A DAVIC Video-on-Demand System Based on the RTSP

Changwoo Jee* and Kang G. Shin**

** School of Computer Sci. & Eng. Sungshin University, Seoul, Korea*
*** Dept. of Electrical Eng. & Computer Sci., The University of Michigan, MI48105*
{cwjee@cs.sungshin.ac.kr, kgshin@eecs.umich.edu}

Abstract

The lack of a session to guarantee long-lived service and a function for control over the delivery of real-time streaming data in HTTP was a barrier in supporting a Video-on-Demand (VoD) service over the Internet. Recently, this barrier has been overcome with the introduction of Real-Time Streaming Protocol (RTSP).

This paper presents the design and implementation of an end-to-end DAVIC (Digital Audio-VIsual Council)-based VoD system to support the RTSP over the Internet infrastructure. A RTSP client can select VCR-like requests, which are sent in RTSP message format to RTSP-DSM gateway. In RTSP-DSM gateway, the messages are transformed user control commands encapsulated in DSM-CC (Digital Storage Media-Command and Control) format and sent to the server. Upon receipt of transformed commands, the server invokes a CORBA-based DSM-CC service gateway and the media stream pump to actually perform the requested operations. The reply messages from the server are sent to the client in RTSP message format by RTSP-DSM gateway.

Our VoD system has successfully completed feasibility via the Movie-on-Demand trials at our department. The main contribution of this work lies in a simple and effective system integration of a CORBA-based DAVIC VoD server and an RTSP-supported VoD client for MPEG transmission over the Internet via the LAN connection.

Keywords: VoD, RTSP, DAVIC, DSM-CC, CORBA

I. Introduction

The DAVIC, founded in 1994, has adopted a set of standard specifications to define interfaces, protocols and architectures for digital audio-visual applications and services [1]. The DAVIC supports a wide range of digital video applications with a variety of communication bandwidth needs including Movie-on-Demand (MoD), Tele-shopping, Distance Learning, Video Conference, and Karaoke-on-Demand.

One of the first applications enabled by the DAVIC is Video-on-Demand. VoD is created by adopting a number of standards and specifications from several standards bodies which include the ATM Forum Audio-Visual Multimedia Services (AMS) specification [2] and the DSM-CC specification [3] from the Moving Pictures Experts Group (MPEG) standards [4]. The AMS specification has been adopted to define the transform of MPEG-to-ATM. The DSM-CC specification supports the control functions to access a VoD server. The Common Object Request Broker Architecture/Internet-Inter ORB Protocol (CORBA/IIOP) specification [5], which is defined by the Object Management Group (OMG) is used to implement the DSM-CC architecture. The CORBA/IIOP specification is based on the Universal Networked Object (UNO) Remote Procedure Calls.

Apart from the DAVIC activities, the Internet has been expanding its application domains rapidly. Recently, a DAVIC-related work has been extending the design and implementation of a CORBA-based VoD system to support MPEG stream transmission over the Internet. An internet-based VoD service specified by the DAVIC is enabled by defining the transform of HTTP-to/from-DSM-CC, which has been implemented by the CORBA specification with a DAVIC-compliant system. However, the HTTP relies entirely on TCP, but TCP is not suited for time-based multimedia transmissions. In particular, the HTTP [7] has two serious problems with the DAVIC-compliant VoD system. First, the HTTP has no session to guarantee long-lived service because it is a stateless protocol. Second, the HTTP has no function for control and delivery of real-time streaming media over the Internet because it is basically an asymmetric protocol where the client issues requests and the server responds.

This difficulty has been overcome with the introduction of the RTSP [8], which is jointly developed by Netscape, Real-Networks, and Columbia University. RTSP works on top of the well-established Real-Time Transport (RTP), layered on top of UDP along which time-based media is streamed. The timestamp and sequence number fields of RTSP remedy the lack of delay constraints. RTSP supports the set-up and tear-down of a session for interactive multimedia services. Also, RTSP enables controlled delivery of streamed MPEG like VCR-functionality of DSM-CC. In particular, RTSP makes it easy to access the Web via the Internet because of its similarity to HTTP in its syntax and operation.

In this work, we propose a design and implementation of an end-to-end DAVIC-based VoD system to support the RTSP over the current Internet infrastructure. This is achieved through RTSP-DSM gateway that defines the transform of RTSP message format-to/from-DSM-CC message format for accessing a CORBA-based DAVIC VoD server over the Internet. Especially, the proposed system enables end-users to access more advanced DAVIC services and applications with a desktop computer that does not support CORBA runtime engine for the purpose to reduce the memory capacity.

The paper is organized as follows. First, the DAVIC reference model and DSM-CC for supporting a VoD service are described in Section 2. Then, the RTSP for a VoD service is discussed briefly in Section 3. Section 4 and Section 5 present a complete DAVIC-based VoD system architecture and its implementation details using the RTSP, respectively. Finally, the paper concludes with Section 6.

2. The DAVIC model

The objective of DAVIC model for a VoD service is to define technologies and information flows to be used within and between the video server, the delivery system and the Set-Top-Box (STB) at defined reference points. A general model of a DAVIC VoD system is shown in Figure 1.

DAVIC reference points (e.g., A1, A9) are a set of interfaces through which peer objects transfer information from one block to another. The S1 information flow is for a content-information flow (*contents*, e.g., a unidirectional flow of encoded video/audio data and binary navigation objects to be used by the STB from the server to the STB). The S2 information flow, on the other hand, is for a bi-directional control-information flow on the control plane of the application service layer (*presentation*, e.g, end-to-end control such as pause, fast forward/rewind, play, stop a movie). DAVIC has chosen ISO/IEC International Standard 13818-6, MPEG-2 DSM-CC User-to-User (U-U) Interface, for the S2 information flow between the STB and video server. The S3 information flow is a bi-directional control information flow on the control plane of the session and transport service layer (*session*, e.g., messages to establish, modify, or terminate a session; negotiate resource requirements between users, namely the server and STB, and the delivery system). DAVIC has chosen DSM-CC User-to-Network (U-N) Interface, for the S3 information flow.

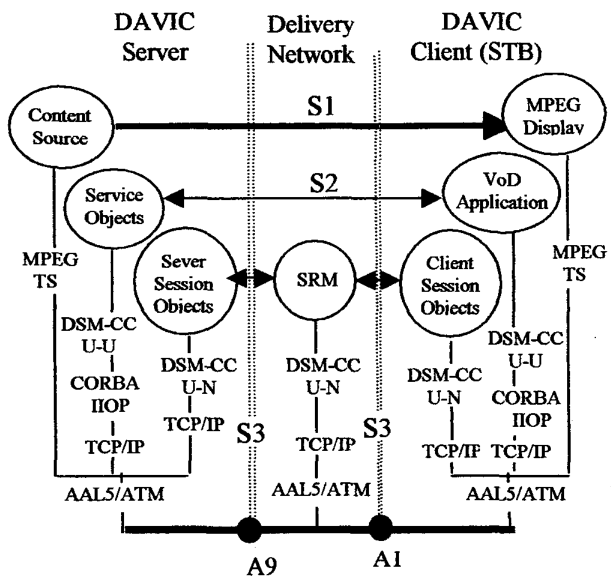

Figure 1. General DAVIC VoD model
The SRM denotes Session & Resource Manager

S1				S2	S3
MPEG PSI	MPEG Video ES	MPEG Audio ES	Other Data	DSM-CC U-U OMG CDR OMG GIOP IIOP	DSM-CC U-N
	MPEG2 PES	MPEG2 PES	MPEG2 SEC	TCP	UDP (TCP)
MPEG TS				IP	
AAL5					
ATM					
Lower Layers Protocols					

Figure 2. DAVIC S1, S2, and S2 protocol stacks

The protocol stacks at specific interfaces for the DAVIC content and control information flows are shown in Figure 2. The protocol stacks are based on an ATM-based transmission system. In the S1 flow, MPEG 2 TS packets are directly mapped to AAL5/ATM. The S2 flow protocol stack can be built upon a variety of underlying physical network architectures.

2.1 DSM-CC specifications

As the standard way of communication towards both the server, the delivery network, and the STB, The ISO/IEC standard has developed the DSM-CC specifications where the control functions and operations to manage the MPEG-1 and MPEG-2 streams are introduced. DSM-CC does not specify the underlying physical, data link, transport, or RPC layers of the overall protocol stack. DSM-CC consists of two broad categories of functionality; U-N interfaces and U-User interfaces. DSM-CC U-N manages sessions, which are an associated collection of resources required to delivery a service. On the other hand, the DSM-CC U-U interfaces support modular, basic building blocks that can be used to enable a wide-range of multimedia applications to configure, load, navigate, and request multimedia data in heterogeneous networked environments.

In particular, the DSM-CC U-U interfaces include the Service Gateway, Directory, Session, File, and Stream for VoD applications. These interfaces provide the basic building blocks for MoD applications to establish a session and browse the MPEG streams residing in the server. The Service Gateway interface as the entry point into the available services (e.g., MPEG streams) inherits Directory and Session interfaces. The ServiceGateway presents to the client applications a graph name service for stream objects (e.g., movies), arranged as a hierarchy of Directory as depicted in Figure 3. The DSM-CC clients obtain the stream object handle (e.g., object reference or IOR) using Directory name service supported by Directory and Session interface and invoke an operation to browse the stream objects. The object handle contains addressing information such as hosts, port, version, and object key. The Directory interface provides a CORBA name service interface plus operations to access stream objects and objects data. The Session interface enables a client application to attach to a name service. The Stream interface is used to emulate interactively VCR-like controls for manipulating MPEG streams such as video and audio, as defined by ISO/IEC 13818. The control operation defined by interface contain pause, resume, status, reset, jump, and play. The File interface enables a client application to read and write opaque data of an object.

A simple MoD scenario between the server and a STB to play a movie consists of three steps. First, a session is established S3 flow defined by DAVIC between the server and a STB. Second, user information such as list of movie and a reference (e.g., handle) to the requested movie is retrieved from the server over S2 flow. The STU use the handle and send VCR-like commands to control the movie over S2 flow. Third, The movie stream is sent over S1 flow established on PVC.

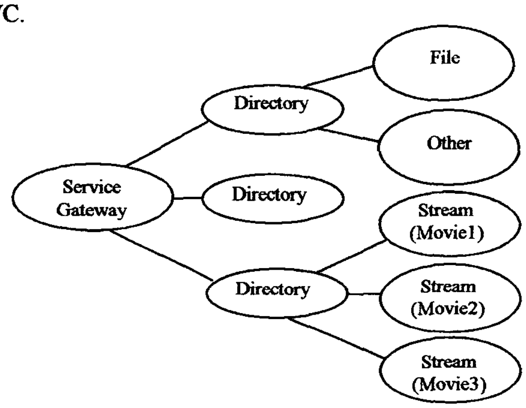

Figure 3. Multimedia Object Directory Hierarchy

3. Real-Time Streaming Protocol

RTSP provides an extensible framework to enable controlled, on-demand delivery of real-time data to address the needs for efficient delivery of streamed multimedia over IP networks. It is designed to minimize the overhead of multimedia delivery and to take full advantage of Internet infra-structural improvements such as IP multicast. RTSP has some overlap in functionality with HTTP so that extension to mechanisms to HTTP can in most cases also be added to RTSP. Therefore, RTSP may interact with HTTP in that the initial contact with streaming content is often to be made through a web page. However, RTSP differs fundamentally from HTTP in that delivery of streamed media takes place out-of-band in a different protocol. Both RTSP client and server can issue requests. Both RTSP client and server need to maintain state by default in almost all cases to set parameters and to continue to control a streamed media long after the request has been acknowledged. Currently, RTSP specification has been submitted for consideration as Internet-draft(RFC 2326) standard to the IETF.

RTSP works on top of the well-established Real-Time Transport Protocol (RTP), RFC 1889 layered on top of UDP along which time-based media is streamed. RTSP will use RTP as transport protocol. In an RTSP session server and client exchange the requests and response messages for setting up a transport mechanism for the streamed media, starting the streamed media, and closing the streamed media. An RTSP request message from a client to a server or versa includes the method to be performed to the resource, the identifier of the resource (e.g., the "rtsp://" syntax), and the protocol version in use. An RTSP response message defines an additional status codes to an RTSP request message. Basically, RTSP request method has defined the nine commands. But since not all RTSP servers and clients have the same functionality, the servers by necessity will support different sets of methods. A server may only be capable of playback for Movie-on-Demand services and may not support setting stream parameters. Therefore, the methods SETUP, PLAY, PAUSE, and TRARDOWN play a central role in defining the allocation and usage of stream resources on the server. In RTSP, new methods can be added to existing method as well.

We show in Figure 4 the transactions between the RTSP server and client in a simple session where the client controls and plays a MPEG stream on a remote server. RTSP control connection and data connection works on top of TCP/IP and UDP/IP, respectively. The SETUP request issued by the client specifies the transport mechanisms to be use when delivering the requested MPEG stream. On receiving the response message including the transport parameters and a generated session identifier from the server, the client issues PLAY request to the server. The PLAY request tells the server to start sending data via the mechanism specified in SETUP. An MPEG stream is sent over RTSP data connection. Also the PAUSE request causes the stream delivery to be stop temporarily. The TEARDOWN transaction is for releasing an existing session. It stops the stream delivery, freeing the resource associated with it.

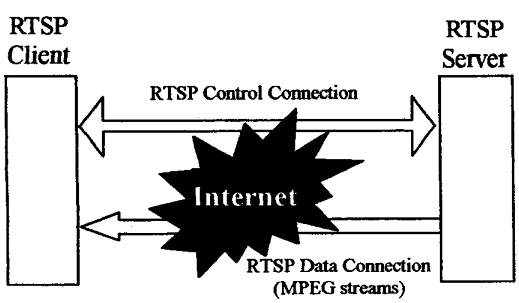

Figure 4. RTSP session

4. System Architecture

The DAVIC VoD model based on the RTSP protocol is illustrated in Figure 5. The Content Source, Service Objects, and Server Session Objects are identical to Figure 1 as shown in the server-side. RSTP Session objects are used to interact with the Server Session Objects. As the role of the SRM, we propose RTSP-DSM gateway to allow the Internet user to access the DAVIC server using the RTSP. The S2 and S3 flows use TCP/IP instead of ATM. Also, the S1 flow uses RTP instead of ATM.

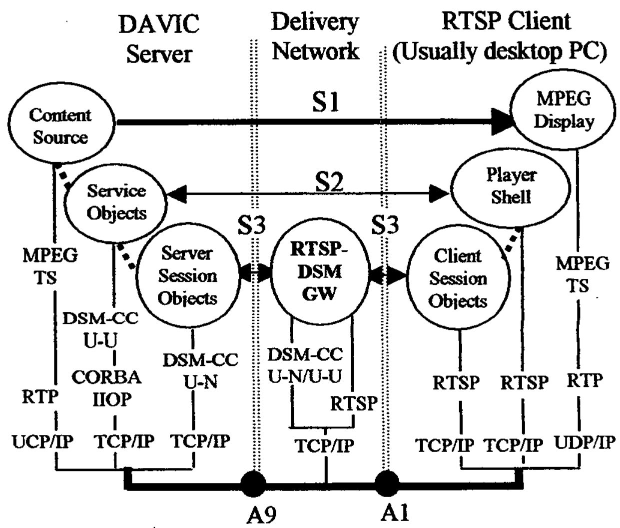

Figure 5. RTSP-based DAVIC VoD model

The proposed system architecture is shown in Figure 6. The DAVIC server comprises of four modules: the DSM-CC U-U Service objects, the DSM-CC U-N Server Session objects (a), a CORBA-related server/libraries, and the Media Stream Pump. The RTSP-DSM gateway consists of the DSM-CC U-N Session objects (b), a RTSP Daemon, and message converter. Finally, the RTSP client has four modules: the RTSP Session objects (c), the RTSP Service objects (d), a Player shell, and an MPEG decoder/TV display module.

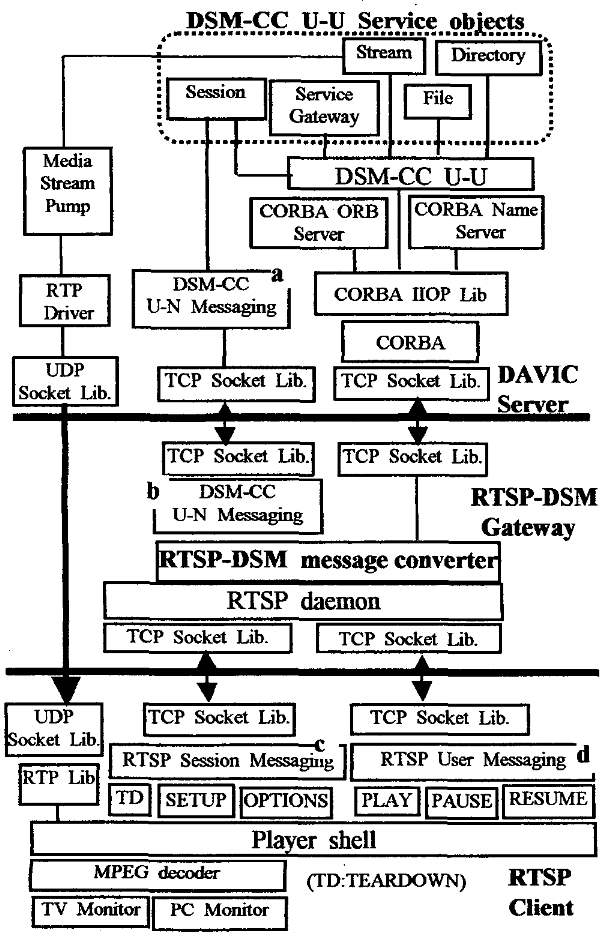

Figure 6. System Architecture

A RTSP Session Messaging and Service Messaging processes the session-related objects and service-related objects invoked by a Player shell, respectively. A RTSP daemon is able to accept the RTSP request commands sent over the S2 or S3 flow and send them to the message converter. On receipt of the RTSP response message from the message converter, the daemon will attempt to send it to the RTSP client via the S2 or S3 flow. The RTSP daemon maintains the waiting status to handle request commands. RTSP-DSM Gateway de-multiplex incoming request commands encapsulated in the RTSP message format. It is in charge of interpreting the commands and then transforms them to the appròpriate DSM-CC commands. At the opposite site, the DSM-CC commands sent over the S2 or S3 flow from the DAVIC server is encapsulated in the RTSP message formats by the gateway. DSM-CC U-N Messaging is used for the signaling such as session set-up or release for providing the VoD service.

The goal of the ServiceGateway object, which acts as the entry point to the DAVIC server, is in charge of interpreting the DSM-CC commands sent over S2 flow and then redirects them to other necessary objects. It invokes the Session Attach/Detach object if the request command is session set-up or tear-down. On the other hand, the Directory and Stream objects are invoked by the ServiceGateway if request commands are related to browsing and playing a movie. DSM-CC U-N Messaging, which is a set of U-N session operations, is invoked by the Session Attach/Detach object to establish or release the session between the DAVIC server, the SRM, and the RTSP client. With the proposed RTSP-DSM Gateway, each object in the DAVIC server receives the transformed commands and processes them as though they came from a DAVIC STU.

The CORBA Name server on top of CORBA IIOP provides the Directory object with CORBA naming service, which is compliant CORBA Naming specification, to easily locate the movies. The Directory object implements the function for binding names of the stream (movie) object to an object reference (IOR). It allows for RTSP client to browse an object reference, given a stream name and then to obtain an object handle. The Stream object based on Basic State Machine transition [3], which is defined by DSM-CC specification, enables the RTSP clients to interactively control the MPEG continuous media streams using an object handle. The Media Stream Pump, which acts as separately implemented module, is invoked by the Stream object to actually deliver a MPEG streams between the DAVIC server and the RTSP client. Also, the Media Stream Pump is based on Basic State Machine transition. The transport protocol for delivery of MPEG streams includes RTP over UDP/IP.

The following shows interaction of how RTSP works with RTSP-DSM gateway and DSM-CC server to receive the multiple streams. To begin establishing a new session with the server, the RTSP client send OPTIONS message to RTSP-DSM gateway:

```
C->GW OPTIONS rtsp://141.213.9.101:6060 RTSP/1.0
        Cseq : 1
```

RTSP-DSM gateway sends the SesSetUpInd to the server to get IORs of all movies in the server. On receipt of the SesSetUpRes, the gateway sends the information (e.g., movie path and title) associated with the movies to the client:

```
GW->S   SESSETUPIND
S->GW   SESSETUPRES
        IORs-length :750
GW->C   RTSP/1.0  200  OK
```

Cseq : 1
/video/mpeg/moviel.mpg
/video/mpeg/movie2.mpg
/video/mpeg/movie3.mpg

The client requests specific transport mechanisms to use when delivering the media streams. The gateway assigns the value of session identifier (*sessionid*) to identify the session throughout the life of the session. Also, the gateway determines the value of transport identifier (*transportid*) to specify the transport mechanism. On receipt of the SesSetUpInd message, the server should satisfy the session and transport request by establishing a new session with the client and allocate the requested transport parameters to the Media Stream Pump. The server sends SesSetUpRes message, which contains *sessionid* and *transportid*, to the gateway:

```
C->GW SETUP rtsp://141.213.9.101:6060/video/mpeg/
            moviel.mpg RTSP/1.0
      Cseq : 2
      Transport: rtp/avp; unicast; client_port=6970-
            6971;mode=play, x-real-rdt/udp;
            client_port=6970; mode=play
C->GW SETUP rtsp://141.213.9.101:6060/video/mpeg/
            movie3.mpg RTSP/1.0
      Cseq : 4
      Transport: rtp/avp; unicast; client_port=6974-
            6975;mode=play, x-real-rdt/udp;
            client_port=6974; mode=play
GW->S  SESSETUPIND   sessionId: 968367008-2
GW->S  SESSETUPIND   sessionId : 968367009-3
GW->S  SESSETUPIND   sessionId : 968367010-4
S->GW  SESSETUPRES   sessionId: 968367008-2
S->GW  SESSETUPRES   sessionId: 968367009-3
S->GW  SESSETUPRES   sessionId: 968367010-4
GW->C  RTSP/1.0  200  OK
       Cseq : 2
       Session: 968367008-2
       Transport: rtp/avp; client_port=6970-6971;
            server_port=6970
GW->C  RTSP/1.0  200  OK
       Cseq : 3
       Session: 968367009-3
       Transport: rtp/avp; client_port=6972-6973;
            server_port=6972
GW->C  RTSP/1.0  200  OK
       Cseq : 4
       Session: 968367010-4
       Transport: rtp/avp; client_port=6974-6975;
            server_port=6974
```

The client now requests the server to begin sending a Movie associated with the specific session using the transport mechanism specified in the SETUP response message. While the movie is being transmitted through the specific session, the client has the ability to send VCR-like command requests to the server. The following sets of messages from the client to the server are requests for the movies associated with the specific URL:

```
C->GW  PLAY rtsp://141.213.9.101:6060/video/mpeg/
            moviel.mpg  RTSP/1.0
       Cseq : 5   Session: 968367008-2
C->GW  PLAY rtsp://141.213.9.101:6060/video/mpeg/
            movie2.mpg  RTSP/1.0
       Cseq : 6   Session: 968367009-3
C->GW  PLAY rtsp://141.213.9.101:6060/video/mpeg/
            movie3.mpg RTSP/1.0
       Cseq : 7   Session: 968367010-4
GW->S  DSM_PLAY(ObjRef (moviel.mpg))
       Session: 968367008-2
GW->S  DSM_PLAY(ObjRef (movie2.mpg))
       Session: 968367009-3
GW->S  DSM_PLAY(ObjRef (movie3.mpg))
       Session: 968367010-4
GW->C  RTSP/1.0  200 OK
       Cseq: 5
GW->C  RTSP/1.0  200 OK
       Cseq: 6
GW->C  RTSP/1.0  200 OK
       Cseq: 7
```

To start the procedure for releasing an existing session, the client send TEARDOWN to the RTSP-DSM gateway to request that the server stop the delivery of movies associated with each of the sessions. On receipt of ReleaseInd, the server release all resources assigned to the session and then send ReleaseRes to the gateway. The gateway sends the RESPONSE message to the client:

```
C->GW  TEARDOWM  rtsp://141.213.9.101:6060/
            video/mpeg/moviel.mpg RTSP/1.0
       Cseq : 8   Session: 968367008-2
C->GW  TEARDOWN  rtsp://141.213.9.101:6060/
            video/mpeg/movie2.mpg RTSP/1.0
       Cseq : 9   Session: 968367009-3
C->GW  TEARDOWN  rtsp://141.213.9.101:6060/
            video/mpeg/movie3.mpg RTSP/1.0
       Cseq : 10   Session: 968367010-4
GW->S  RELEASEIND   sessionId: 968367008-2
GW->S  RELEASEIND   sessionId : 968367009-3
GW->S  RELEASEIND   sessionId : 968367010-4
S->GW  SESSETUPRES
S->GW  SESSETUPRES
```

S->GW	SESSETUPRES		
GW->C	RTSP/1.0	200	OK
	Cseq : 8		
GW->C	RTSP/1.0	200	OK
	Cseq : 9		
GW->C	RTSP/1.0	200	OK
	Cseq : 10		

Figure 7 illustrate signaling for a session establishment and release initiated by the RTSP client.

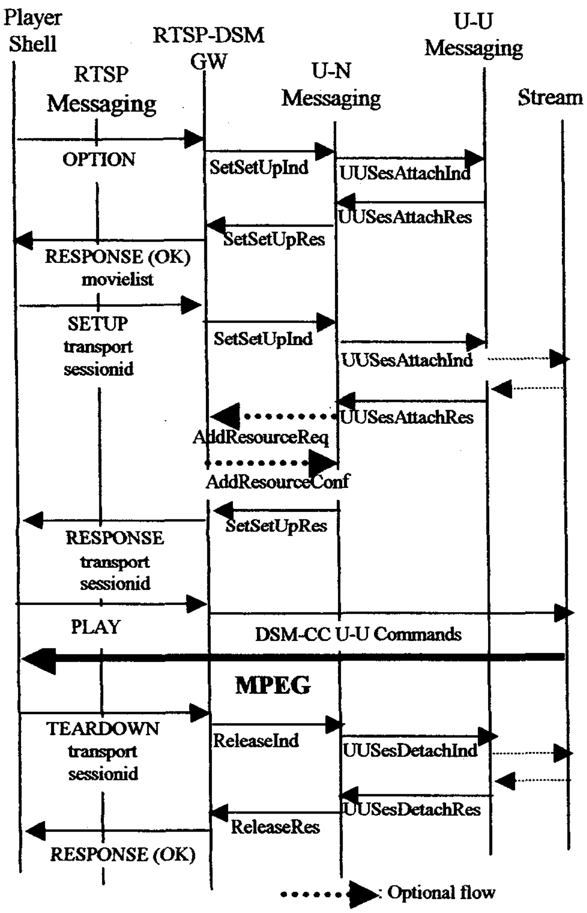

Figure 7. Scenario for RTSP Client Session Set-up

5. Implementation

CORBA-based DAVIC server software, which is implemented on a Solaris operation system in SUN platform, uses Iona Technologies – Orbix product for C++ to provide the CORBA 2.0 compliant DSM-CC implementation. DSM-CC U-U Service objects are specified by the Interface Definition Language (IDL). DSM-CC U-U ServiceGateway object and Directory object are implemented by the NamingContext defined in the CORBA Object Services Naming module. In Stream object, RESUME, PAUSE, STOP, and PLAY functions are implemented to allow the user VCR-functionality using the IDL. Berkely UNIX TCP sockets library is used to communicate with processes on RTSP-DSM gateway. We have designed and implemented a Media Stream Pump algorithm based on the DSM-CC Basic State Machine (BSM) transitions to control the delivery of a movie stream when the stream primitive request is received. It provides a set of ready running streaming processes, which can be attached to any Stream object upon the user request. To support multiple streams to being deliver for the multiple clients, Our Media Stream Pump algorithm supports the controller that can automatically invoke a new stream pump process when more users request the MPEG streams. The RTP driver working over UDP is used to carry the MPEG streams to the RTP driver on the client.

RTSP-DSM gateway is also implemented on a Solaris operation system in SUN platform. It runs the daemon process, the message converter process, and U-N messaging process. A daemon and the message converter process are used to transform the messages between a DAVIC server and a RTSP Client. The U-N Messaging process is used to process the session-related messages. Processes communicate via the IPC mechanism. Berkely Sockets are also used for inter-process communications on both a DAVIC server and a RTSP Client.

The RTSP Client software, equipped with a VideoPlex MPEG-2 PCI playback board connected to a TV display and Ethernet port associated with the session establishment with the RTSP-DSM gateway, runs on the Windows 98 platform. We have implemented the RTSP session and service messaging process to demonstrate access to the DAVIC VoD server via the Internet using Player shell. The Player shell has the session-related buttons and the VCR-like buttons for user interface when viewing a movie. The OPTIONS button allows the users to get the movie path and list to browse the movies. The OPTIONS operation interacts with U-U Directory object. The SETUP and TEARDOWN button allows the user to establish and release the multiple session with the ServiceGateway. The PLAY buttons allow the user to start playing the current selected movie. For playback, the MPEG streams are carried over RTP/UDP to the RTSP client and decoded by a MPEG decoder for TV display. Other buttons allow the user to provide VCR-like playback functionality.

The setup of the system configuration for a Movie-on-Demand services trials within our laboratory is as shown in Figure 8. A SUN Sparc workstation in our department, whose IP address is 141.213.9.101, runs the CORBA-based DAVIC DSM-CC server software program. Any Pentium processor-based Win32 PC, equipped with a LAN card, can run the RTSP client program. The RTSP client is assigned IP address 141.213.9.75. The intermediate machine over the delivery network is RTSP-DSM gateway, whose IP address is 141.212.9.30. The SRM runs the daemon software program, the message converter software program, and DSM-CC U-N messaging software program for the session and the resource manager.

The prototype of system integration of a CORBA-based DAVIC VoD server and an RTSP-supported VoD client has been successfully demonstrated the Movie-on-Demand service at the DAVIC interoperability. Simple VCR commands could be exchanged via RTSP formats for processing on the DAVIC VoD server. Visually, the TV playback display of the received MPEG bit-stream at the client site had equal quality as that in the display generated from decompressing the same MPEG file stored on-site. The display was able to playback at the 30 frame per seconds rate, and the lag from the time that transmission was activated to the time display began to play was pretty short.

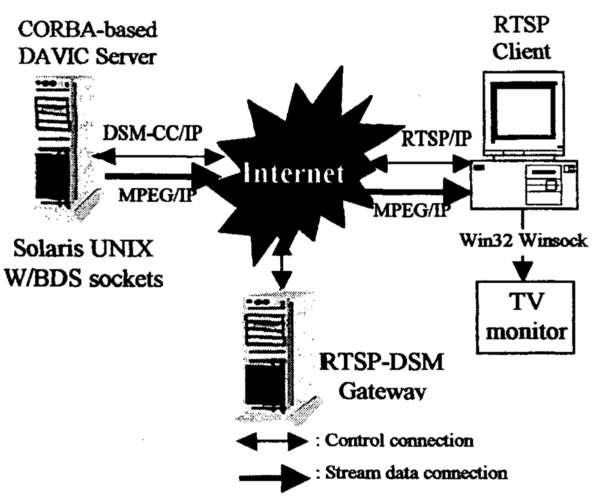

Figure 8. The testbed for DAVIC VoD service

6. Conclusions

In this paper, we focus primarily on the design and implementation of an end-to-end DAVIC VoD system consisting of the DSM-CC based server, RTSP-DSM Gateway, and the RTSP-supported clients. A prototype testbed has successfully demonstrated feasibility via the Movie-on-Demand service at our department. Enabling access of DAVIC-based VoD system by RTSP clients should neither be seen as a replacement for, nor an alternative to, using a DAVIC-compliant S2 flow, which is implemented with a CORBA-based DSM-CC. The S2 flow is more flexible and powerful compared to the relatively limited and simple format exchange due to incompatibility via RTSP. However, using recently released real-time streaming protocol, while DAVIC compliant systems are scarce and Internet capable systems are widely available, our system allow the clients to get DAVIC-based Movie-on-Demand service over the current Internet infrastructure using a desktop computer that does not support CORBA runtime engine for the purpose to reduce the memory capacity.

7. References

[1] Digital Audio-Visual Council, *DAVIC specifications* 1.0, http://www.davic.org

[2] ATM forum, *Audiovisual Multimedia Service: Video on Demand specification* 1.0, *ATM forum*, http://www.atmforum.org

[3] ISO/IEC 13818-6, *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 6: Digital Storage Media Command and Control (DSM-CC)*, December 1995.

[4] ISO/IEC 13818-1,2,3, *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – System*, July 1995.

[5] Object Management Group (OMG), *The Common Object Request Broker Architecture*, Revision 2.0, http://www.omg.org

[6] Alex de Jong and Taewoon Kang, "Access to a DAVIC-Base Video on Demand System Using the World Wide Web", *11-th International Conference on Information Networks*, Taipei, Jan 27-29, 1997.

[7] R. Fielding, J. Gettys, J. Mogul, H. Nielsen, and T. Berners-Lee, " *Hypertext transfer protocol – HTTP/1.1*," RFC 2068, Internet Engineering Task Force, Jan 1997.

[8] H. Schulzrinne, A. Rao, and R. Lanphier, " *Real Time Streaming Protocol (RTSP),* " RFC 2326, Internet Engineering Task Force, Feb 1998.

[9] Chang-Woo Jee, Sung-Han Park, and Jun-Jae Yoo, " VoD System for Interactive Multimedia Services, *International Conference on Consumer Electronics'97*, Consumer Electronics Society, Vol.16, pp.68-pp.69, 1997.