled information appliances.

Multimedia Systems and Applications V, Andrew G. Tescher, Bhaskaran Vasudev, V. Michael Bove, Jr., Ajay Divakaran, Editors, Proceedings of SPIE Vol. 4861 (2002) © 2002 SPIE · 0277-786X/02/$15.00

The architecture of ISMA, at its simplest, consists of a media server, an IP network, and a media client. This simple model can be easily extended to allow for a number of intermediary systems in the transmission process that perform a range of services. Examples of the services provided by intermediary systems are: short-term or long-term storage, transcoding to new bit rates, mixing with other media streams.
The specification describes some basic functions for the Internet media streaming system as follows:

- Media Transmission
  - The media content is transmitted over the network either in real-time or off-line.
- Media Control
  - Users request transmission of media content for rendering and optionally control the transmission (e.g. pause, rewind, fast-forward etc.).
- Media Announcement
  - Either human or computer discovers the existence of available media content and the information necessary to request access to the media content.

In the current specification, there are two profiles defined. Profile 0 is aiming for audio/video at low bit rate suitable for narrow bandwidth and mobile wireless infrastructures. Profile 1 is targeting at medium bit rate to allow for a richer streaming experience over infrastructures with broad bandwidth. This specification initially focuses on MPEG-4 technologies, future adaptations and revisions may include MPEG-2, MPEG-7 and other non-MPEG technologies. The following lists the requirements common to all profiles.

- Media Decoding
  - either video or audio, or both
- Transports
  - RTP [2]: IETF RFC 1889
  - RTP [3] profile: IETF RFC 1890
  - UDP: IETF REC 768
- RTP Payloads [4]
  - IETF RFC 3016 RTP Payload Format for MPEG-4 Audio/Visual Streams
- Content Distribution
  - MPEG-4 MP4 Format – ISO/IEC 14496-1:2000(E)
- Media Control
  - RTSP [5]: IETF RFC 2326
- Media Announcement
  - SDP [6]: IETF RFC 2327

The requirements for profile 0 and profile 1 are:
Profile 0

- Video
  - MPEG-4 ISO/IEC 14496-2:1999 + Cor 1:2000 + Cor 2:2001
  - Simple profile @ level 1
  - Visual session size is QCIF (176x144)
  - Maximum bit rate is 64kbit/s
- Audio
  - MPEG-4 ISO/IEC 14496-3:1999 and AMD1 2000
  - High quality audio profile @ level 2
  - Up to 2 channels
  - Up to 48000 Hz sampling rate

Profile 1

- Video
  - MPEG-4 ISO/IEC 14496-2:1999 + Cor 1:2000 + Cor 2:2001
  - Advanced simple profile @ level 3
  - Visual session size is CIF (352x288)
  - Maximum bit rate is 1.5 Mbps
- Audio
  - MPEG-4 ISO/IEC 14496-3:1999 and AMD1 2000
  - High quality audio profile @ level 2

- Up to 2 channels
- Up to 48000 Hz sampling rate

2.1 Real-time Transport Protocol (RTP)

RTP provides end-to-end network transport functions suitable for real-time applications, such as audio and video, over multicast or unicast network services. However, it does not address resource reservation and does not guarantee QoS for real-time services. It relies on upper layer to provide on-time delivery. Since it typically runs on top of UDP, it suffers packet loss and out of order. Therefore, RTP provides functions including payload type identification, sequence number and time stamp to allow receiver to detect lost packets and to reorder received packets. The payload type identifier in the RTP packet is used to indicate the type of encoding. It allows the sender to change the encoding in the middle of a session to increase the quality or to decrease the stream bit rate.
RTP Control Protocol (RTCP) is a control protocol designed to work with RTP. It is based on periodic transmission of control packets from all participants of a session to all other participants using the same distribution mechanism as the RTP data packets. Its major function is to provide feedback on the quality of the data distribution. The feedback information can be used for adaptive encoding or for diagnostic purpose in the transmission.

2.2 Real-Time Streaming Protocol (RTSP)

RTSP is a session control protocol for streaming media over the Internet. It is responsible for establishing and controlling media streams between media servers and clients. Clients can request a presentation description file from a server, and ask the server to setup a session to send the requested media stream. RTSP also provides VCR-like control functions to allow clients to pause/resume, fast forward and rewind media streams in the session.

2.3 Session Description Protocol (SDP)

SDP is purely a format for session description. It is intended to describe the properties of multimedia sessions for the purposes of session announcement and session initiation. SDP itself does not incorporate any transport protocols. It has to rely on other protocols, such as Session Announcement Protocol (SAP), Session Initiation Protocol (SIP), RTSP etc. for distributing session descriptions. SDP usually includes the following information:

- Session name and purpose
- Expected duration of the session
- Payload types offered in the session
- Information needed to receive and decode the media data (addresses, ports, formats etc.)
- Bandwidth estimations to facilitate resource reservations, contact information for the initiator or session manager

3. VIDEO TRANSMISSION SYSTEM ARCHITECTURE

Based on the ISMA implementation specification and its requirements, we developed a real-time video transmission platform over heterogeneous networks. The system architecture is shown in Figure 1. It consists of a server and a number of clients.

3.1 Server

The server is composed of six modules:

- **Live video capturing module** The digital video camera captures live video, encodes it into MPEG-2 transport stream and then sends it to the server via a 1394 cable.
- **Video storage** It stores MPEG-1 ES, MPEG-2 ES/TS, MPEG-4 ES and MP4 files.
- **MPEG-4 video transcoder** It converts MPEG-1 and MPEG-2 ES to MPEG-4 ES. Due to the network resource variations, it is often needed to adapt the bit rate to an available bandwidth over heterogeneous networks. A typical transcoding process involves decompression and reencoding operations, its computational complexity is very critical in real-time video streaming. The transcoder integrated in our platform can achieve real-time transcoding by employing low-complexity compressed domain transcoding [7].

- **Packetizer** The packetizer implemented in the platform is fully compliant with RFC 3016. It packetizes MPEG-4 elementary video stream and embeds it into RTP packets.
- **SDP and RTSP control module** Clients and the server initiate their session connections through SDP. Clients use RTSP to control the media stream delivery with real-time properties.
- **RTP/UDP module** It does RTP packets to UDP datagrams mapping. RTP packets are encapsulated within UDP datagrams. This module fulfills a high throughput and an efficient bandwidth usage.

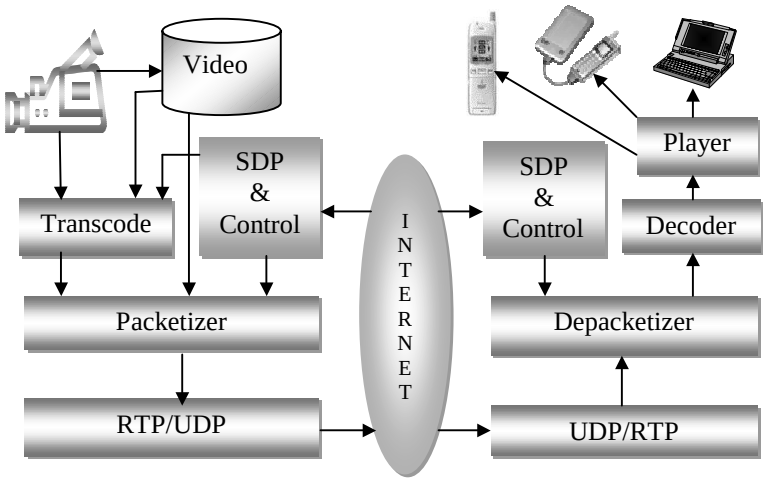

Figure 1: Video transmission system architecture

3.2 Client

The client is composed of five modules. UDP/RTP, Depacketizer, SDP and RTSP control modules are just counterparts of corresponding modules in the server. The client is based on a multithreaded architecture that receives the RTP packet, depacketizes it and delivers it to the decoder.

- **Decoder** This is a robust MPEG-4 decoder. It decodes the depacketized packet into YUV values and passes the YUV frame to the player.
- **Player** The player was implemented on Win32 and WinCE platform with a plugin architecture and supports multiple decoders to handle different media types.

The video transmission system operates as follows. Clients first launch a player window to send a SDP message to the server for initiating a TCP session. The SDP message includes the desired video stream, its URL address, bit rate and resolution etc. When the server receives the SDP message, it retrieves the video and opens a RTP/UDP session with clients. When the session is setup, the server starts to send the requested video to the clients via unicast or multicast. The server has a multithread architecture. One thread is dedicated to listening to the client requests. Other threads are used for RTP data transmission and RTSP control. According to client request or according to RTCP feedback information, the server will adapt its transmission through its transcoder to guarantee the best quality on the client side. The client has two buffers. One is used for receiving packets. Since UDP datagrams may arrive out of order, this buffer is used to reorder the packets in terms of sequence number in the RTP header. Another buffer is used for smooth playback. Due to network environment change and congestion, jitter may occur in the video transmission. Buffering a sufficient amount of packets for a period of time, the smooth playback can be achieved, although it causes some delays.

This platform has been used in different network environments. For Intranet video streaming via unicast or multicast through both wired and wireless (IEEE 802.11b) connections, due to the sufficient network resources, there is no traffic congestion, no packet loss. The video quality is very good. For Internet streaming, we ran our server outside of the firewall in the public domain, we had the same results as the Intranet situation. We did not suffer the congestion and packet loss. This may be the case of a few hops between the server and our clients. We also tried using mobile phone to dial up to the Internet and via our

VPN link to the Internet, we experienced very severe packet loss problem because of the bottleneck at the low bit rate connection. In the next section, we will present a strategy to deal with the packet loss problem.

4. PACKET LOSS RECOVERY

The Internet is a "best effort" network, there is no guarantee that packets will arrive at their destination or will be received in their sending order. When there is heavy network traffic or the delivery of streaming media is over lossy and heterogeneous channels, a random or a burst packet loss will occur. Packet loss can be detrimental to compressed video with inter-dependent frames because errors potentially propagate across many frames. The packet loss problem has been addressed by many researchers and companies. Common solutions are automatic repeat-request (ARQ) mechanism and interleaving technique. The ARQ mechanism allows the client to request the server to retransmit the lost packets. If they are successfully delivered within the available time, the loss is recovered. However, in the practical applications, the latency requirements often do not permit retransmission of all lost packets. In [8], authors proposed a selective retransmission method to selectively retransmit the most important data in the bitstream. The interleaving technique is used to minimize the perceptual damage caused by the packet loss. Since it distributes one packet loss error to several packets, it results in a much less noticeable type of disruption. Due to the large sizes of video frames, simple interleaving technique is not effective for the packet loss.

In this paper, we propose a simple and efficient scheme for packet loss recovery. Our method is to selectively add some redundant packets to the video stream. In the MPEG-4 bitstream, I frames are more important than P frames, and P frames are more important than B frames since P frames are coded using one directional motion-compensated prediction from previous I or P frame, B frames are coded using predictions from either past or future I or P frames. In the packetization, each frame is fragmented to several packets. According to the fragmentation rule of MPEG-4 video stream in RFC 3016, the VOP header is contained in the first packet of the frame as shown in Figure 2. Since this packet contains information for reconstructing the frame, it is more important than other packets.

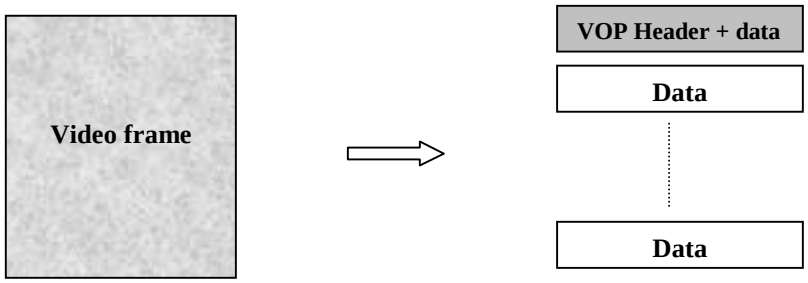

Figure 2: A video frame is fragmented to several packets; the first packet contains VOP header

We now describe the packet loss scenarios during the transmission.

- If packets in I/P frame get lost, it not only degrades its own picture quality, but also causes error propagation to other frames.
  - If the first packet gets lost; its whole picture is damaged, and the subsequent P or B frames have severe degradation
  - If other packets get lost, its own picture and subsequent dependant pictures are only corrupted by the lost packets
- If packets in B frame get lost, it degrades its own frame quality, does not cause error propagation to other frames.
  - If the first packet gets loss; the whole picture will be damaged
  - Otherwise the picture is corrupted by the lost packet

In terms of above analysis, it can be seen that if we want to ensure the video quality on the client side, we can protect some important packets by repeat sending them to the client to compensate packets loss. In order to reduce overhead, these redundant packets are chosen based on the frame type and their position in the frame. We chose the first packet from each I frame and some P frames as redundant packets due to their important features mentioned above. If there is no packet loss, the client simply discards the redundant packets. When packet loss occurs, especially when some VOP header packet gets lost, the client will use these redundant packets to reconstruct the corresponding frames as shown in Figure 3. These

redundant packets not only help with recovering the missing VOP header frame, but also help with the reconstruction of subsequent frames. Without the redundant packets, the received packets become useless because the client can not reconstruct them with no header information. An example of packet loss is shown below. The first packet which contains the VOP header and the third packets in a frame are lost in the transmission. The reconstructed frame without receiving the redundant packet is totally damaged. However, with receiving the redundant packets, we can see that the frame is almost reconstructed as shown in Figure 4. The corrupted strip in the picture is caused by the third packet loss which we will deal with it in the future work. In our packet loss recovery scheme, we use a small amount of redundant packets to trade off the packet loss and ensure the video quality.

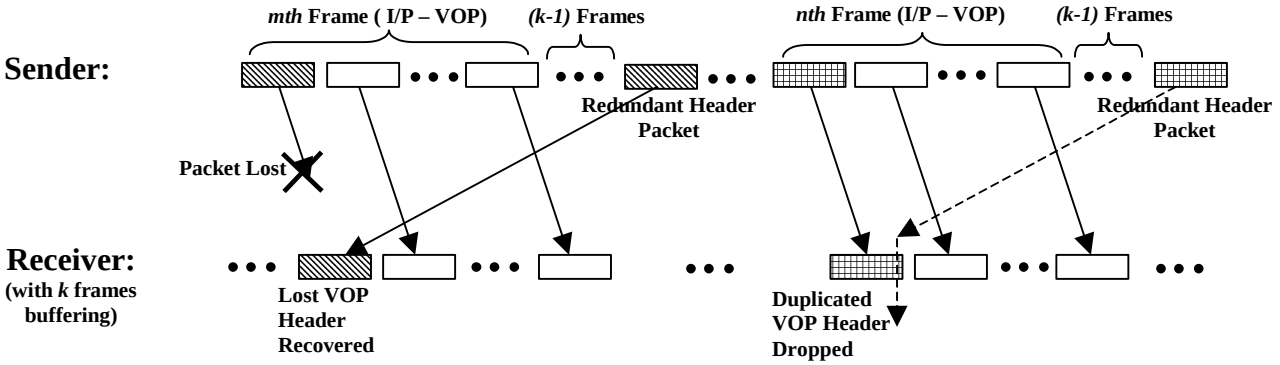

Figure 3: Packets loss recovery scheme illustration

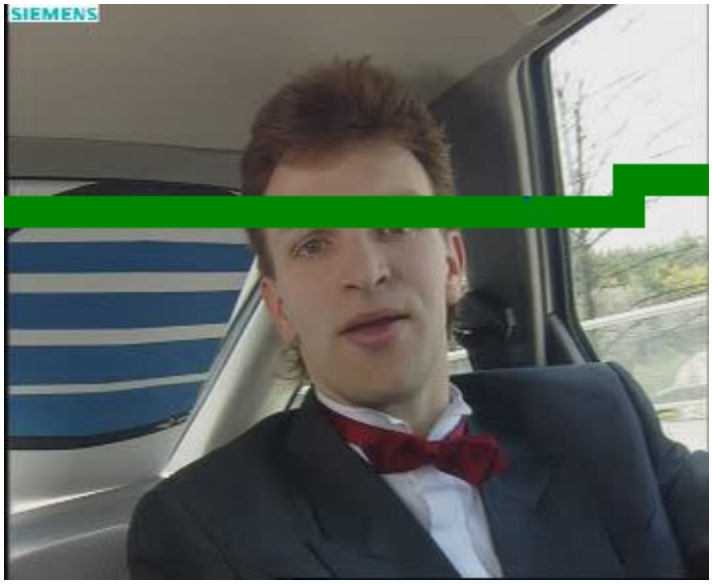

Figure 4: reconstructed frame with the third packet loss

5. CONCLUSION AND FUTURE WORK

In this paper, we introduced the Internet Streaming Media Alliance Implementation Specification and its requirements. Based on the ISMA specification, we described our platform that we have developed for real-time adaptive video streaming and transcoding over heterogeneous networks. The platform works well over our intranet and the Internet for delivering high quality video in real-time. We also presented a simple and efficient scheme for packet loss recovery. Currently, this scheme is

only protecting the packet that contains the VOP header. In the future, we will further improve this scheme so that it can deal with general packet loss and perform error concealment.

6. REFERENCES

[1] Internet Streaming Media Alliance Implementation Specification Version 1.0, 28 August 2001
[2] H. Schulzrinne etc., "RTP: A transport protocol for real-time applications", RFC 1889, Jan 1996.
[3] H.Schulzrinne, "RTP Profile for Audio and Video Conferences with Minimal Control", RFC 1890, Jan 1996.
[4] Y. Kikuchi etc., "RTP Payload Format for MPEG-4 Audio/Visual Streams", RFC 3016, November 2000.
[5] H. Schulzrinne etc., "Real-Time Streaming Protocol", RFC 2326, April 1998.
[6] M. Handley etc., "SDP: Session Description Protocol", RFC 2327, April 1998.
[7] A. Vetro, H. Sun and Y. Wang, "Object-based transcoding for adaptable video content delivery," IEEE Trans. Circuits and Systems for Video Technology, vol. 11, no. 3, March 2001.
[8] N. Feamster and H. Balakrishnan, "Packet Loss Recovery for Streaming Video", *12th International Packet Video Workshop* , Pittsburgh, PA, April 2002.