ELSEVIER

Signal Processing: *Image Communication* 15 (1999) 95–126

SIGNAL PROCESSING: IMAGE COMMUNICATION

www.elsevier.nl/locate/image

Scalable Internet video using MPEG-4

Hayder Radha*, Yingwei Chen, Kavitha Parthasarathy, Robert Cohen

Philips Research, 345 Scarborough Rd, Briarcliff Manor, New York, 10510, USA

Abstract

Real-time streaming of audio-visual content over Internet Protocol (IP) based networks has enabled a wide range of multimedia applications. An Internet streaming solution has to provide real-time delivery and presentation of a continuous media content while compensating for the lack of Quality-of-Service (QoS) guarantees over the Internet. Due to the variation and unpredictability of bandwidth and other performance parameters (e.g. packet loss rate) over IP networks, in general, most of the proposed streaming solutions are based on some type of a data loss handling method and a layered video coding scheme. In this paper, we describe a real-time streaming solution suitable for non-delay-sensitive video applications such as video-on-demand and live TV viewing.

The main aspects of our proposed streaming solution are:

1. An MPEG-4 based scalable video coding method using both a prediction-based base layer and a fine-granular enhancement layer;
2. An integrated transport-decoder buffer model with priority re-transmission for the recovery of lost packets, and continuous decoding and presentation of video.

In addition to describing the above two aspects of our system, we also give an overview of a recent activity within MPEG-4 video on the development of a fine-granular-scalability coding tool for streaming applications. Results for the performance of our scalable video coding scheme and the re-transmission mechanism are also presented. The latter results are based on actual testing conducted over Internet sessions used for streaming MPEG-4 video in real-time. © Published by 1999 Elsevier Science B.V. All rights reserved.

1. Introduction

Real-time streaming of multimedia content over Internet Protocol (IP) networks has evolved as one of the major technology areas in recent years. A wide range of interactive and non-interactive multimedia Internet applications, such as news on-demand, live TV viewing, and video conferencing rely on end-to-end streaming solutions. In general, streaming solutions are required to maintain real-time delivery and presentation of the multimedia content while compensating for the lack of Quality-of-Service (QoS) guarantees over IP networks. Therefore, any Internet streaming system has to take into consideration key network performance parameters such as bandwidth, end-to-end delay, delay variation, and packet loss rate.

To compensate for the unpredictability and variability in bandwidth between the sender and receiver(s) over the Internet and Intranet networks, many streaming solutions have resorted to variations of layered (or scalable) video coding methods (see for example [22,24,25]). These

* Corresponding author.
E-mail address: hmr@philabs.research.philips.com (H. Radha)

solutions are typically complemented by packet loss recovery [22] and/or error resilience mechanisms [25] to compensate for the relatively high packet-loss rate usually encountered over the Internet [2,30,32,33,35,47].

Most of the references cited above and the majority of related modeling and analytical research studies published in the literature have focused on delay-sensitive (point-to-multipoint or multipoint-to-multipoint) applications such as video conferencing over the Internet Multicast Backbone – MBone. When compared with other types of applications (e.g. entertainment over the Web), these delay-sensitive applications impose different kind of constraints, such as low encoder complexity and very low end-to-end delay. Meanwhile, entertainment-oriented Internet applications such as news and sports on-demand, movie previews and even 'live' TV viewing represent a major (and growing) element of the real-time multimedia experience over the global Internet [9].

Moreover, many of the proposed streaming solutions are based on either proprietary or video coding standards that were developed at times prior to the phenomenal growth of the Internet. However, under the audio, visual, and system activities of the ISO MPEG-4 work, many aspects of the Internet have being taken into consideration when developing the different parts of the standard. In particular, a recent activity in MPEG-4 video has focused on the development of a scalable compression tool for streaming over IP networks [4,5].

In this paper, we describe a real-time streaming system suitable for non-delay-sensitive[1] video applications (e.g. video-on-demand and live TV viewing) based on the MPEG-4 video-coding standard. The main aspects of our real-time streaming system are:

1. A layered video coding method using both a prediction-based base layer and a fine-granular enhancement layer: This solution follows the recent development in the MPEG-4 video group for the standardization of a scalable video compression tool for Internet streaming applications [3,4,6].
2. An integrated transport-decoder buffer model with a re-transmission based scheme for the recovery of lost packets, and continuous decoding and presentation of video.

The remainder of this paper is organized as follows. In Section 2 we provide an overview of key design issues one needs to consider for real-time, non-delay-sensitive IP streaming solutions. We will also highlight how our proposed approach addresses these issues. Section 3 describes our real-time streaming system and its high level architecture. Section 4 details the MPEG-4 based scalable video coding scheme used by the system, and provides an overview of the MPEG-4 activity on fine-granular-scalability. Simulation results for our scalable video compression solution are also presented in Section 4. In Section 5, we introduce the integrated transport layer-video decoder buffer model with re-transmission. We also evaluate the effectiveness of the re-transmission scheme based on actual tests conducted over the Internet involving real-time streaming of MPEG-4 video.

2. Design considerations for real-time streaming

The following are some high-level issues that should be considered when designing a real-time streaming system for entertainment oriented applications.

2.1. System scalability

The wide range of variation in effective bandwidth and other network performance characteristics over the Internet [33,47] makes it necessary to pursue a scalable streaming solution. The variation in QoS measures (e.g. effective bandwidth) is not only present across the different access technologies to the Internet (e.g. analog modem, ISDN, cable modem, LAN, etc.), but it can even be observed over relatively short periods of time over a particular session [8,33]. For example, a recent study shows that the effective bandwidth

[1] Delay sensitive applications are normally constrained by an end-to-end delay of about 300–500 ms. Real-time, non-delay-sensitive applications can typically tolerate a delay on the order of few seconds.

of a cable modem access link to the Internet may vary between 100 kbps to 1 Mbps [8]. Therefore, any video-coding method and associated streaming solution has to take into consideration this wide range of performance characteristics over IP networks.

2.2. Video compression complexity, scalability, and coding efficiency

The video content used for on-demand applications is typically compressed off-line and stored for later viewing through unicast IP sessions. This observation has two implications. First, the complexity of the video encoder is not as major an issue as in the case with interactive multipoint-to-multipoint or even point-to-point applications (e.g. video conferencing and video telephony) where compression has to be supported by every terminal. Second, since the content is not being compressed in real-time, the encoder cannot employ a variable-bit-rate (VBR) method to adapt to the available bandwidth. This emphasizes the need for coding the material using a scalable approach. In addition, for multicast or unicast applications involving a large number of point-to-multipoint sessions, only one encoder (or possibly very few encoders for simulcast) is (are) usually used. This observation also leads to a relaxed constraint on the complexity of the encoder, and highlights the need for video scalability. As a consequence of the relaxed video-complexity constraint for entertainment-oriented IP streaming, there is no need to *totally* avoid such techniques as motion estimation which can provide a great deal of coding efficiency when compared with replenishment-based solutions [24].

Although it is desirable to generate a scalable video stream for a wide range of bit-rates (e.g. 15 kbps for analog-modem Internet access to around 1 Mbps for cable-modem/ADSL access), it is virtually impossible to achieve a good coding-efficiency/video-quality tradeoff over such a wide range of rates. Meanwhile, it is equally important to emphasize the impracticality of coding the video content using simulcast compression at multiple bit-rates to cover the same wide range. First, simulcast compression requires the creation of many streams (e.g. at 20, 40, 100, 200, 400, 600, 800 and 1000 kbps). Second, once a particular simulcast bitstream (coded at a given bit-rate, say R) is selected to be streamed over a given Internet session (which initially can accommodate a bit-rate of R or higher), then due to possible wide variation of the available bandwidth over time, the Internet session bandwidth may fall below the bit-rate R. Consequently, this decrease in bandwidth could significantly degrade the video quality. One way of dealing with this issue is to switch, in real-time, among different simulcast streams. This, however, increases complexities on both the server and the client sides, and introduces synchronization issues.

A good practical alternative to this issue is to use video scalability over *few ranges* of bit-rates. For example, one can create a scalable video stream for the analog/ISDN access bit-rates (e.g. to cover 20–100 kbps bandwidth), and another scalable stream for a higher bit-rate range (e.g. 200 kbps–1 Mbps). This approach leads to another important requirement. Since each scalable stream will be build on the top of a video base layer, this approach implies that multiple base layers will be needed (e.g. one at 20 kbps, another at 200 kbps, and possibly another at 1 Mbps). Therefore, it is quite desirable to deploy a video compression standard that provides good coding efficiency over a rather wide range of possible bit-rates (in the above example 20 kbps, 200 kbps and 1 Mbps). In this regard, due to the many video-compression tools provided by MPEG-4 for achieving high coding efficiency and in particular at low bit-rates, MPEG-4 becomes a very attractive choice for compression.

2.3. Streaming server complexity

Typically, a unicast server has to output tens, hundreds, or possibly thousands of video streams simultaneously. This greatly limits the type of processing the server can perform on these streams in real-time. For example, although the separation of an MPEG-2 video stream into three temporal layers (I, P and B) is a feasible approach for a scalable multicast (as proposed in [22]), it will be quite difficult to apply the same method to a large number of unicast streams. This is the case since the proposed layering requires some parsing of the compressed video bitstream. Therefore, it is desirable to use a very simple scalable video stream that can be easily processed and streamed for unicast sessions. Meanwhile, the scalable stream should be easily divisible into multiple streams for multicast IP similar to the receiver-driven paradigm used in [22,24].

Consequently, we adopt a single, fine-granular enhancement layer that satisfies these requirements. This simple scalability approach has two other advantages. First, it requires only a single enhancement layer decoder at the receiver (even if the original fine-granular stream is divided into multiple sub-streams). Second, the impact of packet losses is localized to the particular enhancement-layer picture(s) experiencing the losses. These and other advantages of the proposed scalability approach will become clearer later in the paper.

2.4. Client complexity and client-server communication issues

There is a wide range of clients that can access the Internet and experience a multimedia streaming application. Therefore, a streaming solution should take into consideration a scalable decoding approach that meets different client-complexity requirements. In addition, one has to incorporate robustness into the client for error recovery and handling, keeping in mind key client-server complexity issues. For example, the deployment of an elaborate feedback scheme between the receivers and the sender (e.g. for flow control and error handling) is not desirable due to the potential implosion of messages at the sender [2,34,35]. However, simple re-transmission techniques have been proven effective for many unicast and multicast multimedia applications [2,10,22,34]. Consequently, we employ a re-transmission method for the recovery of lost packets. This method is combined with a client-driven flow control model that ensures the continuous decoding and presentation of video while minimizing the server complexity.

In summary, a real-time streaming system tailored for entertainment IP applications should provide a good balance among these requirements: (a) scalability of the compressed video content, (b) coding efficiency across a wide range of bit-rates, (c) low complexity at the streaming server, and (d) handling of lost packets and end-to-end flow control using a primarily client-driven approach to minimize server complexity and meet overall system scalability requirements. These elements are addressed in our streaming system as explained in the following sections.

3. An overview of the scalable video streaming system

The overall architecture of our scalable video streaming system is shown in Fig. 1.[2] The system consists of three main components: an MPEG-4 based scalable video encoder, a real-time streaming server, and a corresponding real-time streaming client which includes the video decoder.

MPEG-4 is an international standard being developed by the ISO Moving Picture Experts Group for the coding and representation of multimedia content.[3] In addition to providing standardized methods for decoding compressed audio and video, MPEG-4 provides standards for the representation, delivery, synchronization, and interactivity of audiovisual material. The powerful MPEG-4 tools yield good levels of performance at low bit-rates, while at the same time they present a wealth of new functionality [20].

The video encoder generates two bitstreams: a base-layer and an enhancement-layer compressed video. An MPEG-4 compliant stream is coded based on an MPEG-4 video Verification Model (VM).[4] This stream, which represents the base

[2] The figure illustrates the architecture for a single, unicast server-client session. Extending the architecture shown in the figure to multiple unicast sessions, or to a multicast scenario is straightforward.

[3] http://drogo.cselt.stet.it/mpeg/

[4] The VM is a common set of tools that contain detailed encoding and decoding algorithms used as reference for testing new functionalities. The video encoding was based on the MPEG-4 video group, MoMuSys software Version VCD-06-980625.

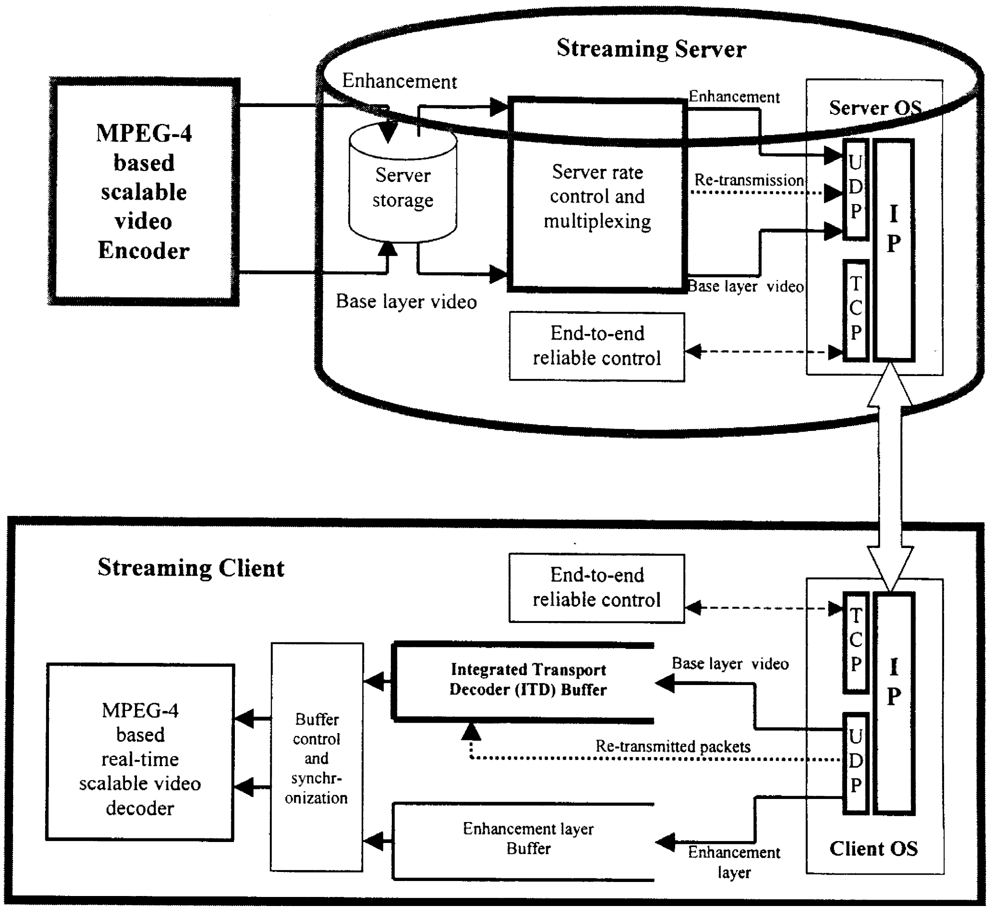

Fig. 1. The end-to-end architecture of an MPEG-4 based scalable video streaming system.

layer of the scalable video encoder output, is coded at a low bit-rate. The particular rate selected depends on the overall range of bit-rates targeted by the system and the complexity of the source material. For example, to serve clients with analog/ISDN modems' Internet access, the base-layer video is coded at around 15–20 kbps. The video enhancement layer is coded using a single fine-granular-scalable bitstream. The method used for coding the enhancement layer follows the recent development in the MPEG-4 video fine-granular-scalability (FGS) activity for Internet streaming applications [4,5]. For the above analog/ISDN-modem access example, the enhancement layer stream is over-coded to a bit-rate around 80–100 kbps. Due to the fine granularity of the enhancement layer, the server can easily select and adapt to the desired bit-rate based on the conditions of the network. The scalable video coding aspects of the system are covered in Section 4.

The server outputs the MPEG-4 base-layer video at a rate that follows very closely the bit-rate at which the stream was originally coded. This aspect of the server is crucial for minimizing underflow and overflow events at the client. Jitter is introduced at the server output due, in part, to the packetization of the compressed video streams. Real-time Transport Protocol (RTP) packetization [15,39] is used to multiplex and synchronize the

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.