Access to a DAVIC-Based Video on Demand System Using the World Wide Web

Alex de Jong
Information Technology Laboratory, NIST
Gaithersburg, MD 20899, U.S.
alex.dejong@nist.gov

Taewoon Kang
Electronics and Telecommunications Research Institute
Yusong-Gu, Taejon, 305-350, Korea
twkang@winky.etri.re.kr

December 13, 1996
This paper has been accepted for the 11-th International Conference On Information Networks, Taipei, Taiwan, January 27– 29, 1997.

Abstract

This paper describes the development of a simple access mechanism of a DAVIC-based VoD server, using a WWW browser. A user can access the server simply by selecting the server's Web address via a Web browser. A menu enables the user to select VCR-like buttons and send them via HTTP forms to the server. The VoD server invokes a CORBA based DSM-CC implementation to actually perform the requested operations.

1 Introduction

The Digital Audio-Visual Council (DAVIC) [1] has adopted a set of standards to define a framework for digital audio and video services. This framework defines a wide range of digital video applications with a variety of communication bandwidth needs.

The first application specified by DAVIC is Video on Demand (VoD). In order to do so, DAVIC has adopted standards and specifications from several standards bodies which include the ATM Forum Audiovisual Multimedia Services (AMS)[2] specifications, and the Data Storage Media Control and Command (DSM-CC)[3] specifications from the Moving Pictures Experts Group (MPEG) standards. The former defines the MPEG-to-ATM mapping. The latter includes the specification for the control functions to access a VoD server. The DSM-CC architecture uses a Common Object Request Broker Architecture (CORBA) [4] as defined by the Object Management Group (OMG) to implement Remote Procedure Calls (RPCs).

In parallel to the DAVIC activities, the Internet has been expanding exponentially, largely due to the popularity of the World Wide Web (WWW). Web browsers and the HyperText Markup Language (HTML) [5] make it easy to access and publish information on the Web via the Internet. This paper presents a DAVIC based VoD server which allows access via the Web, taking advantage of the wide availability of Web browsers. VoD access via the Web represents an ideal means to provide users with access to the more advanced DAVIC services while DAVIC networks are being developed.

The paper starts with a brief discussion of the DAVIC model for a VoD service. Section 3 presents the extension to the DAVIC model for Web access. This section is followed by a complete VoD system architecture which is implemented at the National Institute of Standards and Technology (NIST). The prototype implementation details are described in section 5. Section 6 presents related research and ongoing work at NIST.

2 The DAVIC System

A general model of a DAVIC compliant VoD server is shown in figure 1. DAVIC defines a system in terms of information flows (e.g. S1, S2) and reference points (e.g. A0, A1). Only DAVIC defined information flows S1 and S2 are shown, since they are responsible for transfer of the MPEG content and the User-to-User (U-U) DSM-CC commands, respectively. Other information flows deal mainly with User-to-Network (U-N) related issues which are not our primary interest. We assume that the user already has access to the Internet via a Web browser. This means that some kind of U-N relation already exists in order to enable Internet traffic.

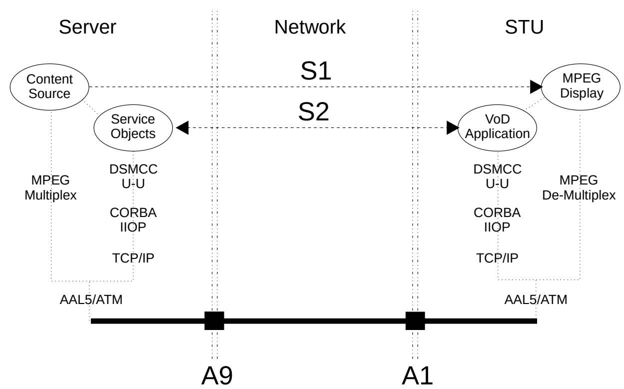

Figure 1: DAVIC VoD Model

The DAVIC server consists of 2 main components that are built upon a communication protocol stack: the *Service Objects* and the *Content Source*. The former consists of the DSM-CC service objects such as *Session*, *Directory*, *File*, and *Stream* services. These services provide the basic building blocks for applications to establish a session and browse the server's contents. The *Content Source* of the server in general consists of MPEG 2 Transport Streams (TSs).

A simple scenario between the server and a SetTop-Unit (STU) to play a movie consists of a sequence of service object invocations. After a session is established, user information– such as menus and graphics– is retrieved from the server via directory and file services. This information is exchanged via DAVIC's S2 flow. After a reference to the requested stream (i.e. movie) object is obtained, a stream *Play* request can be made. This request will be followed by the establishment of an S1 flow to transport MPEG data to the STU.

The DAVIC S2 flow protocol stack can be built upon a variety of underlying physical network architectures. Asynchronous Transfer Mode (ATM) provides the flexible bandwidth over the physical link, while ATM Adaptation Layer 5 (AAL5) together with Classical IP provide the network layer. The higher layers include the DSM-CC U-U and CORBA 2.0 specifications. The S1 flow uses the same lower layer as the S2 flow, but MPEG 2 TS packets are directly mapped to AAL5/ATM [2].

3 WWW Access Extension

The scenario explained in the previous section demonstrates how a DAVIC compliant STU can access a DAVIC VoD server. S1 flows carry MPEG data which can be based on pre-established Permanent Virtual Circuits (PVCs) or Switched Virtual Circuits (SVCs). The former eliminates the need for an interoperable signalling stack. Control information is exchanged via the S2 flow which is implemented by a CORBA-based DSM-CC U-U protocol stack and Classical IP. While interoperability problems may occur at the IP level, interoperability problems predominantly occur between CORBA implementations. Interoperability between implementations is still not guaranteed and most STUs have not implemented a compliant S2 flow yet.

Internet Web browsers provide an easy-to-use and flexible tool to display text and/or graphical information. The underlying language and communication protocol, HTML and the HyperText Transfer Protocol (HTTP) respectively, are powerful as well as flexible for the developer. Using a Web browser to access a VoD server to browse available movies and select them for viewing provides a simple access mechanism, assuming the STU (or a workstation in combination with the STU) can use a Web browser.

Figure 2 shows the DAVIC VoD model with a Web access extension. The *Content Source* and *Service objects* are identical to figure 1. A *WWW-DSM Access* object is added to handle user requests from the Web browser.

The WWW-DSM object receives user commands encapsulated in HTTP forms. After initial configuration of the DSM-CC (e.g. getting access to the DSM-CC, setting the proper directories, etc.), it converts user requests into DSM-CC commands. The DSM-CC server receives the converted commands and processes them as if they came from a compliant DAVIC STU. Note that the DAVIC reference points and information flows have been omitted from the figure since

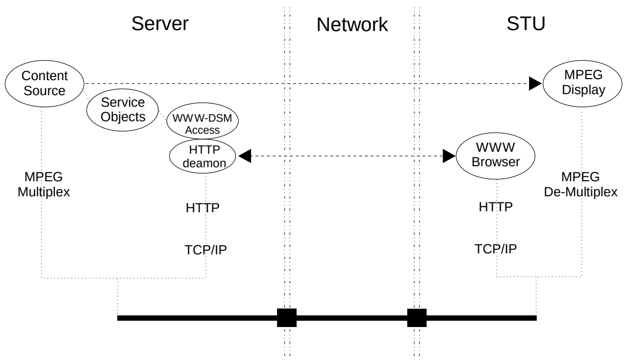

Figure 2: WWW Access Extension

they do not conform to the DAVIC specifications anymore. However, the structure of the model fits directly into the DAVIC model.

4 System Architecture

The system architecture shown in figure 3 follows the model presented in the previous section. The server consists of 4 components: the HTTP daemon, WWW-DSM Access, and the DSM-CC itself. The STU has 2 components: a Web browser and an MPEG decoding/display component.

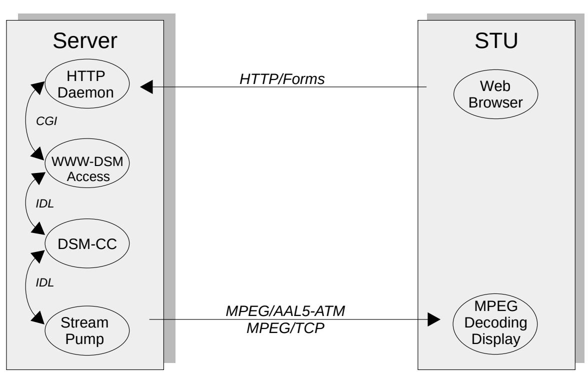

Figure 3: System Architecture

The HTTP daemon accepts requests from Web browsers to access information on the server. This type of daemon runs continuously to handle requests. After selection of a movie in the VoD Web page, a form is sent to the server which is processed by the WWW-DSM object. Communication between the WWW-DSM and the HTTP daemon is via the Common Gateway Interface (CGI). The CGI provides a means of enabling scripts to handle incoming forms. Gateway scripts process incoming forms and pass the information to the WWW-DSM object.

WWW-DSM incorporates the client part of the CORBA based DSM-CC, while the DSM-CC itself incorporates the server part. It is responsible for demultiplexing incoming requests (from multiple Web browsers) and converting them into the appropriate DSM-CC requests. Because of the strict performance requirements (See [6]) for streaming MPEG data to an STU, a separate stream pump component has been included. The stream pump, controlled by the DSM-CC, is capable of streaming multiple streams to multiple STUs . The DSM-CC and the stream pump interfaces are specified by the Interface Definition Language (IDL). The DSM-CC IDL complies with the DSM-CC standard [3]. The stream pump IDL is proprietary, but resembles the DSM-CC *Stream* IDL.

5 Implementation

NIST has implemented the Web access extension as shown in the previous sections. Figure 4 shows NIST's VoD server Web page, available online with the Universal Resource Locator (URL): *http://mace.ncsl.nist.gov/vod/histvod.html*. The page is split into an upper and lower frame; the upper frame for general information, the lower for user interaction when viewing a movie. A *Play* button allows the user to start playing the current selected movie. Other buttons allow VCR-like playback functionality such as *Pause* and *Stop*. A *Skip* option allows the user to skip 60 seconds in the movie. More options, such as *Fast Forward* and *Rewind* are under development.

Perl scripts [7] implement the CGI to WWW-DSM interface. The scripts process incoming forms and pass commands via a simple Inter-Process Communication (IPC) mechanism implementation, such as Berkely Sockets [11], to the WWW-DSM object. WWW-DSM activates the appropriate DSM-CC objects when a new *Play* request is received. This normally includes activation of the *Session* service to start a new session and the *Directory/Stream* services to find and open the appropriate *Stream* object. Once a reference to a *Stream* object has been obtained, commands such as *Play*, *Pause*, etc. can be exchanged directly with the object. The *Reset* button shown in figure 4 resets the stream object and terminates the current session.

The DAVIC-based VoD server, implemented on a UNIX-based workstation, uses a CORBA 2.0 com-

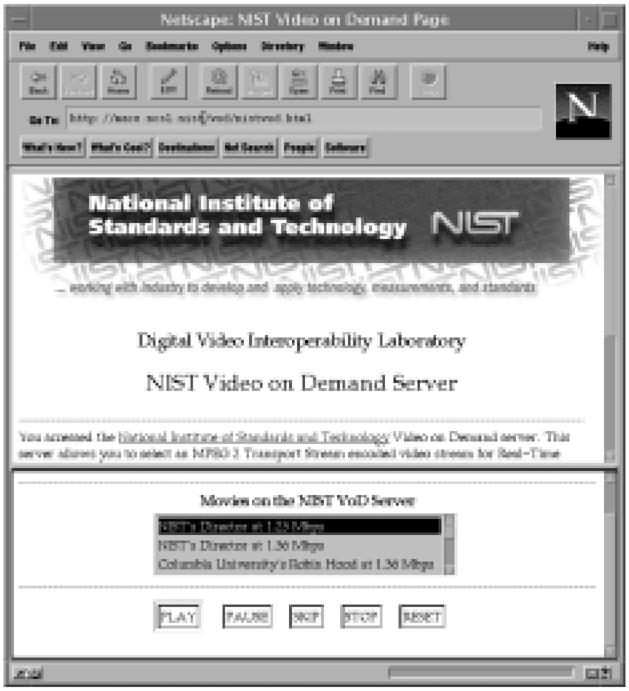

Figure 4: NIST VoD Web Page

pliant DSM-CC implementation [1]. As mentioned in the previous section, the stream pump is a separate component in the VoD server. A distributed stream pump architecture enables the use of a heterogeneous set of UNIX-based workstations to deliver Constant Bit Rate (CBR) MPEG data to STUs. A multiple stream pump controller automatically invokes multiple workstations when more STUs request the reception of an MPEG stream. The controller normally resides together with the DSM-CC on a single workstation, while stream pumps can be activated on a number of workstations. The controller maintains a registry of available workstations and their capacity to distribute requests properly.

Two experimental STUs demonstrate access to the VoD server with a Web browser. A workstation based STU uses a software MPEG decoder/display while a PC based STU uses a hardware decoder [2]. Upon selection of a movie, a connection is established between the VoD server (or more precise, the stream pump) and the STU to carry the MPEG stream. The connection is established either by invoking the appropriate signalling or by using a pre-established channel, such as an ATM PVC. Transport protocols supported include AAL5/ATM[2] and the Internet protocols, TCP/IP and UDP/IP.

The first prototype of the NIST VoD server with Web access has been was successfully demonstrated at the DAVIC Interoperability Event, at Columbia University, New York City, USA, June 1996. NIST's server enabled participants to access its VoD server using a Web browser. Simple VCR commands, such as *Play*, *Pause*, and *Stop* could be exchanged via HTTP forms for processing on NIST's VoD server.

6 Related Research

While the Internet and WWW have great opportunities, they also have limitations. The Internet transport protocol, TCP/IP, can currently not guarantee bandwidth requirements, which are especially important when transporting time sensitive data, such as audio and video. HTTP forms are powerful to specify keywords for search queries and other relatively simple commands, but prove insufficient for more advanced applications such as the VoD applications envisioned by DAVIC.

Research on the ReSerVation Protocol (RSVP) and smart compression algorithms are promising technologies to enable time critical applications to use the Internet. However, a network that supports bandwidth guarantees, such as envisioned by DAVIC, is critical to enable an S1 flow to the customer's premises.

Java [8] can provide the other part of the solution. Its Object-Oriented (O-O) character matches the CORBA-based DSM-CC, while most common Web browsers provide a Java run-time environment. Products are already available that enable DSM-CC implementations in Java. Since Java is platform inde-

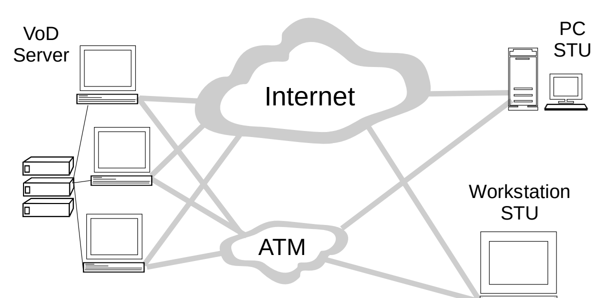

Figure 5: NIST Prototype Testbed

[1] The implementation of NIST's testbed is a joined effort between NIST and Korean Telecom Research Group.

[2] Integration of the hardware decoder with NIST's testbed is under development.

pendent, one single Java program can run on a variety of platforms. Implementing the STU component of the DSM-CC in Java has the potential to allow access via a Web browser while maintaining compliance with DAVIC's S2 flow.

A paper, "A VoD Application Implemented in Java" [9], discusses a complete Java based approach. This means that in addition to the DSM-CC implemented in Java, all user menus and interfaces to browse the VoD server's content are implemented by Java objects, too (versus MHEG-5 objects [10]). A prototype is being developed at NIST to demonstrate feasibility of this approach.

7 Conclusions

This paper describes a simple extension to a DAVIC-based VoD system to enable access by Web browsers. The extension uses HTTP forms to send requests to the server. Processing of the requests is performed via a WWW-DSM object, using HTTP's CGI. The main purpose of this extension was to enable S1 flow testing between systems with incompatible S2 flows. Since Web browsers are widely available for any platform and provide facilities to process user requests, the mechanism is low-cost and simple. A prototype tesbed, which includes a VoD server and STUs, has been developed at NIST which demonstrates the feasibility of the approach.

Enabling a Web access extension should neither be seen as a replacement for, nor an alternative to, using a DAVIC-compliant S2 flow. The S2 flow, implemented with a CORBA based DSM-CC, is more flexible and powerful compared to the relatively limited and simple forms exchange via HTTP. However, in the interim, while DAVIC compliant systems are scarce and Internet capable systems are widely available, it provides a simple and low-cost extension to access a DAVIC-based VoD server with an STU that does not support compliant DSM-CC capabilities.

References

[1] Digital Audio-Visual Council, DAVIC Specifications 1.0, December 1995.

[2] ATM Forum, Audiovisual Multimedia Services: Video on Demand Specifications 1.0, ATM Forum, January 1996.

[3] ISO/IEC, Information Technology– Generic Coding of moving pictures and associated audio information, Digital Storage Media Command & Control, International Standard, ISO/IEC 13818-6, August 1996.

[4] R. Otte, O.Patrick, and M. Roy, *Understanding CORBA: the common object request broker architecture*, Prentice Hall, 1996.

[5] D. Scharf, *HTML Visual Quick Reference*, Que Corporation, 1995.

[6] S. Jacobs and A. Eleftheriadis, "Video Pump Design for Interoperability With Set Top Units: the case against small PDUs". Proceedings of *8th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV)*, Zushi, Japan, April 1996, pp. 123–129.

[7] L. Wall, T. Christiansen, R.L. Schwartz, *Programming Perl*, O'Reilly & Associates, Inc., 1996.

[8] G. Cornell and C.S. Horstmann, *Core Java*, SunSoft, Sun Microsystems, Prentice Hall, 1996.

[9] A. de Jong, K. Hsing, and D. Su, "A VoD Application Implemented in Java". To be published in: *Multimedia Tools and Applications Journal*, Special Issue, Kluwers Academic Publishers, 1997.

[10] ISO/IEC, Information Technology– Coding of Multimedia and Hypermedia Information, Support for Base-Level Interactive Applications, International Standard, ISO/IEC 13522-5, December 1995.

[11] W.R. Stevens, *UNIX Network Programming*, Prentice Hall, 1990.