

# MPEG-2 비트스트림의 ATM 전송을 위한 DSM-CC U-U 구현

최성중\*, 김용한\*, 이호장\*, 심재규\*, 김재동\*\*, 고종석\*\*

\*서울시립대학교 전자전기공학부 (chois@scucc.scu.ac.kr)

\*\*한국통신 전송기술연구소

## Implementation of DSM-CC U-U for MPEG-2 Bit Stream Transfer over ATM

SeongJong Choi\*, Yong Han Kim\*, HoJang Lee\*, JaeKyu Shim\*, Jae D. Kim\*\*, Jong-Seog KOH\*\*

\* Department of Electrical Engineering, University of Seoul (chois@scucc.scu.ac.kr)

\*\* Transmission Technology Research Labs, Korea Telecom

### 요약

본 논문에서는 MPEG-2 비트스트림의 ATM 전송을 위한 DSM-CC U-U 인터페이스 구현에 대해 기술하였다. DSM-CC에서 정의한 주된 기능은 MPEG-2 비트스트림의 VCR과 같은 제어기능과 서비스 도메인을 열람할 수 있는 브라우저 기능이다. 구현된 DSM-CC를 시험하기 위하여 ATM 전송을 사용한 VoD 시스템을 구성하였다. 이러한 통합 시스템 구현을 위해 DAVIC에서 정의된 시스템 모델을 사용하여 기본적인 구조를 설계하였고, ATM Forum에서 정의된 표준을 기본으로 ATM 비트스트림 전송 시스템을 구현하였다.

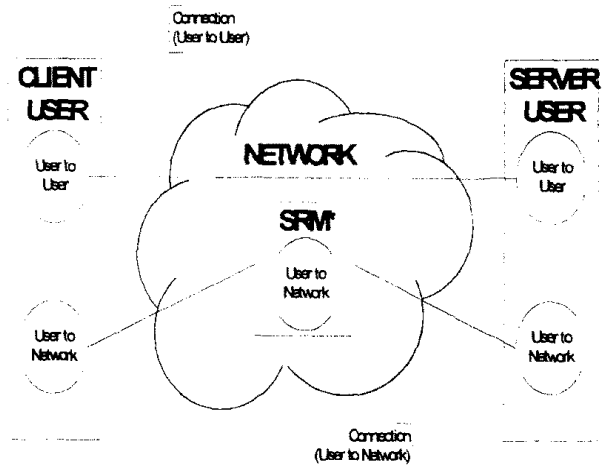
### 1 서론

DAVIC (Digital Audio-Visual Council)은 VoD (Video on Demand)와 같은 응용 서비스를 위한 통합 시스템 표준을 정의하고 있다 [1]. ATM을 사용하여 MPEG-2 비트스트림을 전송하는 시스템을 위해 DAVIC에서는 ATM Forum과 Moving Pictures Expert Group (MPEG)에서 정의된 사양을 채택하였다. ATM Forum 규격 AF-SAA-0049.001에서는 MPEG-2 비트 스트림을 ATM을 사용하여 전송하는 방법에 대해 기술한다 [5]. Digital Storage Media Command and Control (DSM-CC)는 이종 환경하에서 공통적으로 적용 가능한 개방형 환경에서의 비트스트림 제어를 위한 표준이다.[1]

본 논문에서는 DSM-CC에 정의되어있는 참조 모델과 User-to-User Interface에 대해 검토하고, VoD 시스템에 적용하기위해 필요한 요구 기능 및 고려 사항을 기술하고, 이를 기반으로 하는 시스템 구현 및 구현 방안에 대해 설명하였다.

### 2 DSM-CC 참조 모델 및 User-to-User 인터페이스

디지털 저장 매체 명령 및 제어(Digital Storage Media Command and Control: DSM-CC) 규격의 가장 중요한 목적은 ISO/IEC 11172 (MPEG-1) 및 ISO/IEC 13818 (MPEG-2) 비트스트림들을 관리하기 위해 필요한 제어 및 작동 기능을 제공하는 프로토콜의 집합을 정의하는 데에 있다.



\*My provide session, connection and configuration management and control.  
그림 1 DSM-CC 시스템 참조 모델

그림 1은 DSM-CC 시스템 참조 모델이다. 이 모델에서는 스트림이 서버로부터 클라이언트로 전달된다고 가정한다. 서버와 클라이언트는 모두 DSM-CC 망의 "사용자"이다. 어떤 서비스를 제공하기 위해 필요한 자원들을 연관 지어 놓은 모음을 세션이라 하는데, DSM-CC는 세션도 관리한다. DSM-CC 세션 및 자원을 중앙집중 형태로 관리하는 논리적인 개체를 "세션 및 자원 관리자(Session and Resource Manager: SRM)"라 정의한다. 시스템 참조 모델에서는 이 SRM과 그 기층 망을 합하여 "망"이라 한다. DSM-CC 모델에서는 망, 클라이언트, 그리고 서버가 피어(peer)들이다.

DSM-CC는 크게 U-N 및 U-U의 두 가지 범주의 인터페이스로 구성된다. U-U 는 서비스를 위한 시스템 개체들 사이의 인터페이스를 정의한다. 세션 게이트웨이 관리, 서비스 게이트웨이 관리, 스트림 서비스 제어, 파일 서비스 제어, 데이터베이스 서비스 제어, 이벤트 관리 등이 U-U 인터페이스에서 정의된다. 이러한 DSM-CC 기능들은 CORBA (Common Object Request Broker Architecture) 환경에서 구현되어진다. CORBA 환경에서의 인터페이스는, 상이한 기종간의 호환성을 보장하기 위하여, 객체지향의 (Object-Oriented) 방법으로 정의되고, OMG (Object Management Group)에서 제안된 IDL(Interface Description Language)로 표현된다. 이러한 기능들과 인터페이스들의 모음을 U-U 라이브러리라 부른다. DSM-CC를 사용한 최상위 계층의 응용 프로그램 (Application)을 개발하려면 U-U 라이브러리를 사용해야만 한다.

환경설정, 세션 관리 그리고 자원 관리는 DSM-CC U-N에 속한다. 환경설정 기능은 SRM과의 통신 및 디폴트 U-U 세션을 초기화를 위해 필요한 DSM-CC 고유의 혹은 망 고유의 파라미터들을 사용자, 즉 클라이언트와 서버에게 제공한다. 세션 및 자원 관리 기능은 U-U 세션에 필요한 유한 자원을 관리하는 기능을 제공한다.

U-U Library는 어느 개체와 연결되는가에 따라 다음 세 가지의 인터페이스로 분류될 수 있다 [그림2]. Application Portability Interface (API)는 Application를 제작을 위해, Service Interoperability Interface는 프로토콜스택을 통한 Remote Procedure Call을 위해, U-N Session Interoperability Interface는 동일 시스템의 U-N 계층과의 연결을 위해 정의되어진다. 예를 들어 Session을 열기 위해 Application에서

Session API를 사용하면 U-N 인터페이스가 구현되어져 있는지에 따라 각기 다른 인터페이스가 사용되어진다. 즉, U-U 라이브러리는 U-N 인터페이스가 구현된 시스템에서는 SessionUU 인터페이스가 사용되고, 구현되지 않은 시스템에서는 SessionSI 인터페이스를 사용한다.

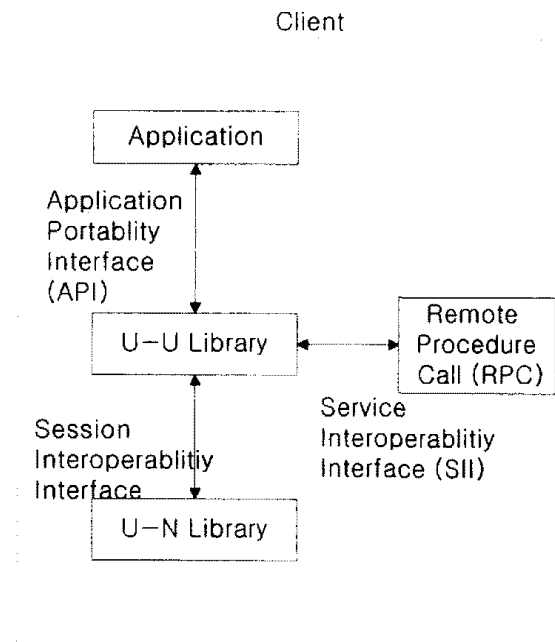


그림 2. U-U Library와의 연결로 분류된 Interface 구분

DSM-CC U-U 인터페이스를 기능적인 기준으로 분류하면 최소한의 요구 사항인 Core Interfaces와 선택적인 Extended Interfaces로 구성된다. 그림 3은 Core Interface와 이들이 제공하는 기능이다. Core 인터페이스는 다시 abstract 인터페이스와 instantiable 인터페이스로 구분된다. abstract 인터페이스는 실제로는 응용프로그램이나 서버 구현에는 사용되지 않지만 각 인터페이스에 공통적으로 속해있는 속성과 함수들을 정의하고 있다. instantiable 인터페이스는 abstract 인터페이스로부터 상속 받아 실제로 사용될 수 있도록 설계된 인터페이스이다.

세션 인터페이스는 세션 및 자원 관리를 위한 DSM-CC 프로토콜에 대한 U-U 라이브러리에 대해 정의된다. (세션 관리와 자원 관리 기능을 실제로 구현하는 데에는 반드시 U-U 라이브러리를 사용할 필요는 없다.) 서비스 게이트웨이 인터페이스는 이용 가능한 서비스를 찾을 수 있는 기능을 제공한다. 스트림 서비스 인터페

이스는 플레이, 정지, 점프, 패스트 포워드 등, MPEG-2 스트림을 제어하는 기능을 제공한다. 파일 서비스 인터페이스는 다른 U-U 라이브러리 서비스와 일관성을 유지하면서 일반적인 작업을 이용할 수 있는 기능을 제공한다.

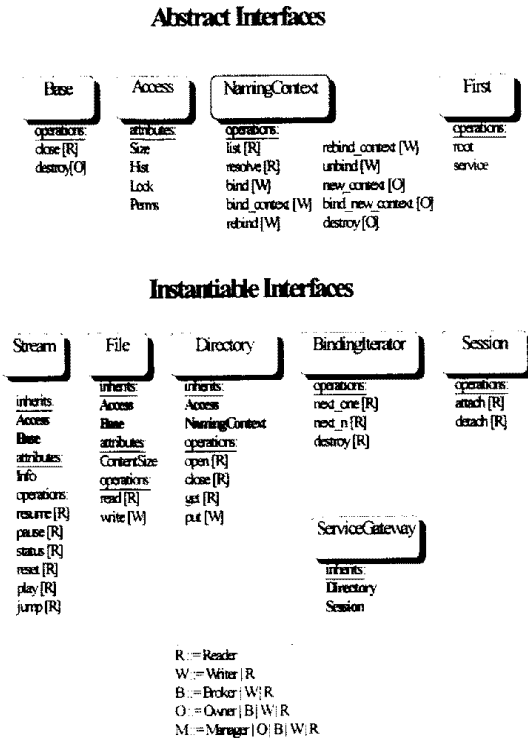


그림 3. Core Interface

### 3 VoD 시스템 구현

그림 4는 본 연구에서 구현된 VoD 시스템의 프로토콜 스택 구성도이다.

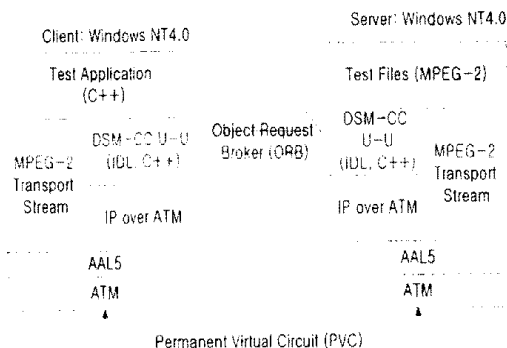


그림 4. VoD 시스템 프로토콜 스택

VoD 시스템 구현을 위해 Windows NT 환경을 사용하였다. 서버와 클라이언트에는 155Mbps ATM 카드를 장착하였고 AAL5를 사용하기 위한 API로 Windows Socket 2.2, ATM extension을 사용하였다[3,4]. 프로그램 언어는 C++를 사용하였고 효율적인 I/O를 위해 Win32 API를 사용하여 크라스를 구현하였다. 서버에는 약 1분 분량의 Constant Bit Rate MPEG-2 Transport Stream으로 제작된 파일이 저장되어 있다. 클라이언트에서 작동되는 응용 프로그램은 크게 두가지가 있다. 하나는 디렉토리를 열람할 수 있는 브라우저이고 다른 하나는 브라우저를 사용하여 비트스트림이 선택되었을 때 이를 VCR과 같이 제어할 수 있는 제어기이다. 현재 구현된 제어기능은 play, pause, FF, Slow이다. 모든 Graphic User Interface는 MFC를 사용하여 구현되었다.

본 연구에서는 CORBA를 사용한 DSM-CC UU 인터페이스 구현을 위하여 IONA사에서 제공하는 ORBIX를 사용하였다. 이 패키지는 크게 두가지 기능을 제공한다. 첫 번째 기능은 기존의 IDL 파일로부터 서버 skeleton, 클라이언트 stub 및 구현 크라스를 위한 원형을 컴파일하는 것이고, 두 번째 기능은 Remote Procedure Call (RPC) dmf 위한 Object Request Broker (ORB)이다.

현재 CORBA service 중 Naming specification를 사용할 수 있도록 서버에 구현한 프로그램이 많이 있다. 하지만 이로부터 상속 받은 인터페이스를 구현하려면 문제점이 있게 된다. 현재 CORBA에서는 구현된 개체 (Object Implementation)로부터 상속 받아 새로운 개체를 구현하는 방법에 대해 정의하고 있지 않기 때문이다. 따라서 NamingContext에서 상속을 받는 Directory의 구현을 위해 제 3자가 구현한 서비스를 사용하는 데 많은 제약이 따른다. 이 문제를 해결하기 위해 NamingContext와 BindingIterator의 소스코드나 라이브러리 파일이 필요하다. 또 다른 해결 방법은 서버에 구현된 개체로부터 상속 받은 새로운 개체를 구현하기 위하여 제안된 Naming Service를 사용한 Server Pools 구조를 사용하는 것이지만, 이를 사용하면 가장 상위 레벨의 root Directory만 사용하여야 하는 단점이 있다 [6]. 본 연구에서는 완벽한 서비스를 위해 NamingContext와 BindingIterator를 새로 구현하여 이로부터 상속 받는 Directory를 구현하였다.

Stream 인터페이스를 구현하기 위해 서버와 클라이언트에서는 multithreading을 사용하였다. 서버에서는 MPEG-2 파일을 읽는 thread와 읽은 스트림을 ATM 을 사용하여 클라이언트에 전송하는 thread를 생성하였고, 클라이언트에서는 ATM으로부터 전송된 스트림을 버퍼로 복사하는 thread와 버퍼로부터 MPEG-2 Decoder로 스트림을 전송하는 thread를 생성하였다. 이들 thread 간의 동기화는 semaphore와 Event를 사용하였다.

ATM Forum에서는 현재 Constant Packet Rate (CPR)의 MPEG-2 Single Program Transport Stream (SPTS)를 Service Specific Convergence Sublayer (SSCS)가 없는 AAL5를 사용하도록 규정하고 있다. SPTS의 패킷의 크기는 188 byte 인데 ATM Forum에서는 하나의 CPCS-SDU에 정수 (N)개의 SPTS를 포함하도록 규정하고 있다. 예를 들어 두개의 SPTS 패킷을 하나의 CPCS-SDU에 포함시킬 경우 CPCS-PDU의 크기는 384 byte가 된다. 하위층의 SAR은 384개의 CPCS-PDU를 8개의 48 byte 크기의 패킷으로 자른 다음 ATM 층에서 5 byte의 header를 삽입하여 전송한다.

#### 4 결론

본 논문에서는 DAVIC에서 정의된 멀티미디어 서비스 시스템을 기초로 하는 VoD 시스템을 구현하기 위한 고려 사항을 분석하고 이에 따르는 세부 기능에 대하여 언급하였다. ATM Forum에서 제안된 표준안을 기본으로 비트스트림의 ATM 전송시스템을 구성하였고, 이를 위해 Window Socket 2.2, ATM Extension API를 사용하였다. 또한 이기종간의 호환성을 보장해주는 DSM-CC 표준안을 채택하여 비트스트림의 제어 및 서비스 열람을 위한 브라우징 기능을 구현하였다.

본 연구에서 구현한 기술들은 앞으로 셋톱 박스와 같은 멀티미디어 단말기 제작, VCR과 같은 제어기능을 갖춘 비디오 서버 설계, ATM을 사용한 멀티미디어 통신 등 여러 분야에 활용될 것으로 기대된다.

\*본 연구는 한국통신 전송기술 연구소의 연구비 지원으로 수행되었음.

[참고 문헌]

- [1] ISO/IEC 13818-6, "Information technology – Generic coding of moving pictures and associated audio information – Part6: Extension for Digital Storage Media Command and Control", 12-July-1996 (pre-editing release).
- [2] DAVIC 1.0 Specification, December 1995.
- [3] Windows Sockets 2, Application Programming Interface, An Interface for Transparent Network Programming, Under Microsoft Windows™, Revision 2.2.0, May 10, 1996
- [4] Windows Sockets 2 Protocol-Specific Annex Revision 2.0.3, May 10, 1996
- [5] Audiovisual Multimedia Services :Video on Demand, Specification 1.1, af-saa-0049.001, March, 1997
- [6] [http://www.iona.com/Developers/Cookbook/s\\_p\\_ns/index.html](http://www.iona.com/Developers/Cookbook/s_p_ns/index.html)