Journal on Multimedia Tools and Applications, Special Issue on Video on Demand,
Kluwer Academic Publishers, 1997

Columbia's VoD and Multimedia Research Testbed With Heterogeneous Network Support

Shih-Fu Chang, Alexandros Eleftheriadis, Dimitris Anastassiou
Stephen Jacobs, Hari Kalva, and Javier Zamora

Image and Advanced Television Laboratory
Department of Electrical Engineering
Columbia University, New York, NY 10027, USA

Abstract

This paper reports our progress in developing an advanced video-on-demand (VoD) testbed, which will be used to accommodate various multimedia research and applications such as Electronic News on Demand, Columbia's Video Course Network, and Digital Libraries. The testbed supports delivery of MPEG-2 audio/video stored as transport streams over various types of networks, *e.g.*, ATM, Ethernet, and wireless. Both software and hardware video encoders/decoders are used in the testbed. A real-time video pump and a distributed application control protocol (MPEG-2's DSM-CC) have been incorporated. Hardware decoders and set-tops are being used to test wide-area video interoperability. Our VoD testbed also provides an advanced platform for implementing proof-of-concept prototypes of related research. Our current research focus covers video transmission with heterogeneous quality-of-service (QoS) provision, variable bitrate (VBR) traffic modeling, VBR server scheduling, video over Internet, and video transmission over IP-ATM hybrid networks. An important aim is to enhance interoperability. Accommodation of practical multimedia applications and interoperability testing with external VoD systems has also been undertaken.

Keywords — Video on Demand, Interactive Video, Video Interoperability, Video Servers, MPEG-2 Video over ATM, Internet Video

1. Introduction

At Columbia University, we are developing a VoD testbed with advanced features of video storage, coding, manipulation, transmission, and retrieval. The main objective is to use this testbed as a platform for state-of-the-art multimedia research and application development. Among the potential applications are Columbia's Electronic News System, Digital Libraries, Interactive Video Courses on Demand, and other interactive multimedia applications.

Development of advanced VoD systems has been a prominent subject in research as well as commercial trials of broadband interactive audio/visual services in industry. Although some economic issues still exist and prevent large-scale deployment of VoD and interactive video services, designing a full-function VoD system for general multimedia applications remains a critical technical challenge and requires extensive interdisciplinary research. Much related work has been reported in the literature. The VoD workshop held at Columbia University [3] includes informative discussion participated by many researchers involved in testbed development, commercial trials, and new media applications. Several special issues in the literature also include comprehensive information about the state of the art technology [1, 2]. Specific technical approaches for individual components can be found in various fields as well. For example, dedicated storage architectures for real-time multi-access have been studied in [8, 9, 10]. Systematic approaches to the design of video servers (VS) are reported in [11, 12, 13]. Innovative methods for indexing/searching images by image contents were addressed in [4, 5, 6, 7]. In addition, many field trials of VoD services using proprietary high-performance VS technologies have made news headlines. Lastly, a major international forum, DAVIC, has been active in specifying standards for critical protocols and interfaces for achieving interoperability between various audio-visual applications [14]. In June 1996, we organized the first international interoperability event to test interoperability among clients and servers developed by different organizations. A total of eight organizations participated in the event [15, 16]. Description of the test process and result is included in another paper in this special issue [24].

Our testbed has several features which distinguishes it from other testbeds. The most important feature is the support for heterogeneous networks. Our testbed supports delivery of multimedia information over ATM, wireless, and Ethernet. Connected to these networks are different types of clients with different quality requirements. We have digital set-top boxes that are connected to ATM networks to handle high quality video, PCs/workstations with hardware/software decoders, and mobile terminals with hardware/ software decoding capabilities. The bitrate and quality of the delivered stream varies depending on the capabilities of the client. While hardware decoders connected to an ATM network can handle high quality video at very high bitrates, the software decoders and mobile clients have to settle for lower quality video at lower bitrates. This has influenced our server design. We have designed a server that supports all these different networks and quality requirements.

This paper provides details on the different components of our testbed, clients, server, networks, and content. Overall architecture and design goals are presented in Section 2. VoD clients are discussed in Section 3, server architecture and design issues in Section 4, network issues in Section 5, content encoding in Section 6, and finally related research activities are presented in Section 7.

2. Architecture and Design Considerations

Right from the outset our goal was to design a testbed that would serve as a vehicle to conduct research and supports different *classes* of users. Figure 1 shows the components of the testbed. Starting from the right, the end-users can be classified into 4 classes depending on their decoding capabilities and network connectivity. Class 1 users are located on campus and are connected to the campus Ethernet, class 2 users are located on campus and use mobile terminals, class 3 users are connected to the testbed via Internet, we classify them as remote users and class 4 users are connected via local or wide area ATM connections. Each of these classes require different QoS and are served by the same server. The wide area connectivity to GTE labs in the Boston area is used to understand the effects of video transmission over wide area networks.

Figure 2 shows the high level architecture of the testbed. The client shown can belong to any one of the 4 classes mentioned above. Each client maintains two connections to the server during a session. A bidirectional connection for user control and a unidirectional connection from server to the client for video delivery. The level 1 gateway functionality allows the clients to select a server before establishing a connection. When a client selects a server, it is connected to the application server which allows it to browse the directories available on the server and select videos. Example functions and components in the application server includes Service Gateway, Directory Service, and File Service. When a video is selected, the application server invokes a video-pump process which delivers the selected video.

The envisioned applications include Interactive News on Demand and Video Courses on Demand. These applications stress the importance of real-time interactivity and multi-user access efficiency. The research issues driven by these applications are briefly discussed later on in Section 7.

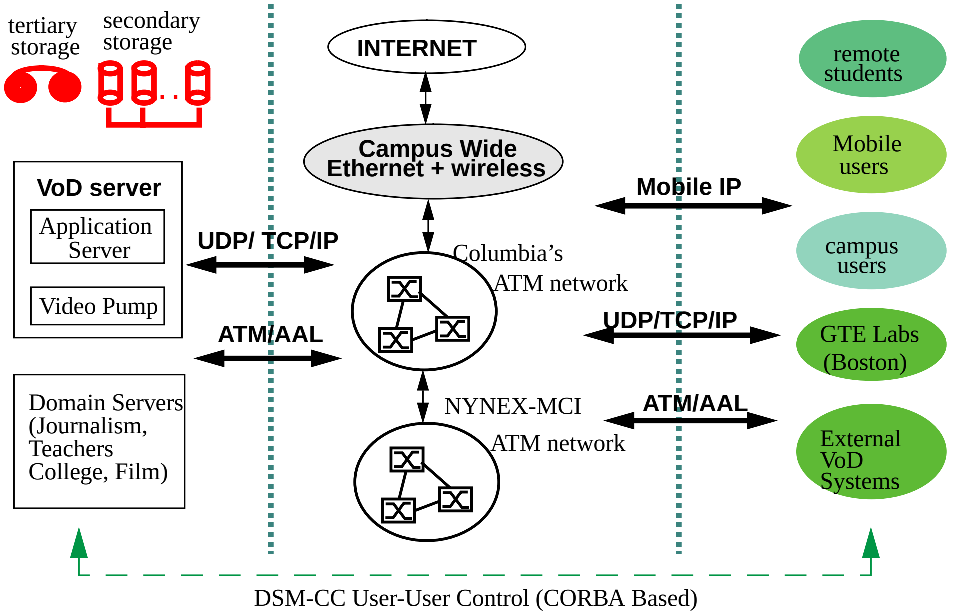

Figure 1. Components of Columbia's VoD Testbed

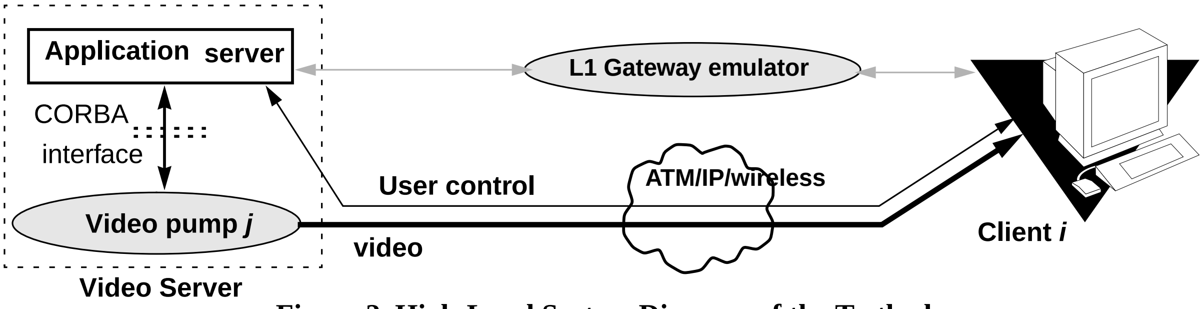

Figure 2. High-Level System Diagram of the Testbed

3. Client

The client-side platforms may be composed of workstations, PCs, mobile terminals, or stand-alone set-tops. Both MPEG-1 and MPEG-2 decoders are already available for PCs and consumer electronics products. ATM adapters on workstations and PCs are also on the brink of wide proliferation. Due to the highly asymmetric complexity in MPEG coding and multiplexing, the client side capability will not be a bottleneck in the end-to-end real-time data pumping chain. There are still some design issues, however, such as the trade-off between decoder complexity and scalable video coding, and the trade-off between the client-side buffer space and stream playback interactivity. These issues need to be investigated from a global, system-level perspective.

We have implemented software-based MPEG-2 decoder and playback routines with VCR interactive control functions on several workstation platforms. Hardware decoders (both MPEG-1 and MPEG-2 transport) are available for attachment to these general-purpose platforms. The mobile terminals are based on notebook computers with wireless Ethernet adapters and MPEG-2 decoders. Dedicated hardware set-tops with specific network interface modules are currently provided by external partners. In order to maintain maximal flexibility and extensibility, we focus on the general-purpose computing platforms at this stage. Thus, we do not require the use of real-time operating systems on the client side. As mentioned above, using the hardware transport stream decoders will reduce the processing load of the client computer and therefore may avoid the need of real-time operating systems.

To make the access simple and have a uniform user interface for all platforms, we have developed web-browser based client interfaces. Figure 3 shows the control and data flow paths between a server and a client with a web-browser interface. A user connects to a server by accessing a web page with a list of all the available servers. When a server is selected, the server ID is passed to the DSMCC interpreter by the command dispatcher. The DSMCC interpreter listens on a TCP port which does not change during a session. The information returned by the server, *e.g.*, in response to the `directory::list` operation, is passed back to the client. When a video playback is requested, the video is delivered to the client directly by the server through the video pump.

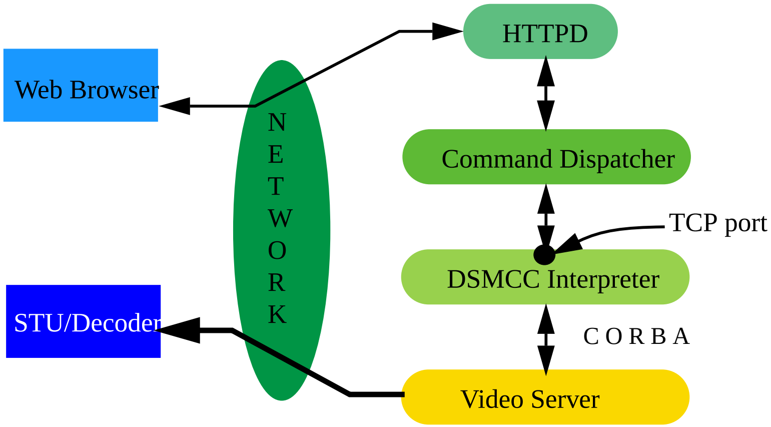

Figure 3. Control and Data Flow for Clients with Web-browser Interfaces

For user interaction with the server we have adopted DSMCC [17] user-user primitives. DSMCC UU primitives allow a user to connect to a server, browse a server's directory hierarchy, select a service, make queries, and play videos interactively. We have implemented a subset of these primitives known as *core interfaces* that allow the users to connect to a server, browse directories, and play videos.

4. Server

Optimization of the overall VoD system performance requires a balanced system approach in exploring all the critical design factors for the video server. Fundamentally, it's a real-time data pumping problem — how to store massive video streams in a hierarchical storage unit (including memory, disk, tape, and tertiary storage), move them through the I/O interface and memory, and then pump them to the network interface. Careful data layout within the storage hierarchy, efficient real-time scheduling, and admission control mechanisms are all required in optimizing the system performance. We are investigating all these research issues in designing our video server.

Our server platform currently includes an SGI Onyx multiprocessor graphics computer as a super-server (with 6 CPUs and 1GB of memory), and clusters of workstations as distributed servers. The Onyx super-server is equipped with the high-end computing power and 3D-graphics capabilities that are needed in many interactive multimedia applications and real-time video manipulations. Dedicated disk array secondary storage is connected to the server, while local storage systems are available on distributed workstations. The server's communication interface is enhanced by the connection to an ATM LAN, which in turn is connected to an external ATM WAN.

The main components of the video server include a video pump for real-time CBR/VBR video stream retrieval, and an application server high-level control/management entity as shown in Figure 2. The former is responsible for retrieving the video stream from the storage unit to the network and guarantee real-time performance. Our video pump is a generic design with support for different network interfaces (e.g., TCP/

UDP/IP and ATM) and video types (e.g., MPEG-1 and MPEG-2). We take advantage of the multiprocessor architecture and the real-time process scheduling control of the Onyx machine to achieve real-time guarantees at a certain temporal granularity (e.g., 50 µs) [18]. For typical MPEG-2 video rates and ATM Service Data Unit (SDU) sizes, the temporal resolution at this level is sufficient.

The application server allows a user to select content and is responsible for forwarding the playback control commands such as pause and resume to the video pump. The application server and the video pump are two different software objects with a well-defined CORBA interface. This interface allows the application server to launch the video pump on a remote machine thus supporting a distributed video pump. When ever a user selects to play a video stream, a new video pump process is launched. The video pump architecture is shown in Figure 4. Each video pump object consists of three separate threads, one for control, reading and interpreting the commands from the application server, one for reading video data from the disk and one for writing data to the network. In case of our video server, the reader and writer threads run on separate processors on the multiprocessor SGI-ONYX.

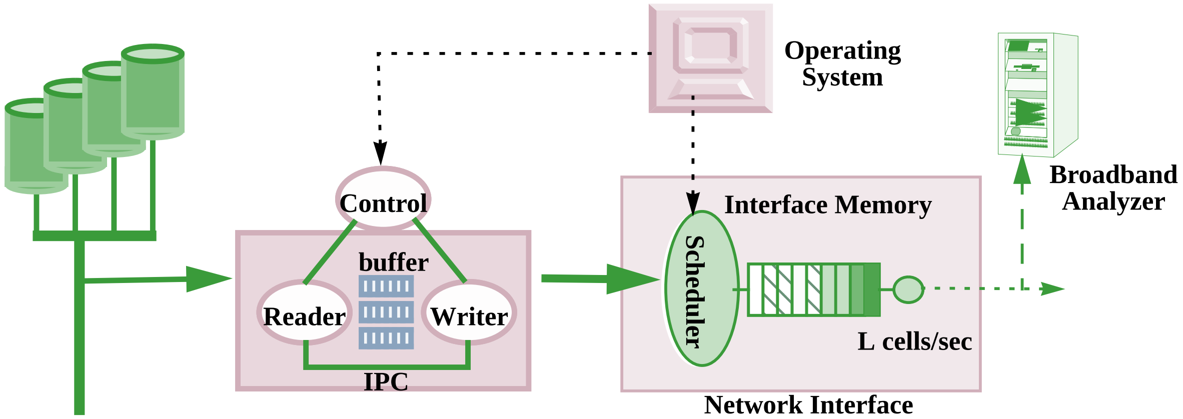

Figure 4. Video Pump Architecture

The video data is stored on the disks in the form of multiplexed MPEG-2 transport streams. The video pump reads this data and schedules the video packets for writing to the network interface. The scheduling schemes vary depending on the network types. In the case of ATM, two MPEG-2 transport stream packets are mapped to 8 ATM cells (AAL 5). The shorter PDU size significantly reduces the performance of the video pump. However an increase in the PDU size will result in an increase in PCR jitter. This increases the buffer size on the client to do de-jittering. This is a design trade-off which affects only the cost of hardware decoders. In the case of software decoders, the buffer size can be increased and hence the PDU size without affecting the end quality. The video server adjusts the PDU size depending on the client so as to keep the performance as high as possible.

5. Network

Our testbed is composed of different types of networks, ATM and IP. The ATM network is composed of four ATM switches and is connected to the ATM networks on medical, and Earth Science campuses. The network is connected to the external testbeds via NYNEX ATM network. We had a wide area connection to GTE labs during July-August 1996. This connection was used to study the effect of wide area networks and public ATM switches on the quality of video delivered. Studies were done specifically related to the measurement of cell delay variation and the effect of cell delay and cell loss on the perceived quality of the delivered video. All the measurements were made with the HP Broadband Analyzer and the network impairment module. In the case of the ATM network, the bidirectional control channel uses IP over ATM and video is delivered over ATM with AAL 5 adaptation.

The IP network is composed of Ethernet and wireless. The wireless network is composed of several wired-to-wireless gateways and supports mobile IP. The mobile terminals use a wireless Ethernet adapter with a maximum throughput of 2 Mbps. But video streams of up to 700 Kbps are used in mobile terminals as quality degrades at higher rates. For the user control information, TCP is used while UDP is used to deliver video to clients. The software decoders running on PCs can decode streams of up to 600 Kbps in real-time. To deliver video information at different rates for different classes of clients, a dynamic rate shaping (DRS) strategy is used [19]. DRS allows the server to reduce the bitrate of the video stream accordingly depending on the type of the network and the network conditions.

6. Content Generation

All visual material is stored on the video server in a compressed form (e.g., MPEG-2 transport streams and MPEG-1 system streams). In order to reduce the time spent in content preparation, it is desirable to have fast encoding, packetization, and multiplexing tools. For live video production, real-time encoding facilities are required. We use an MPEG-2 hardware encoder (C-Cube) to compress audio and video. To multiplex the audio and video we rely on our software. We also use our software encoder to generate scalable MPEG-2 streams. Our software encoder and transport stream multiplexer currently run at a speeds lower than real time. The rapid advancement of today’s computing technology, however, will hopefully resolve this problem soon. We should note that, for stored-video applications such as video on demand, an off-line non-real-time video coding facility is generally sufficient. For MPEG-1 video compression, a real-time hardware-based system hosted on a PC is utilized, generating MPEG-1 system streams (including both video and audio).

We have developed flexible software facilities for digitizing and encoding video from various sources (e.g., live camera, VCR, and LD player) and various domains (e.g., recorded lecture video, general movies, and test video from the public domain). Columbia’s MPEG-2 software encoder (a full implementation, including all scalability profiles) is utilized to generate scalable MPEG-2 video streams while the MPEG-2 mux/demux software is used to packetize the elementary streams, and multiplex multiple packetized elementary streams (PES) to MPEG-2 transport streams. Timing information is captured by time stamps at different layers of the compressed bit streams, such as decoding time stamps, presentation time stamps, and program clock references, in order to maintain synchronization at different levels. Multimedia data such as text, graphics, and audio may be multiplexed into the transport streams as well, as described in the MPEG-2 system standard.

The bit rate allocation of each video stream depends on the type of the video source. In addition to the constant bit rate MPEG-2 main profile video streams, we also use hybrid scalable MPEG-2 coding to generate scalable video streams. We combine the spatial scalability and SNR scalability modes to produce three different layers of video. The base layer has a small spatial size and is suitable for video browsing and preview functionalities. The first enhancement layer increases the spatial size and keeps the video signal quality (i.e., SNR values) at a consistent level. The second enhancement layer improves the signal quality as well as the spatial resolution. Based on our preliminary and subjective evaluations, the base and first enhancement layers provide a subjective quality comparable to VHS video quality, while the highest layer provides a subjective quality comparable to that of a LaserDisc. The decision of whether to use scalable video coding or not should be based on the application types, system capacity, and encoder/decoder capabilities. There are currently no commercial scalable MPEG-2 hardware decoders available; scalability, however, is desirable in heterogeneous environments including different networks (wired and wireless), different client processing/display capabilities, and different user preferences.

7. Research Issues

In addition to accommodating development of advanced multimedia applications, Columbia’s VoD testbed also serves as an experimental environment for implementing proof-of-concept prototypes for advanced engineering research. The availability of an end-to-end comprehensive testbed is actually very critical to many research projects which have cross-disciplinary nature. For example, optimization of the video server storage unit cannot be isolated from research on video transmission over networks. There are strong interactions between the server scheduling/buffering mechanisms and network transport mechanisms. These interactions are best understood in an actual experimental testbed covering end-to-end components. We discuss several major research areas highly related to the VoD systems in the following sections.

7.1. QoS guarantees in VoD Services

A key issue in a VoD service and in any video service is to provide an acceptable Quality of Service (QoS) to the end-user. This QoS is a function of many parameters, such as the frame loss frequency, blockiness effects, audio and video synchronization (lip synchronization), chroma stability (i.e. in NTSC display systems). Some of these parameters are not easily quantifiable since they depend on the subjective perception of the viewer. The QoS at the video client reflects how the original video stream has been delivered from the remote video server, where concepts of semantic transparency and time transparency characterize the performance of video services over networks. Such services require specific constraints regarding the delay, specifically the delay variation or jitter, experienced across the connection, as well as constraints regarding the rate of errors acceptable in order to have a guaranteed end-to-end QoS, from video server to video client. Therefore, an important question for VoD is how to map a specific QoS required in the video client into a QoS specification for the video server and network.

To answer this question, we have developed a novel and simple approach for mapping QoS from video server to video client, by using a generic model which can incorporate any VoD system [18]. We then provide an example of this methodology by applying it to the Columbia VoD testbed. The goal of our model is to provide an end-to-end QoS. In other words, we are examining how impairments in the video server, network and video client affect the quality of the video perceived by the end user. Since VoD is a point-to-point service, we provide QoS guarantees per individual stream rather than over an aggregate video traffic in the video server or network switch. Using this methodology we then derive an admissible region of traffic load in the video server which guarantees QoS end-to-end.

7.2 VBR Video Retrieval Scheduling

Variable-bit-rate (VBR) video usually allows for better trade-off between video quality and network bandwidth. However it also demands more intelligent scheduling algorithms to guarantee the quality of service. In this research we investigate techniques to enhance our VoD testbed with video servers supporting both constant bit rate (CBR) and VBR *scalable* video. Our goal is to support heterogeneous clients and video interoperability.

In a generic model, the video server has a disk retrieval scheduler and a network stream scheduler. The disk retrieval scheduler determines how video data for multiple, concurrent video streams are to be transferred from the disk arrays to the memory buffers of each storage computer. The network stream scheduler determines how video data is to be transmitted from the memory buffers into the ATM switch for transmission to the clients. In our research, we found that the video server disk retrieval schedulers are critical in maximizing resource utilization for the retrieval of VBR video [25]. Note that the disk retrieval schedulers are different from disk head schedulers and operate over different time cycles.

Currently, we are investigating the network stream scheduler mechanism and also the interaction between the disk retrieval scheduler and the network stream scheduler. The network stream scheduler determines the transmission of stored VBR video over the ATM network. There is a fundamental difference between the transmission of stored VBR video for VoD applications and the transmission of real time VBR video for broadcast and conference applications. In our research we are maximizing the advantage of having a prior knowledge of the data trace of stored video to optimize the network stream scheduling.

In order to achieve maximal interoperability, we are investigating issues of scheduling real-time retrieval of scalable MPEG-2 video. We have developed a progressive display scheme, in which the video server supports different degrees of interactivity for the different layers of video.

7.3 Internet Video

One of our current research effort is to enhance the interactive video service over Internet. Most techniques for developing video services in networks without QoS, i.e. Internet, involves an explicit attempt at avoiding network congestion. Clearly, network congestion hurts the performance for all users of the network. The goal is to send only the data that can fit into the network at a particular time. This requires an estimate of the bandwidth in the network. Much work has been done in the past regarding bandwidth estimation [20, 21]. However, providing good quality networked video requires innovative solutions from both the networking and image processing perspectives. Past attempts at solving the problem of video services over non-QoS networks have tried to isolate the problem as being purely networking. This has resulted in crude techniques, such as frame dropping, for forcing high bandwidth video streams through small bit pipes [22].

We are currently implementing a solution based on this dual-perspective approach. Namely, we are concerned with both the difficulties of bandwidth estimation as well as developing a methodology for shaping compressed MPEG-2 streams to a continuum of possible bandwidths. This technique is called Dynamic Rate Shaping (DRS) [19]. In its simplest form, DRS selectively drops coefficients from the bit stream which are least important in terms of image quality. This gives us the ability to dynamically change the bit rate of a pre-compressed stream.

7.4 Video Transmission Over IP-ATM Hybrid Networks

Another approach we take for developing advanced Internet video service deals with a realistic, hybrid network environment including both ATM and IP networks. Recent research on Internet audio and video service has resulted in a draft Internet standard for transporting real-time data known as Real Time Transport Protocol (RTP) [23].

We are using RTP and RTCP, the control protocol of RTP, to study quality of service issues in video transmission over hybrid networks. The RTP protocol has features to report packet loss and packet jitter to the source. This information in-turn will be used by the source to regulate the transmission rate by source flow control or source rate shaping described above. Another important issue is mapping of MPEG payload. A payload mapping should consider the packet loss inherent in the IP networks and support suitable error concealment and recovery schemes.

8. Conclusion

We have developed a state-of-the-art VoD testbed following a design approach that allows to have flexible components, while at the same time adapting to emerging open standards such as DAVIC. Our testbed was the central component of the global interoperability event organized here at Columbia University. With the campus-wide multimedia applications and video interoperability tests as the initial driving forces, we are building a VoD testbed with advanced capabilities of audiovisual representation/storage/retrieval/transmission. The testbed serves as an advanced prototyping platform for both engineering research and practical applications. It also facilitates the interaction between engineering researchers and application practitioners. Many practical applications are also being developed in this VoD testbed, including Columbia’s Interactive Electronic News Experiment.

Our testbed has an entire spectrum of clients and networks and is very flexible to implement and validate new methodologies, algorithms, and applications. On the research side, our focus at this point covers innovative video server design, content-based video traffic modeling, heterogeneous QoS provision through multi-resolution coding and dynamic rate shaping algorithms, innovative compressed-domain video manipulation, and packet video transmission over ATM and IP networks.

Acknowledgments

We would like to acknowledge generous support for our testbed infrastructure from C-Cube, GTE Labs, HP-Idacom, IBM, NEC, NYNEX, and SGI.

The wide-area VoD interoperability test was completed jointly with the research group led by Dr. Leonard Ulbricht of GTE Labs. The DAVIC interoperability test was a collaborative effort with other participants described in [16].

We would like to acknowledge Xi Chen, Seungyup Paek, Paul Bocheck and other people involved in the Columbia VoD effort for their contributions to the testbed.

References

[1] Special Issue on Video on Demand, International Journal of Image Communication — Signal Processing, Vol. 8, No. 3, April 1996.

[2] Special Issue on Video on Demand, IEEE Communications Magazine, May 1994, Vol. 32, No. 5.

[3] Workshop on Video on Demand Systems - Technology, Interoperability, and Trials, Columbia University, June 1996. (*http://www.itnm.columbia.edu/workshop/vod96*)

[4] A. G. Hauptmann and M. Smith, "Text, Speech and Vision for Video Segmentation: The Informedia Project," AAAI Fall Symposium, Computational Models for Integrating Language and Vision, Boston, November 10-12, 1995.

[5] M. Flickner, *et al*, "Query by Image and Video Content: The QBIC System," IEEE Computer Magazine, Sep. 1995, Vol.28, No.9, pp. 23-32.

[6] J. Meng and S.-F. Chang, "CVEPS: A Compressed Video Editing and Parsing System," ACM Multimedia Conference, Boston, MA, Nov. 1996.

[7] H. Zhang, C.Y. Low, and S. Smoliar, "Video Parsing and Browsing Using Compressed Data," J. of Multimedia Tools and Applications, Vol. 1, No.1, Kluwer Academic Publishers, March 1995, pp.89-111.

[8] H. M. Vin, P. V. Rangan, "Designing a Multi-User HDTV Storage Server", IEEE Journal on Selected Areas in Communications vol. 11, No. 1, January 1993.

[9] K. Keeton and R. Katz, "The Evaluation of Video Layout Strategies on a High-Bandwidth File Server," Intern. Workshop on Network and Operating System Support for Digital Audio and Video, Lancaster, England, UK, Nov. 1993.

[10] E. Chang and A. Zakhor, "Scalable Video Data Placement on Parallel Disk Arrays," SPIE Symposium on Imaging Technology, San Jose, 1994.

[11] M. Vernick, C. Venkatramini and T.-C. Chiueh, "Adventures in Building the Stony Brook Video Server," ACM Multimedia Conference, Nov. 1996, Boston, MA.

[12] C. Federighi and L.A. Rowe, "The Design and Implementation of the UCB Distributed Video On Demand System," Proc. of IS&T/SPIE 1994 Int'l Symp. on Elec. Imaging: Science and Technology, San Jose, CA, February 1994.

[13] T.D.C. Little, G. Ahanger, R.J. Folz, J.F. Gibbon, W.W. Reeve, D.H. Schelleng, and D. Venkatesh, "A Digital On-Demand Video Service Supporting Content-Based Queries," ACM 1st Multimedia Conference, Anaheim, CA, Aug. 1993.

[14]DAVIC 1.0 Specification, The Digital Audio-Visual Council (DAVIC) 1995.

[15]Columbia University, “DAVIC New York Interoperability Experiments: Report and Results,” DAVIC Contribution No. DAVIC/TC/SYS/96/09/008, Geneva, Switzerland, 1996.

[16]DAVIC VoD Interoperability Test Event, June 1996, Columbia University, New York. (*http://www.itnm.columbia.edu/vod/interop96*)

[17]Generic Coding of Moving Pictures and Associated Audio (DSMCC), ISO/IEC 13818-6, International Standard, 1996.

[18]J. Zamora, S. Jacobs, A. Eleftheriadis, S.-F. Chang and D. Anastassiou, “A Practical Methodology for Guaranteeing QoS for Video on Demand,” submitted for publication, April 1996.

[19]A. Eleftheriadis and D. Anastassiou, “Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video,” Proceedings, 5th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV-95), Durham, New Hampshire, April 1995, pp. 95-106.

[20]I. Busse, B. Deffner, and H. Schulzrinne, “Dynamic QoS control of multimedia applications based on RTP,” in First International Workshop on High Speed Networks and Open Distributed Platforms, (St. Petersburg, Russia), June 1995.

[21]T. Sakatani, “Congestion avoidance for video over IP networks,” Multimedia Transport and Teleservices Proceedings, 13-15 Nov. 1994. pp. 256-273.

[22]Z. Chen, S. M. Tan, R. H. Campbell and Y. Li, “Real Time Video and Audio in the World Wide Web,” World Wide Web Journal, Volume 1, January 1996.

[23]H. Schulzrinne, S. Casner, and R. Frederick, “RTP: A transport protocol for real-time applications,” RFC 1889, Internet Engineering Task Force, Jan. 1996

[24]H. Kalva, S.-F. Chang, and A. Eleftheriadis “DAVIC and Interoperability Experiments,” International Journal on Multimedia Tools and Applications, Kluwer Academic Publishers, 1997.

[25]S. Paek and S.-F. Chang, “Video Server Retrieval Scheduling for VBR Scalable Video”, Third IEEE International Conference on Multimedia Computing & Systems, Japan, 1996.