An MPEG-4 Based Interactive Multimedia System

Bo Li†, Ishfaq Ahmad†, Ming L. Liou‡, Yong He‡ and Bruce K. Chan†

†Department of Computer Science
‡Department of Electrical and Electronic Engineering,
The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
The Hong Kong University of Science and Technology, Hong Kong

Abstract†

With the rapid convergence in telecommunications, computer, and TV/film industries, rigorous efforts are in progress for designing the emerging multimedia standard — MPEG-4. The main features of MPEG-4 are its *content-based interactivity, efficient coding,* and *universal access.* For realizing MPEG-4 based application oriented systems, various issues must be carefully addressed. This paper describes a full scale MPEG-4 based interactive multimedia system project underway at the Video Technology Center of the Hong Kong University of Science and Technology. This project aims to provide complete MPEG-4 based interactive services tailored to various applications. In order to do so, a number of key enabling technologies need to be investigated. These include encoding and decoding, multimedia server design, a user interface, and MPEG-4 stream transmission over high speed networks. The unique features of the project are: 1) to explore the feasibility of parallel and distributed technology using Network-of-Workstations (NOWs) to hinder the intensive computation required for real-time encoding; 2) to provide flexibility of software-based encoding/decoding; 3) to design an intelligent transport service mechanism which can satisfy QoS requirement of MPEG-4.

1 Introduction

Due to the rapid progress in information technology, the traditional boundaries between the areas of telecommunication, computer and TV/film are blurring. As a result, multimedia-based computer systems, and interactive video are being added to telecommunications. Audio-visual information based multimedia services are now considered to be an integral part of our daily lives, and their essential role in the success of industrial and commercial business is well recognized. With the availability of network infrastructure in place, such as the Internet, such services are now widespread and are creating new ways of education, entertainment, communication, and business opportunities. A wide range of such services such as HDTV transmission, digital terrestrial television broadcast, home television theatre, Photo-CD, interactive video games, Video On Demand (VOD), medical imaging, scientific visualization, video communications on Asynchronous Transfer Mode (ATM) networks, video conferencing, multimedia mailing, remote video surveillance, satellite news gathering, and network database services, are now being realized in both experimental and commercial domains.

These applications require ways for communication, access, and manipulation of audio-visual data. Due to the enormous amount of digital video data, compression of video is an inevitable requirement for its effective communication. During the past decade, significant progress has been made in these technologies leading to a variety of techniques to support creation, transmission, presentation, and synchronization of multimedia information. Within these developments, compression technology, due to its capability of making the storage and transmission of digital data more efficient, is at the forefront of audio-visual (AV) representation and communication arena. Standardization of coding algorithms is required to enable the sharing of technology among various industries. In the past, a number of coding standards, such as JPEG, MPEG1/2, and H.261/H.263, have been proposed in recent years and are now in commercial use.

The current and emerging multimedia services demand much more rich set o functionalities than the ones offered by the traditional standards. For example, mobile communication and database access require very low bitrate video coding and error resilience across various networks; virtual reality requires integration of natural and synthetic hybrid object coding; interactive video games require a high degree of object based interactivity. Instead of traditional frame based interaction such as fast-forward, fast-backward, etc., new ways of interactivity are needed to efficiently realize such applications.

The new standard, MPEG-4, which is currently being developed by MPEG and is scheduled to be finalized in November 1998, is gaining immense interest across industries. The aim of MPEG-4 is to support content-based communication, access, and manipulation of digital audio-visual objects. Compared to the conventional frame-based compression techniques, the object-based representation and coding of the multimedia information enable MPEG-4 to cover a broad range of emerging applications. In addition, MPEG-4 also supports new functionalities not available in existing standards.

This paper describes a full scale MPEG-4 based interactive multimedia system we are currently building for supporting a variety of applications. The system is conceptually based on a client-server model, with a number of clients making interactive requests to a server. It consists of an encoder, decoder, database, a user-interface and a network backbone connecting all components. The server encodes and delivers the requested audio-visual information using MPEG-4 format. The applications are aimed to be truly interactive in that the user is able to modify, edit, and manipulate the objects in a scene. A client can manipulate the content of the multimedia document on its site or interact with the server to request more drastic changes. This project is carried out based on the state-of-art facilities we have at the Video Technology Center of the Hong Kong University of Science and Technology, and involves a number of key enabling technologies including: encoding and decoding, multimedia server design, a user interface, and the efficient transport mechanism.

The rest of the paper is organized as follows. An overview of MPEG-4 is given in Section 2. We describe the interactive multimedia system that we are building in Section 3, which

†. This work is supported in part by a grant from the Hong Kong Telecom Institute of Information Technology (HKTIIT).

includes three major elements: encoding and modeling, real-time scheduling and transport issues. Section 4 concludes the paper.

2 Overview of MPEG-4

Several standards for representing and coding digital audio-visual data have been recommended. These standards reduce the size of digital data through compression and facilitate creation, storage, transmission, and presentation of multimedia information by providing compatibility among various supporting technologies. H.261, developed by ITU, is targeted for audiovisual service at $p \times 64$ kbps ($p = 1, \ldots, 30$) bit rate which covers the entire ISDN channel capacity [14]. The successor to H.261, H.263 is intended for compressing the moving picture component of audio-visual services and applications at a very low bitrate [5], yet offering improved picture quality. MPEG-1, the first standard developed by the MPEG (Moving Picture Expert Group) of ISO/IEC, specifies the coding of audio-visual data at a bitrate of up to 1.5Mbps [11]. The second standard, MPEG-2, is a generic coding standard for low to high-resolution moving pictures and associated audio data, with the bitrate ranging from 2Mbps to 30Mbps [1], [8].

MPEG-4 is scheduled to become an international standard in November 1998. During the development, the so called "Verification Model" (VM) methodology is adopted to specify the candidate technologies which may be included in the final standard [3], [20]. The VM is supposed to evolve through a core experimental process. MPEG-4 video VM is one of the main parts of MPEG-4 with the objective to support three major functionalities: content-based interactivity, coding efficiency, and universal access [19]. Its bitrate can range from 10kbits/s up to several Mbits/s. The spatial resolutions include SQSIF/SQCIF, QSIF/QCIF, SIF/CIF, 4*SIF/CIF, and CCIR601. In contrast to the existing 'frame-based' or 'pixel-based' standards used in MPEG-1/MPEG-2 and H.261/H.263, MPEG-4 video is object-based hybrid coding standard which specifies the technologies for representing and processing video object efficiently to support various content-based functionalities within the compression domain. A user at the decoder side can access arbitrarily shaped objects in the scene or request the encoder to manipulate the objects and then deliver the requested objects.

The current functionalities supported by MPEG-4 can be clustered into *three* classes [9]: content-based interactivity, high efficient compression and universal access. *Content-based interactivity* involves the ability to interact with meaningful objects in an audio-visual scene which is crucial for the applications such as interactive home shopping, video games and content-based storage and retrieval. *High compression* is essential for efficient use of storage and transmission bandwidth. *Universal access* provides error robustness for storage and communication applications in error-prone environment, especially for the wireless communications.

Figure 1 is the overall architecture of MPEG-4 system [3]. One or more AV objects, including their spatio-temporal relationships, are transmitted from an encoder to a decoder. At the encoder, the AV objects are compressed, error protected, multiplexed, and transmitted downstream. The transmission may occur across multiple channels offering various qualities of service. At the decoder, the AV objects are demultiplexed, error corrected, decompressed, composited, and presented to an end user. The end user is given an opportunity to interact with the presentation. Interaction information can be used locally, or transmitted upstream to the encoder.

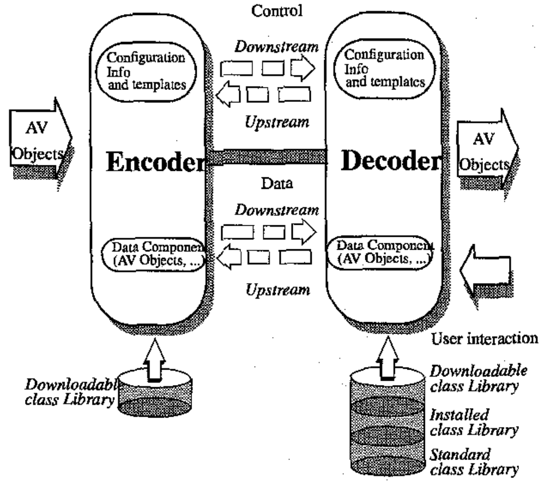

Figure 1: MPEG-4 encoder and decoder architecture.

The stream can be classified either as control or data streams. Control streams carry control information, such as connection setup, profile, and class definition information. Data streams carry all other information. Control streams must be transported over reliable channels because of the critical nature of control information. However, data streams may be transported over a variety of channels with different qualities of service (QoS).

Before the AV objects are transmitted, the encoder and decoder must exchange configuration information. The encoder determines which classes of algorithms, tools and other objects are needed by the decoder to process the AV objects. Each class of objects can be defined by a data structure plus executable code. The definitions of any missing classes are downloaded to the decoder. As the decoder executes, new class definitions may be needed in response to user interaction. The decoder can request that the encoder download specific additional class definitions. The additional class definitions may be downloaded in parallel with the transmitted data.

With such flexible mechanism and toolbox approach, MPEG-4 will enable the user to configure and build systems for many applications with varying requirements on different aspects. In addition, MPEG-4 is also capable to cope flexibly with newly emerging advanced technologies.

The AV object defined by MPEG-4 is a representation of a real or virtual object that can be manifested aurally and/or visually, such as conventional video and audio, static 2D images, VRML, synthetic audio (MIDI,...), etc. Each AV object has its local ($3D+T$) coordinate system serving as a handle for manipulating in space and time. Either the encoder or the end-user can place an AV object in a scene by specifying a coordinate transformation from the object's local coordinate system into a common, global $3D+T$ coordinate system, or scene coordinate system. The composition feature of MPEG-4 makes it possible to do the bitstream editing in compressed domain which is an important feature for the content-based functionalities in MPEG-4.

MPEG-4 video is one of the major parts of MPEG-4 standard which aims at providing standardized core technologies allowing efficient storage, transmission and manipulation of video data in multimedia environments [19]. In order to support a broad spectrum of functionalities such as efficient compression, object scalability, spatial and temporal scalability, error resilience, MPEG-4 video will provide a toolbox containing tools and algorithms bringing solutions to these functionalities and more.

The overall structure of MPEG-4 video coder is based on the concept of video object planes (VOPs) defined as the instances of video objects at a given time. The video encoder is composed of identical VOP encoders. Each object is segmented from the input video signal and performed the same encoding scheme separately, the bitstreams of different VOPs are then multiplexed and transmitted. At the decoder, the received bitstreams are demultiplexed and decoded by each VOP decoder, the reconstructed video objects are composited by the composition information which is sent together with the bitstream and presented to the user. The user interaction with the objects such as scaling, dragging, replacement and linking can be handled either on the encoder or on the decoder.

3 An Interactive Multimedia System

Our interactive multimedia system (see Figure 2) under construction is generic in that it provides basic functionalities, and can be easily tailored to various environments. At present, it is aimed at providing object-oriented, content-based

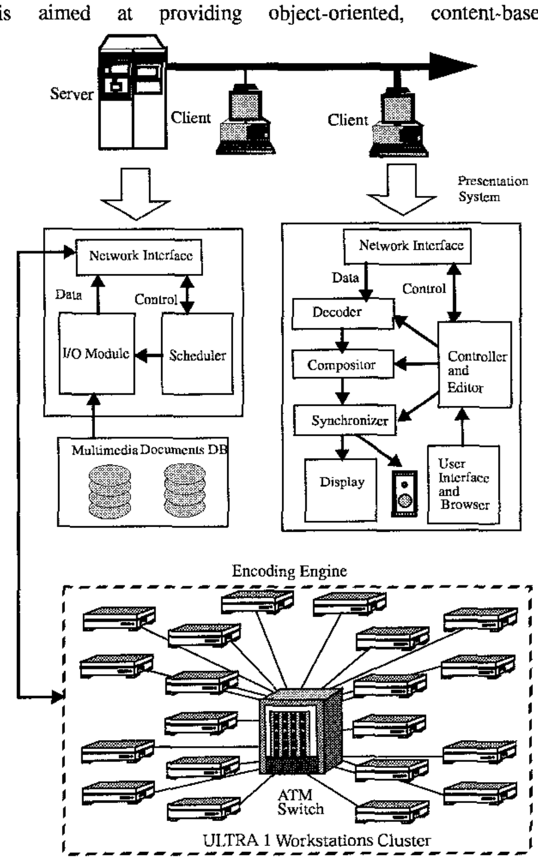

Figure 2. The MPEG-4 based interactive multimedia system.

functionalities where the user can access, modify, and arrange various objects in a scene. The system consists of a server and a number of clients connected via an ATM network. The heart of the server includes a cluster of Ultra-1 workstations that collectively provide the computing power for the real-time encoder. At present no real-time MPEG-4 encoder is reported except our own (see [6]) which is software-based and has been implemented using parallel processing.

The scheduler at the server retrieves the video data from the database, partitions it into smaller pieces and assigns those pieces to the workstations which encode the data in parallel. The encoded bitstreams from individual workstations are multiplexed into a single bitstream which is then delivered to the clients. The architecture of a client includes the decoder, a scene compositors, and a synchronizer. The user interface allows to perform fast editing operations locally; more extensive changes are carries out at the encoder.

The encoding and delivery of MPEG-4 should preserve the spatio-temporal relationships between various audio-visual objects. Before assigning these objects to the workstations for encoding and then transporting them through the network, the spatio-temporal relationships must be captured using a modeling scheme. The information generated by the model includes the synchronization requirements of various objects at various levels as well as their display deadlines. This information is used by a scheduling algorithm for allocating objects to workstations for encoding. The coded bitstream also needs to be scheduled for transporting the video data over the network. We next describe in detail the modeling, scheduling, and transport techniques.

3.1 Modeling and Encoding

In MPEG-4 video VM encoder, one of the most important issues to consider is the synchronization of various video objects. Each object may have certain presentation timing constrains which, in turn, may be dependent on the other objects. The playout time requirement and associated synchronization constrains among multiple video objects must be satisfied in real time to guarantee a smooth flow of video sequence presented to the user.

To identify the timing constraints among multiple objects, a synchronization reference model is required to describe various presentation requirements and temporal relationships for determining an appropriate scheduling scheme. There are several modeling tools proposed in [2] for specifying the temporal behavior of various multimedia systems. We choose Petri nets as the modeling tool since it is a simple but effective tool for describing and studying the systems with both concurrent and sequential activities.

Petri net is a modeling tool for describing systems with concurrent, asynchronous, distributed, parallel, non-deterministic, and/or stochastic characteristic. Most variations of the Petri nets model, such as OCPN [15] have been widely used in manufacturing and automation applications, computer networks and multimedia communication applications due to its intuitive graphical representation and the simplicity of the modeling concept.

A marked Petri net can be mathematically defined as 4-tuple (P,T,A,M) where:

$P = (p_1, p_2, p_3, \ldots, p_m)$ is a set of places,

$T = (t_1, t_2, t_3, \ldots, t_m)$ is a set of transitions,

$A \subseteq \{T \times P\} \cup \{P \times T\}$ is a set of arcs,

M: $P \rightarrow I$, I = (0, 1, 2, 3, ...) associates a marking to each place in the net.

Figure 3 depicts a graphical representation of a Petri net. The circles and bars represent the places (P) and transitions (T), respectively, the arcs (A) indicate both input and output flow directions. A Petri net is executed by the firing rules that transmit the marks (M) or tokens from one place to another, and such firing is enabled only when each input place has a token inside. Thus, by using firing transition and a tokens distribution state, Petri nets can describe the information flow or system activities in a straightforward way.

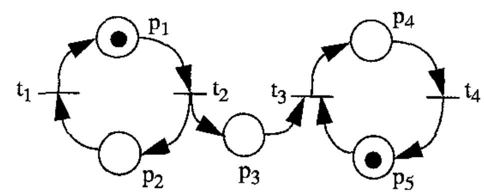

Figure 3. A graphical representation of a Petri net.

A video may have many sessions and each session may involve many objects and layers of objects which in turn may have multiple instances in time. A VOP lasts over a number of video frames. In the Petri nets model, we define the places as *object intermedia unit* (OIU) and the transition as *object synchronization point* (OSP). At the VS level, an OIU represents the whole video session with just two OSPs (the session start and end point). At the VO level, each OIU represents one object within the session; here the OSPs indicate the temporal relationship and synchronization constrains among various objects. At the VOP level, each OIU represents one frame of the object whereas the OSPs indicate the intra inter VOPs synchronization on the frame level.

By using a hierarchical model we can achieve coarse or fine grained synchronization by applying a scheduling scheme on different levels. Figure 4 shows the playout time chart of a general MPEG-4 video example. The session has 4 video objects (VOs); VO_0, VO_1, VO_3 start at time 0, and VO_2 starts at time unit 4. VO_0 and VO_1 end at time unit 4, while VO_2 and VO_3 end at time unit 12. The frame rates of VOs are different. The duration of a frame for VO_1 is 1 time unit, while the duration of a frame is 2 for VO_1 and 4 for VO_2 and VO_3. Figure 5 represents the hierarchical Petri net model for the above case.

3.2 Real-time Scheduling

Assuming that the Quality of Service (QoS) can be guaranteed by the network for transmitting the synchronized video stream between source and destination, it is important for the encoder to schedule the appropriate tasks in order to satisfy the synchronization requirement and to meet presentation deadlines.

The objective of real-time scheduling is to assign the tasks to the available processors and determine the execution order of each task so that tasks are completed before their deadlines [18]. A real-time scheduling can be characterized as being either static and dynamic. In static scheduling, the algorithm determines the schedule with the complete knowledge of all the tasks in advance. In contrast, a dynamic scheduling algorithm deals with task assignment at run-time because the information about the tasks is

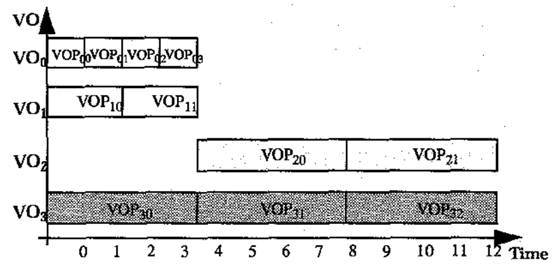

Figure 4. Playout time chart for video session

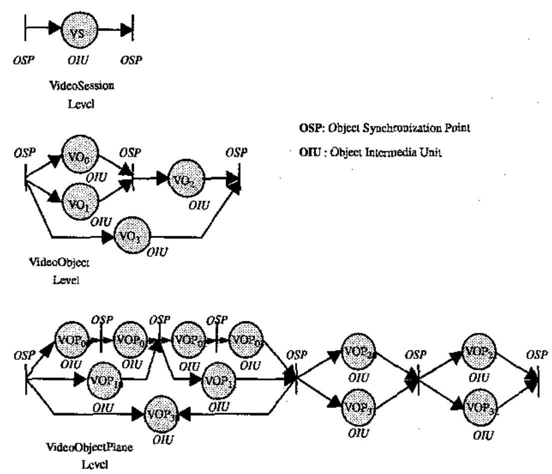

Figure 5. The hierarchical Petri net model for the previous example.

not available in advance. Static scheduling incurs little run-time cost but cannot adapt to the indeterministic behavior of the system. On the other hand, dynamic scheduling is more flexible as it can be adjusted to system changes but incurs a high run-time cost.

A number of scheduling algorithms have been developed for both distributed and parallel systems [18]. In our implementation [6], we use a variant of the ***earliest-deadline-first*** (EDF) algorithm which has been widely employed in many applications [21]. The principle of this algorithm is that the tasks with earlier deadlines are assigned higher priorities and run before tasks with lower priorities. In our implementation, VOPs with the earlier playout deadlines or synchronization points get to be encoded and delivered first. For the tasks with the same deadline, we assign a portion of available processors to each object, with the number of allocated processors depending upon the size ratio among these objects because a video object with a larger size generally requires more computing and vice versa.

4 MPEG Delivery over ATM backbone

Given that the MPEG-4 has yet to become a standard, there has been little study on how to effectively delivery the MPEG-4 traffic stream over a network. In this project, we will investigate the issues associated with delivering MPEG-4 over an ATM backbone, in particular, how to efficiently schedule and transport the MPEG-4 bitstreams over the ATM network. Notice, however, the modeling of the encoded bitstream, and the scheduling thereafter, does not accurately represent the bitstreams fed into the network. This is highly dependent on the transport layer protocol(s) which we are describing in this subsection.

4.1 MPEG-4 Transport Issues

What MPEG-4 encoder generates is referring to as *Raw Data*, which includes the audiovisual object binary files/streams and the Scene Description file/streams. They need to be multiplexed together before being transmitted. Recall MPEG-4 audiovisual objects have a both spatial and temporal extent while the Scene Description, as part of the MPEG-4 Systems specification, describes the format for transmitting the spatial-temporal positioning information that individual audiovisual objects are composed within a particular scene. Two methodologies can be adopted for scene description: parametric such as the Binary Format for Scenes (BIFS) and programmatic (e.g., AAS, Adaptive Audio-Visual Session format). We are currently focusing on BIFS only in this project.

Before MPEG-4 encoded data can be transported, it has to go through the usual necessary networking multiplexing functions; in addition, the MPEG-4 timing information needs extra processing. Figure 6 illustrates the MPEG-4 *system layer model.* The A/V objects in MPEG-4 are conveyed in one or more Elementary Streams (ES), which define the QoS that requires as well as timing information. MPEG-4 decoder, however, only deals with a single entity called Access Units (AU), therefore ES must be fragmented into AU before transmission. The transport layer deals with synchronization between source and destination, exploiting the different QoS available from the network [4] [10]. The transport of MPEG-4 is divided into a number of sub-layers, including an elementary stream layer, access unit layer, an optional flexible multiplexing layer and a transport multiplexing layer. Their respective main functions are described below:

Elementary Stream Layer: responsible for transforming those raw data (AV object binary streams and BIFS) into Elementary Stream by using the ObjectDescriptor. Specifically, two functions are carried out, 1) recovering the timing information (encoder and decoder synchronization and the time stamps of the encoded AV data); 2) adding the additional QoS requirement specification needed by decoder, scene description using the ObjectDescriptor.

Access Unit Layer: specifies a syntax for the fragmentation of Elementary Streams into Access Units or parts thereof. These fragments are called AL-PDUs (AU Layer - Protocol Data Unit). The sequence of AL-PDUs resulting from one Elementary Stream is called AL-packetized Stream (APS). Access Units are used as the basic unit for synchronization. An AL-PDU consist of an AL-PDU Header and the AL-PDU Payload. The AL-PDU Header supplies means for checking possible data loss and carries the encoded representation of the time stamps and associated information.

FlexMux Layer: (Flexible Multiplexing) is an optional layer that multiplexes one Elementary Stream (ES) or one AL-PDU into one FlexMuX-PDU under *simple mode*, and multiple AL-PDUs into one FlexMux-PDU under *MuxCode mode.* This can be used, for example, to group ES with similar QoS requirements. Under the *MuxCode mode*, FlexMuX can accommodate the interleaving of Elementary Streams with largely varying instantaneous bit rate. The sequence of FlexMux-PDUs that are interleaved into one stream are called FlexMux Stream. The FlexMux Layer provides identification of AL-PDUs form different Elementary Streams by means of FlexMux Channel numbers. Hereby each AL-packetized Stream is mapped into one FlexMux Channel.

TransMux Layer: (Transport Multiplexing) offers transport services matching the requested QoS. MPEG-4 only specifies the interface to this layer, allow MPEG-4 to be used under a variety of existing protocol stacks such as (RTP)/UDP/IP, (AAL5)/ ATM, or MPEG-2 Transport Streams (TS).

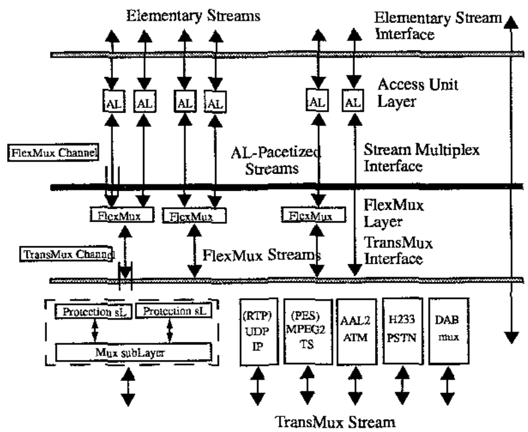

Figure 6: The MPEG-4 System Layer Model.

4.2 Implementation and Issues to be addressed

Choosing ATM as the underlying transport network in our implementation is simply because the inherent QoS guarantee nature of the ATM is well suited for supporting MPEG-4 interactive and real-time services. In addition multiple variable bit rate streams can also be easily supported.

As illustrated in Figure 2, our system consists of 20 Ultra-1 workstations connected with a ForeRunner ATM switch ASX-1000 which can provide flexible port speeds from 1.5 Mbps to 622 Mbps. Each workstations is assigned dedicated Private Virtual Circuits (PVCs) connect to the ATM switch and all the data stream will be transported through those PVCs. The Fore ATM API is used in the implementation.

As described in TransMux layer, the MPEG-4 can be transported under a variety of protocol stacks. At the current stage of the this project, we have so far focused on the implementation of Access Unit and TransMux layers. The following two encapsulations will be employed: one uses AAL5/ ATM (ATM Adaptation Layer 5); the other uses MPEG-2 Transport Streams (TS). We adopted use of the *simple mode* of the FlexMuX layer, which only supports one single AL-PDU in each FlexMux-PDU.

Existing studies have shown that video traffic characteristics has significant impact on the network performance [7] [16]. This project aims to examine a number of issues in delivering MPEG-4 over ATM, specifically, the following issues will be addressed:

Traffic measurement: Traffic measurement is a crucial indicator to the design of a number of network elements, especially traffic control mechanism. The characteristics of both traffic streams have to be carefully measured, these include the MPEG-4 source traffic, i.e., traffic generated from the encoder,

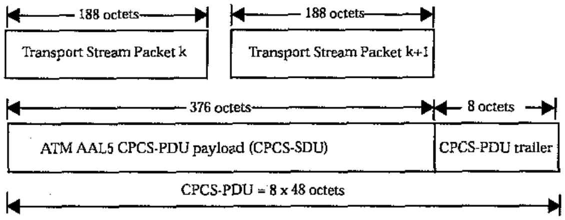

Figure 7: Format of AAL-5 PDU Containing 2 TS Packets.

and the traffic within the network. Specifically, we will compare the ATM cell level measurement and the source rate statistics of the MPEG-4 video bit rate.

Traffic Modeling and scheduling: Our existing work using a hierachical Petri net model has successfully modeled the encoded MPEG-4 traffic [6]. However, this does not accurately represent the characteristics of the video traffic submitted to the ATM network. As illustrated in Figure 6, this strongly depends on the protocol stack employed. Efforts are in progress to extend this model to capture the behavior of transport traffic streams. Furthermore, due to the interactive nature of the MPEG-4, the video objects have to be scheduled and delivered upon users' request. How to schedule multiple traffic streams, over potentially multiple logical channels, while taking into consideration the unique characteristics of MPEG-4 and optimizing the needed performance, remains a challenge

Traffic Analysis: The resource dimensioning, both bandwidth and buffer, relies on the *accuracy* of the performance model [12]. It has been shown that the performance based on steady-state analysis cannot provide reliable QoS for video traffic [13]. We are currently developing performance model based on the transient analysis, and further investigate the resource dimensioning for video traffic based on such transient analysises.

5 Conclusions

This paper describes an on-going project for designing a full scale MPEG-4 based interactive multimedia system. This project aims to investigate a number of key enabling technologies of MPEG-4 including: encoding and decoding, multimedia server design, a user interface, and MPEG-4 stream transmission over high speed networks. The unique features of the project are: 1) to explore the feasibility of parallel and distributed technology using Network-of-Workstations (NOWs) to hinder the intensive computation required; 2) to provide flexibility of software-based encoding/decoding; 3) to design an intelligent transport service mechanism which can satisfy QoS requirement of MPEG-4.

References

[1] S. M. Akramullah, I. Ahmad, and M. L. Liou, "Performance of Software-Based MPEG-2 Video Encoder on Parallel and Distributed Systems," *IEEE Transactions on Circuits and Systems for Video Technology*, Vol. 7, No. 4, August1997.

[2] G. Blakowski, and R. Steinmetz, "A Media Synchronization Survey: Reference Model, Specification and case studies," *IEEE Journal of Selected Areas in Communications*, Vol. 14, No. 1, January 1996.

[3] L. Chiariglione, "MPEG and Multimedia Communications," *IEEE Transactions on Circuit Systems for Video Technologies*, Vol. 7, No. 1, February 1997.

[4] A. Eleftheriadis, *et. al*, (ed), "Text for CD 14496-1 Systems (Source: MPEG-4 Systems)," *ISO/IEC JTC1/SC29/WG11 N1901*, January 1998.

[5] Draft ITU-T Recommendation H.263, "Video Coding for Narrow Telecommunication Channel at <64 Kbit/s," April 1995.

[6] Y. He, I. Ahmad, and M. L. Liou, "An implementation of MPEG-4 Video Verification Model Encoder Using Parallel Processing," *Proceeding of the Third Asia-Pacific Conference on Communications (APCC'97)*, Sydney, December 1997.

[7] D. Heyman and T. Lakshman, "Source Models for VBR Broadcast Video Traffic," *IEEE/ACM Transactions on Networking*, Vol. 4, No. 1, February 1996.

[8] ISO Committee Draft 13818-2, Generic Coding of Moving Pictures and Associated Audio: Recommendation H.262, ISO/IEC JTC1/SC29/WG11, November 1993.

[9] ISO/IEC, "MPEG-4 Proposal Package Description (PPD)," ISO/IEC JTC1/SC29/WG11 N0988, July 1995.

[10] R. Koenen (ed), "MPEG-4 Overview (Source: Requirements, Audio, DMIF, SNHC, Systems Video)," *ISO/IEC JTC1/SC29/WG11 N1909*, October 1997.

[11] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications," *Communication of the ACM*, Vol. 34, No. 4, April 1991.

[12] G. L. Li, J. H. Cui, B. Li and F. M. Li, "Transient QoS Guarantee for Packet Voice," to appear in *IEEE Globecom'98*, Sydney, Australia, November 1998.

[13] G. L. Li, J. H. Cui, F. M. Li and B. Li, "Transient Loss Performance of a Class of Finite Buffer Queueing Systems," *ACM Sigmetrics'98/Performance'98*, Madison, WI, June 1998.

[14] M. L. Liou, "Overview of the $p \times 64$ Video Coding Standard," *Communication of the ACM*, Vol. 34, No. 4, April 1991.

[15] T. D. C. Little and A. Ghafoor, "Synchronization and Storage Models for Multimedia Objects," *IEEE Journal on Selected Areas in Communications*, Vol. 8, No. 2 April 1990.

[16] E. Mellaney, L. Orozco-Barbosa and G. Gagnon, "Study of MPEG-2 Video Traffic in a Multimedia LAN/ATM Internetwork System," *IEEE Transactions on Circuit Systems for Video Technologies*, Vol. 7, No. 4, August 1997.

[17] M. Nilson, "VBR MPEG-2," *The ATM Forum Audiovisual Multimedia Services* BTD-SAA-AMS-VBRMPEG2-0 2.02. 1997.

[18] M. Pinedo, Scheduling: Theory, Algorithms and System, *Prentice Hall*, 1995.

[19] K. Shen and E. Delp, "A Spatial-Temporal Parallel Approach For Real-Time MPEG Video Compression," *Proceeding of 1996 International Conference on Parallel Processing*, vol. II, pp. 100-107, August 1996.

[20] T. Sikora, "The MPEG-4 Video Standard Verification Model," *IEEE Transactions on Circuit Systems for Video Technologies*, Vol. 7, No. 1, pp. 19-31, February 1997.

[21] J. M. Sohn, and G.Y. Kim, "Earliest-Deadline-First Scheduling on Nonpreemptive Real-Time Threads for a Continuous-Media Server," *Proceeding of High-Performance Computing and Networking*, 1997.