DELIVERING OBJECT-BASED AUDIO VISUAL SERVICES

Hari Kalva†, Javier Zamora‡, and Alexandros Eleftheriadis†

†Columbia University, New York ‡Xbind, Inc., New York

Abstract

Delivery of audio-visual objects differs from the traditional video on demand systems in the characteristics of the presentations delivered. Video on demand has mainly been about delivering MPEG-2 audio and video[1]. In the case of object based presentations such as MPEG-4 presentations, the media data and the media composition data are transmitted to a client as separate streams, typically with different QOS requirements, in the same session. Furthermore, as the number of objects in a presentation can be quite large, the overhead required to manage the session is large. Interactivity makes this problem more complex as the resources required for a session will now depend on the user behavior, especially when user interaction with objects changes the number of objects in the scene either by adding or deleting objects. In this paper, we present our recent work on delivering audio-visual services based on MPEG-4. The issues covered include object-based audio-visual (MPEG-4) servers, object scheduling, object-to-object communications, and content creation.

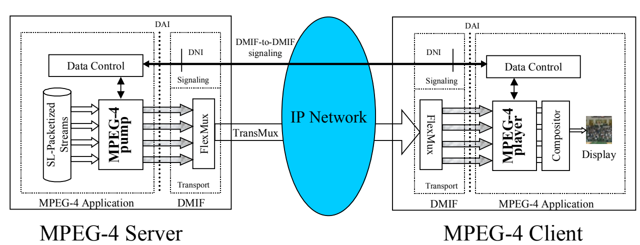
INTRODUCTION

The image and video encoding has been totally transformed with the advent of new coding and representation techniques [6]. These next generation coding techniques have made possible encoding and representation of audio-visual scenes with semantically meaningful objects. Such a new paradigm of object-based representation of audio-visual scenes and presentations will change the way audio-visual applications are created.

MPEG-4 is specifying tools to encode individual objects, compose presentations with objects, store these object based presentations and access these presentations in a distributed manner over networks. The MPEG-4 Systems specification [4] defines the architecture and tools to create audio-visual scenes from individual objects. The scene description and synchronization tools are the core of the systems specification. The scene description is encoded separately and is treated as another elementary stream. This separation allows for providing different Quality of Service (QOS) for scene description which has very low or no loss tolerance and media streams in the scene which are usually loss tolerant. MPEG-4 scene description, also referred to as BIFS (Binary Format for Scenes), is based on VRML and specifies the spatio-temporal composition and dynamic behavior of scenes. BIFS update commands can be used to create scenes that evolve over time. This architecture allows creation of complex scenes with potentially hundreds of objects, thus calling for a high rate of establishment and release of numerous short-term transport channels with the appropriate QoS. Traditional methods of signaling are not adequate to meet this demand because of the high overhead introduced. Furthermore, the applications should not depend on the underlying transport network. The Delivery Multimedia Integration Framework (DMIF) is a general application and transport delivery framework specified by MPEG-4 [3]. DMIF's main purpose is to hide the details of the transport network from the user, as well as to ensure signaling and transport interoperability between end-systems. In order to keep the user unaware of underlying transport details, MPEG-4 defined an interface between user level applications and DMIF, called the DMIF Application Interface (DAI). The DAI provides the required functionality for realizing multimedia applications with QOS support [2].

In this paper we will present our recent work on delivering audio-visual services. The issues covered will include object-based audio-visual (MPEG-4) servers, object scheduling, session management, QOS management, and content creation.

SYSTEM ARCHITECTURE



The MPEG-4 system developed is an end-to-end system consisting of an MPEG-4 server, the DMIF component for transport, signaling, and session management, an IP network, and an MPEG-4 client for media playback/rendering. Figure 1 shows the components of the system.

The complexity of the player (i.e., client) has grown as a result of the new features and functionality offered by

MPEG-4. The player is now also responsible for composing a scene from individual objects in addition to decoding and displaying the objects. A player consists of three logical components, a DMIF instance, elementary stream decoders, and a compositor. The DMIF instance is responsible for managing data access from a network or a file. A player typically contains several decoders each handling a specific elementary stream. Elementary streams are audio-visual streams as well as streams that describe the composition, rendering, and behavior of a presentation. Each object in a presentation is carried in a separate elementary stream. As MPEG-4 presentations can include media objects from several sources, potentially with different clock frequencies, there is additional burden on the client to track multiple clocks. Because of this additional complexity, a player's performance depends on the complexity of the content. Intelligent resource management and usage is necessary to use the resources such as memory efficiently.

CONTENT DELIVERY

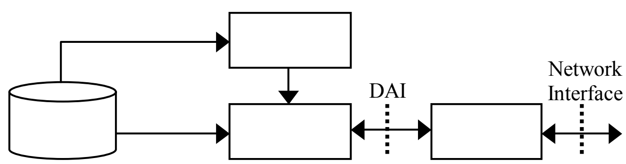


Figure 2. Server Components

Figure 2 shows the components of an MPEG-4 server. An MPEG-4 server typically consists of an MPEG-4, pump, an object scheduler, and a DMIF instance for media transport and signaling. The server delivers Sync Layer Packets (SL-Packets) to the DMIF layer, which multiplexes them in a FlexMux [4] and transmits them to the client. An MPEG-4 server that is transmitting objects should make sure that an access units arrive at the terminal before their decoding time. The server delivers objects in a presentation as scheduled by the presentation scheduler. The scheduler uses the decoding timestamps to schedule the delivery of access units.

The flexibility of MPEG-4 while allowing interactive and complex presentations makes the content creation process non-trivial. Unlike MPEG-2, content creation process involves much more than multiplexing the media streams. The scheduler is also useful during content creation process to determine if the presentation being designed can be scheduled for specific channel rates and client buffer capacity. It may not be possible to find a solution for a given set of resources; i.e, presentation cannot be scheduled with given resources. In order to create a schedulable presentation, some constraints may be relaxed. In the case of scheduling objects, relaxing a constraint may involve increasing the buffer capacity, increasing the channel capacity, not scheduling some object instances, or removing some objects from a presentation.

Network delays and data loss could occur in the system if the contentdoes not request the suitable QoS. A presentation created without the knowledge of target networks and clients could create long startup delays, buffer overflows or underflows. This could cause distortion, gaps in media playback and problems with the synchronization of different media streams. Unlike MPEG-1 and MPEG-2, MPEG-4 presentations are not constant bit rate presentations. The bit rate of a presentation may be highly variable depending on the objects used in the presentation. Further more, there is no notion of bit rate when images are used. These characteristics of MPEG-4 presentations make it difficult to design servers as well as content. A presentation may have to be recreated for different targets or servers have to be intelligent enough to scale a presentation for different networks/clients. Schedulers should be part of content creation process to check the suitability of content for target networks and clients. MPEG-4 has scalable coding tools that allow creation of content that can be adapted to different network and bandwidth conditions.

REFERENCES

1. S.-F. Chang *et. al*, "Columbia's VoD and Multimedia Research Testbed with Heterogeneous Network Support", Journal on Multimedia Tools and Applications, Special Issue on Video on Demand, Kluwer Academic Publishers, Vol. 5, Nr. 2, September 1997, pp. 171-184.
2. J.-F. Huard *et al.,* "Realizing the MPEG-4 Multimedia Delivery Framework," IEEE Network Magazine, pp. 35-45, November/December 1998. Special issue on Transmission and Distribution of Digital Video.
3. ISO/IEC 14496-6 FDIS "Delivery Multimedia Integration Framework, DMIF," October 1998.
4. ISO/IEC/SC29/WG11, "Generic Coding of Moving Pictures and Associated Audio (MPEG-4 Final DIS) ISO/IEC 14496-1," International Standards Organization, November 1998.
5. A. Puri and A. Eleftheriadis, "MPEG-4: A Multimedia Coding Standard Supporting Mobile Applications" *ACM Mobile Networks and Applications Journal*, Special Issue on Mobile Multimedia Communications, Vol. 3, No. 1, June 1998, pp. 5-32 (invited paper).
6. L. Torres, M. Kunt, eds., "Video Coding : The Second Generation Approach," Kluwer Academic Publishers, 1996.