

Video on Demand 서비스를 위한 MPEG-4 시스템 구현

최 승 철, *김 남 규, *최 석 립
세종대학교 컴퓨터공학과, *세종대학교 전자공학과
전화 : 02-3408-3829 / 핸드폰 : 011-9119-4158

Implementation of MPEG-4 System for Video on Demand Services

Seung-cheol Choi, Nam-Kyu Kim, Seokrim Choi
Dept. of Computer Engineering, Sejong University
E-mail : choisc@image.sejong.ac.kr

Abstract

We describe the implementation of MPEG-4 Systems for Video on Demand services. We discuss the issues in implementint MPEG-4 Systems, specially Object Description and Elementary stream management(Buffer control, Synchronization). The file-server pump the multiplexed streams to the client through the DMIF. Interaction between server and client is done with *UserCommand(play, pause, stop, fast forward, ...)*. Stream data are multiplexed off-line. In advance, the file-server read multiplexed streams and pumps it to MPEG-4 terminal(client). Our system can be used in various application not only Video on Demand services, but also Internet broadcasting, remote surveillance, mobile communication, remote education, etc.

I. 서론

ISO/IEC 14496(MPEG-4)은 객체 기반 부호화 기법을 적용한 동영상 압축을 위한 국제 표준으로써, 객체 기반 부호화 기법은 수신 단말 측에서 오디오-비주얼 객체에 대한 다양한 조작을 가능하게 하고, 하나의 SCENE을 구성하고 있는 영상, 음성 등 각 객체별로 부호화하기 때문에 부호화 효율을 높이고 동시에 각 객체에 대한 가공/편집을 가능하게 한다. 또한, MPEG-4는 컴퓨터, 통신, 영상산업 등의 분야들을 위한

표준화된 형식과 인터페이스를 정의하고 있으며, 오디오-비주얼 객체에 대한 객체 기반의 통신, 접근, 관리를 지원한다[1~5].

본 논문은 주문형 비디오 서비스(Video on Demand Services)를 위한 MPEG-4 시스템 구현을 다루고 있다. MPEG-4 시스템을 적용한 VOD 단말의 구현을 통하여 버퍼를 효율적으로 관리하고 객체 또는 객체간의 동기화를 달성하는 방법과 일반적으로 사용될 수 있는 시스템 아키텍처를 제시하고 있다.

2절에서 논문을 통하여 구현된 내용을 중심으로 제안된 VOD 단말의 아키텍처에 대하여 설명하고 VOD 단말에서의 버퍼 관리와 시스템 및 각 객체에 대한 동기화를 이루는 방법에 대하여 알아보도록 할 것이다. 3절에서는 True-Video on Demand(T-VOD)를 달성하기 위한 서버와의 상호동작에 관하여 알아보고 4절에 결론을 내리도록 하겠다.

II. 시스템 아키텍처

이번 절에서는 논문에서 구현된 VOD 시스템의 아키텍처와 버퍼 관리 및 동기화에 관하여 자세히 설명하겠다. 그림 1에서 보이는 아키텍처는 VOD 서비스를 위한 일반적인 모델이라고 할 수 있다. 서버의 역할은 단말의 서비스 요청을 받아 들어 서버의 로컬 저장 매체에 저장되어 있는 FlexMux 스트림 파일을 읽어 서버의 스트리밍 정책에 의하여 단말에게 전송하며, 서버와 단말의 제어 정보(서비스 요청, UserCommand 등)들은 Delivery Multimedia Integration Framework(DMIF)[2]을 통하여 교환된다.

그림 1에서 실선의 화살표는 제어 정보의 흐름을 보여주고, 블록의 화살표는 멀티미디어 데이터 즉,

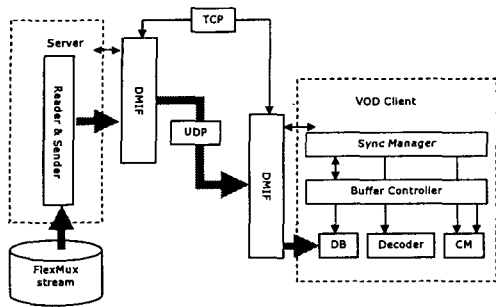


그림 1. 서버와 VOD 단말 사이의 데이터/제어 흐름도

Elementary Stream(ES)의 흐름을 보이고 있다. 단말의 사용자 또는 응용이 서비스를 요청하게 되면 서버는 전송하게 될 콘텐츠를 위한 Object Descriptor(OD) 스트림을 전송하여 콘텐츠를 복호화하고 재생하기 위한 정보들을 VOD 단말에게 알려주게 된다. SyncLayer(SL)-패킷의 형태로 전송되는 OD 스트림을 통하여 VOD 단말은 디코더 등의 환경을 설정하고 ES를 전송할 것을 요청함으로써 서버와 클라이언트의 동작은 이루어진다.

본 논문에서는 MPEG-4 시스템을 이용한 단말의 구현을 중심으로 다루고 있기 때문에 서버에 대한 내용은 언급하지 않고 VOD 단말과 서버간의 데이터/제어 정보의 흐름을 중심으로 어떤 방식으로 통신을 하는지를 알아 보기 위해 논문에서 구현된 VOD 시스템 전체 아키텍처를 간략하게 설명하였다.

2.1 VOD 단말

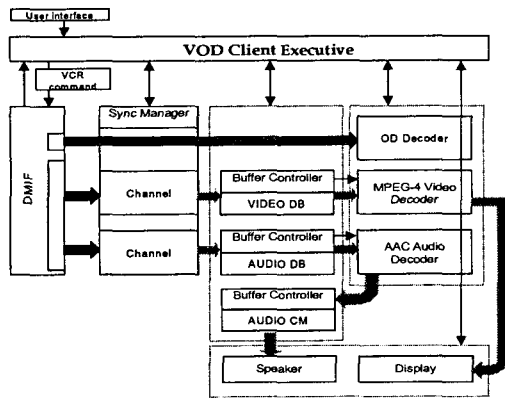


그림 2. VOD 단말의 전체 구성 및 데이터/제어 흐름도

그림 2는 논문에서 구현된 VOD 단말의 전체 구성 및 데이터와 제어 정보의 흐름을 보이고 있다. VOD 단말은 MPEG-4 시스템을 통하여 구현되었기 때문에 서론에 설명한 MPEG-4의 장점들을 이용하여 보다 효율적이고 발전된 서비스를 제공할 수 있다는 점이 논문에서 설계/구현된 VOD 시스템의 장점이다. VOD 단말은 크게 다음과 같은 6개의 모듈로 구성되어 있다.

- **Client Executive** : VOD 서비스를 위한 단말의 시스템을 총괄하고 자원관리 및 인터페이스를 담당

- **DMIF Client** : 서버와의 세션 및 네트워크 자원 관리
- **Sync Manager** : 시스템 또는 객체의 동기화 담당
- **Buffer Controller** : 각 디코더의 버퍼를 관리하고 버퍼의 상태를 단말에 알림
- **Elementary stream Decoder** : 각 ES의 복호화 담당
- **Compositor** : 복호화된 객체의 구성 및 재생

각각의 모듈은 쓰레드를 통하여 독립적으로 동작하면서 Client Executive 모듈이 나머지 모듈을 관리하게 된다. 제시하는 VOD 단말의 구조에서는 각 모듈간에 메시지를 주고 받는 방식이 아니라, 기본적으로 버퍼의 상태와 동기화 정보를 통하여 VOD 단말의 시스템이 동작하게 된다. 단말은 서버에서 전송되어 오는 데이터를 Decoding Buffer(DB)에 입력하고 Sync Manager를 통하여 언제 ES를 복호화 할지를 알게 된다. 따라서 본 논문에서 가장 세심하게 다루어진 부분이 바로 버퍼 관리와 동기화에 관한 내용이다.

실제 구현에 있어서, MPEG-4가 객체 기반의 부호화 기법을 사용하기 때문에 객체 지향 프로그래밍(OOP) 언어인 C++를 이용하여 각 모듈을 클래스 또는 라이브러리로 구현하였다.

2.2 VOD 서비스의 초기화

MPEG-4 시스템에서는 오디오-비주얼 객체의 묘사를 위하여 Object Description Framework(ODF)를 정의하고 있다. 서버와 단말은 ODF를 통하여 MPEG-4의 Scene을 구성하기 위한 계층적 구조와 각 객체 즉, elementary 스트림을 위한 정보를 공유하게 된다.[1]

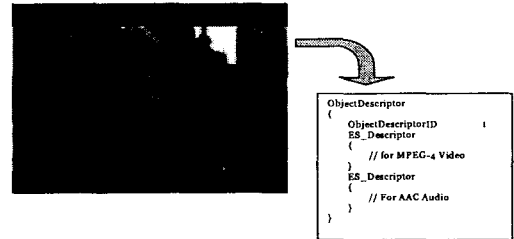


그림 3. Object Descriptor를 통한 객체의 묘사

그림 3은 VOD 서비스를 위한 하나의 객체 또는 콘텐츠가 MPEG-4 비디오 비트스트림과 AAC 오디오 비트스트림을 포함하는 것을 ODF의 ObjectDescriptor를 통하여 정의하고 있다. 그리고 각 ES_Descriptor에는 해당 ES를 복호화하기 위해 필요한 정보와 동기화

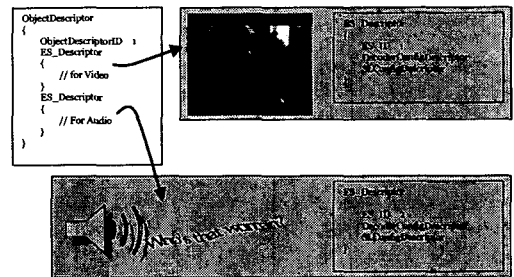


그림 4. ES_Descriptor를 통한 각 elementary 스트림에 대한 묘사

Video on Demand 서비스를 위한 MPEG-4 시스템 구현

를 위해 필요한 정보가 각각 DecoderConfigDescriptor와 SLConfigDescriptor에 정의된다(그림 4). 이 밖에 ES_Descriptor에는 QosDescriptor, IPMP_Descriptor 등을 추가적으로 정의할 수 있다.

일반적으로 ObjectDescriptor에 의해 객체에 대한 정보를 단말에게 알려 줄 수 있지만, 서비스를 위한 프로파일링 정보를 제공하는 특별한 형태의 ObjectDescriptor인 InitialObjectDescriptor(IOD)를 가지고 있다. VOD 서비스를 위한 서버와 단말 간의 초기화는 IOD를 통하여 이루어진다. 다음 그림 5의 순서도를 통하여 단말의 초기화 과정을 알 수 있다.

서버와 단말 사이의 세션이 시작되면, 서버로부터 최초로 IOD 패킷이 전송된다. IOD를 통하여 단말을 초기화하고 단말은 ES 패킷의 수신 대기 상태로 들어간다. J.Deicke *et al.*[6]의 논문에서 보이고 있는 초기화 동작과정과 본 논문에서 제안하는 초기화 과정을 비교하여 살펴해보도록 하겠다.

IOD는 MPEG-4 VOD시스템에서 사용하고 있는 객체를 접근 / 사용할 수 있도록 하기 위해 단말을 초기화하기 위한 정보를 제공한다. 논문에서는 MPEG-4 시스템에서 정의하는 Scene description[1]은 사용되지 않고 서버와 단말 사이의 모든 상호 작용은 DMIF의 UserCommand를 통하여 이루어 지기 때문에, J. Deicke *et al.*에서 보이는 장면 묘사를 위한 스트림과 OD 스트림을 위한 채널을 개설하지 않아도 된다. 따라서 비디오와 오디오를 위한 ES_Descriptor를 IOD를 통하여 바로 전송한다.

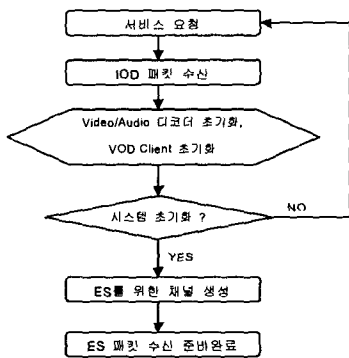


그림 5. 서비스를 위한 단말의 초기화 동작과정

논문에서 제안하는 내용	J.Deicke <i>et al.</i>
• 단말 : 서비스 요청	• 단말 : 서비스 요청
• 서버 : IOD 전송	• 서버 : IOD 전송
• 단말 : ES_Descriptor를 참조하여 시스템/디코더를 초기화한 다음 해당하는 패킷 수신 대기	• 단말 : OD 스트림을 위한 채널 추가
• 서버 : ES 전송	• 서버 : OD 스트림 전송
	• 단말 : OD 스트림을 참조하여 시스템/디코더 초기화한 다음 해당 패킷 수신 대기
	• 서버 : ES 전송

표 1. 논문에서 제안하는 IOD를 통한 초기화 동작 과정

2.3 버퍼관리와 동기화

IOD를 통한 초기화 동작이 완료되면 단말은 서버에게 Play UserCommand를 보낸다. 서버는 ES_ID를 참조하여 해당하는 객체의 FlexMux 패킷[1]을 전송한다. 단말의 DMIF에서 역다중화된 SL-패킷을 해당 DB에 입력하게 된다. 이때 단말의 Sync Manager는 SL-패킷에서 동기화 정보를 얻게 된다. 논문을 통하여 구현된 VOD 단말의 버퍼 아키텍처는 다음 그림 6과 같다.

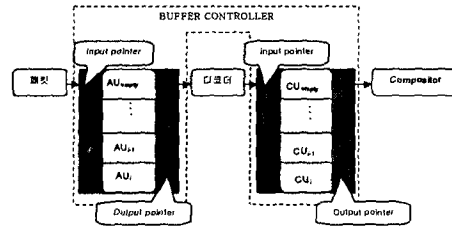


그림 6. 논문에서 구현된 VOD 단말의 버퍼

버퍼 관리에 있어 네트워크 상황 등에 따른 버퍼 언더플로우와 오버플로우에 잘 대응하는 모델을 작성하는 것이 무엇보다도 중요하다. 다음 그림 7은 버퍼 언더플로우 발생시 버퍼의 상태에 따라 단말의 시스템이 대응하는 모습을 보이고 있다.

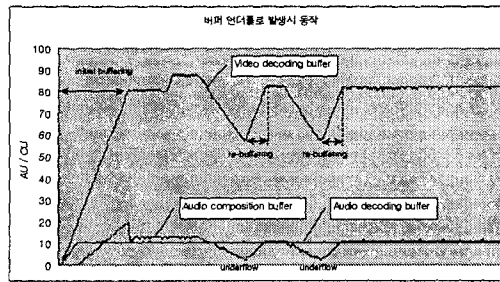


그림 7. 버퍼 언더플로우 발생시 동작

버퍼에 남아 있는 데이터의 양이 정해진 값보다 작을 경우 Buffer Controller가 Client Executive에게 언더플로우가 발생하였음을 알려주면, 현재 콘텐츠의 재생을 멈추고 초기의 버퍼링 양만큼 채워질 때까지 시스템을 잠시 멈춘다. 그림 7을 통하여 두 번째 언더플로우 발생 이후 언더플로우가 발생하지 않는 상황에서는 논문에서 설계되고 구현된 버퍼가 안정적으로 관리되고 있는 것을 알 수 있다. 오버플로우의 경우는 버퍼가 안정적인 양이라고 판단되는 순간까지 도달하는 패킷을 버림으로써 대응하게 된다.

각 DB에 입력된 Access Unit(AU)은 자신의 Decoding Time Stamp(DTS)에 도달하면 곧바로 복호화 되고 재생되거나 Composition Memory(CM)로 입력된다. 논문에서 제시하고 있는 VOD 단말은 비디오의 경우는 CM을 두지 않고 바로 복호화하고 재생하게 되지만, 오디오의 경우는 복호화된 웨이브 데이터를 Microsoft사의 DirectSound 객체를 생성하여 이 객체를 통하여 소리를 재생하게 된다. DirectSound는 Microsoft사의 다양한 멀티미디어 아키텍처 중 하나로써 간단하고 효율적으로 사운드 데이터를 재생할 수

있는 API를 제공하고 있다.

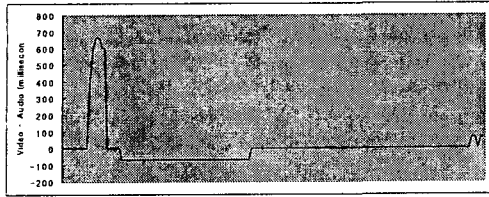


그림 8. 오디오와 비디오의 CTS/DTS를 이용한 동기화

그림 8은 Inter-Sync(또는 Lip-Sync)를 보여주고 있다. 객체간의 동기화를 위하여 주(master) 미디어로 사용되고 있는 것은 오디오이다. 그림 8을 통하여 Lip-Sync가 잘 맞고 있다는 것을 알 수 있다.

III. User interaction

VCR command라고도 하는 서버와 단말 사이의 상호 동작은 T-VOD를 실현하기 위한 가장 중요한 부분이라고 할 수 있다. 서론에서 언급한 바와 같이 MPEG-4는 오디오-비주얼 객체 기반의 부호화 기법을 적용하고 있기 때문에 다른 어떤 표준보다도 효율적인 사용자 상호동작을 제공할 수 있다. VCR command를 실현하기 위하여 비트스트림에 특별한 인덱스 구조를 삽입하거나[7], command 별로 각기 다른 스트림을 이용하는 방법[8]들이 제안되고 현재 사용되고 있다. 본 논문에서는 Sync Layer의 정보를 이용하여 인덱스 구조나 별도의 스트림이 없이도 Fast forward와 Move의 VCR command를 수행하는 방법을 제안한다.

$$T_{move} = T_{click} * SL_{timeStampResolution} \quad [수식 1]$$

먼저 Move의 경우는 위의 [수식 1]을 이용하여 T_{move} 를 계산한 다음 UP 채널을 통하여 서버로 전송하게 된다. 서버는 현재 전송되는 패킷의 DTS와 T_{move} 를

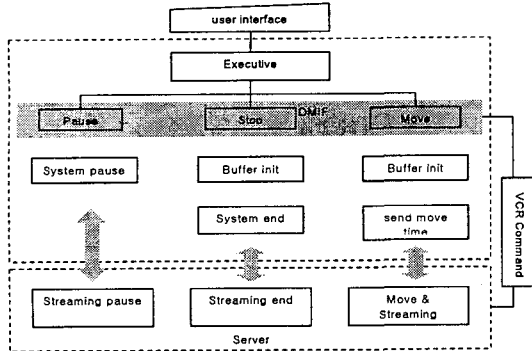


그림 9. VCR command의 동작 과정

비교하여 T_{move} 과 근사한 DTS를 가지는 FlexMux 패킷부터 전송한다. 이 때 단말에서는 현재 버퍼에 남아 있는 데이터를 모두 삭제하고 서버로부터 전송되는 새로운 패킷을 입력받아 동작을 수행하면 된다. Fast forward는 비디오의 I-VOP을 포함하고 있는 SL-패킷 헤더 정보의 randomAccessPointFlag가 설정이 되어 있는 FlexMux 패킷의 전송을 통하여 실현된다.

그림 9는 논문에서 구현된 서버와 단말 사이의 VCR command를 위한 동작의 과정을 보여주고 있다.

IV. 결론 및 향후 연구

본 논문은 MPEG-4 시스템의 장점들을 결합한 주문형 비디오 서비스를 위한 시스템의 구현을 보여주고 있다. 구현을 통하여 단말의 효과적인 버퍼 관리와 동기화를 위한 아키텍처를 제시하고 있다.

MPEG-4의 객체 기반 부호화를 이용한 Interactive 서비스의 장점은 다양한 멀티미디어 객체를 다룰 수 있다는 것, 새로운 interaction이 가능하다는 것, VCR functionality를 위하여 별도의 인덱스 파일이나 Fast forward 파일 등을 필요로 하지 않는다는 것들을 들 수 있는데 이는 논문에서 구현된 VOD 시스템의 장점이라고 말하는 것은 그리 어려운 일이 아닐 것이다.

현재 MPEG-4 시스템은 Version 1이 완성된 상태이고 Version 2에 대한 표준화 작업이 이루어지고 있다. Version 1은 Binary Format for Scenes(BIFS)와 OD, MPEG-4 file format 등에 대하여 논의되었고, Version 2에서는 확장된 BIFS와 Java 지원(MPEG-J)에 대하여 논의되고 있다.

앞으로 단일 객체에 대한 VOD 서비스 뿐만 아니라 다중의 객체를 서비스 할 수 있도록 구현할 것이며, BIFS에 대한 구현 및 연구를 진행하도록 할 것이다.

참고 문헌

- [1] ISO/IEC 14496-1:1999(E), Information technology Coding of audio-visual objects Part 1: Systems.
- [2] ISO/IEC 14496-6:1999(E), Information technology Coding of audio-visual objects Part 6: Delivery Multimedia Integration Framework(DMIF).
- [3] MPEG-4 Standards ISO/IEC JTC1/SC29/WG11 N3747, Overview of the MPEG-4 Standard, October 2000
- [4] O. Avaro, P. A. Chou, A. Elftheriadis, C. Herpel, C. Reader, J. Signes, The MPEG-4 systems and description languages: A way ahead in audio visual information representation, Signal Processing: Image Communication 9 (1997) 385-431.
- [5] C. Herpel, A. Eleftheriadis, MPEG-4 Systems: Elementary stream management, Signal Processing: Image Communication 15 (2000) 299-320.
- [6] J. Deicke, U.mayer, M.Glesner, A client/server application as an example for MPEG-4 systems, Computer Communications 21 (1998) 1302 - 1309
- [7] Doh Sang-Yoon; Min Jang, Fast forward and fast rewind play system based on the MPEG system stream with new concept, Multimedia Computing and Systems, 1999. IEEE International Conference on, Volume: 2, 1999, 846 850
- [8] Andersen, D.B, A proposed method for creating VCR functions using MPEG streams, Data Engineering, 1996. Proceedings of the Twelfth International Conference on, 1996