

# Java 기반 MPEG-4 스트리밍 클라이언트 개발

백국실<sup>1</sup> 원유집<sup>2</sup> 심보욱<sup>2</sup>  
한양대학교 정보통신 대학원 정보통신공학과<sup>1</sup>  
한양대학교 전자전기컴퓨터 공학부<sup>2</sup>  
{bogus,yjwon,shim77}@ece.hanyang.ac.kr

## Development of JAVA based MPEG-4 Streaming Client

Guksil Baek<sup>1</sup>, Youjip Won<sup>2</sup>, Bowook Shim<sup>2</sup>  
GIC, Hanyang University<sup>1</sup>  
ECE Division, Hanyang University<sup>2</sup>

### 요약

본 논문에서는 MPEG-4의 표준 압축/복원방식을 기반으로 한 멀티미디어 스트리밍 클라이언트의 설계 및 구현을 기술한다. 스트리밍 프레임 워크 내에서 클라이언트가 QoS 관리를 위해서 할 수 있는 성능적 측면을 기술한다. JAVA를 기반으로 구현한 클라이언트 내부 시스템 설계 및 구현부분과 네이티브 코드로 구성된 MPEG-4 표준 코덱과 RTP/RTSP/RTCP 부분을 JNI를 통해서 연동되는 기술적 측면을 보여준다. 스트리밍 프레임 워크 내에서 클라이언트의 향후 발전해야 할 과제를 제시한다.

### 1. 서론

스트리밍(streaming)이란 인터넷을 통하여 영상, 음성 등 디지털 멀티미디어 데이터를 실시간으로 전송하여 받아 볼 수 있는 기술을 말한다[1]. 멀티미디어 스트리밍 기술은 PDA, 유무선 통신단말기 등의 멀티미디어 임베디드 시스템과 리눅스 데스크탑에 적용되어 가상교육 시스템, 동영상 광고, IMT2000 동영상 전송, 화상 전화 등 다양한 분야에서 활용되어 질 수 있다[4]. 본 논문에서는 차세대 압축/복원 표준 방식인 MPEG-4를 이용한 멀티미디어 스트리밍 서비스를 받을 수 있는 클라이언트를 제안한다. 현재까지의 MPEG-4 스트리밍 클라이언트들은 서로 다른 플랫폼을 가지고 있는 사용자에게 플랫폼에 종속적인 클라이언트를 통해 서비스를 하고 있다[5]. 또한 멀티미디어 서비스의 QoS보장을 위해 클라이언트의 리소스 정보를 스트리밍 서버에 전송하거나 네트워크 트래픽을 모니터링 할 수 있는 클라이언트도 없다. 따라서 클라이언트가 플랫폼에 독립적이고, 네트워크 트래픽을 모니터링 할 수 있는 클라이언트를 제안한다.

### 2. 시스템 구조

MPEG-4 시스템 표준에서 제안된 클라이언트 구조를 보면 스

트리밍 서버로부터 MPEG-4 미디어 데이터가 DAI(DMIF Application Interface)를 통해서 전송된다. 전송된 데이터는 동기화 계층(Sync Layer)에 있는 각각의 데이터 채널을 통해 AU(Access Unit)로 변화되어 디코딩 버퍼에 저장 된다.

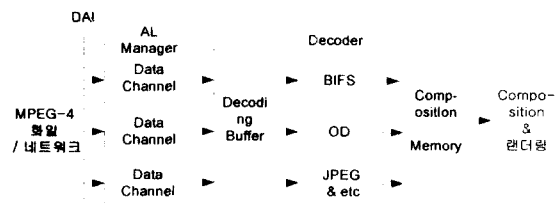


그림 1. MPEG-4 시스템 표준

디코딩 버퍼에 저장된 자료는 각 오브젝트에 최적화된 디코더를 통해 디코딩이 된다. 마지막으로 이 데이터는 컴포지션 메모리에 저장되어진 후 컴포지터에 의해 렌더링된다 [4,7,8].

#### 2.1 클라이언트 시스템 구조

클라이언트 시스템은 C/C++로 구현된 RTP 패킷을 받는 부분. 이 패킷을 분석하여 하여 디코딩 버퍼에 각각의 프레임 단위로 저장시키는 부분, 코덱 라이브러리를 이용하여 디코딩 하는 부분 그리고 디코딩 된 미디어 데이터를 컴포지션하여 화면에 재생시켜주는 크게 네 부분으로 나누어진다.

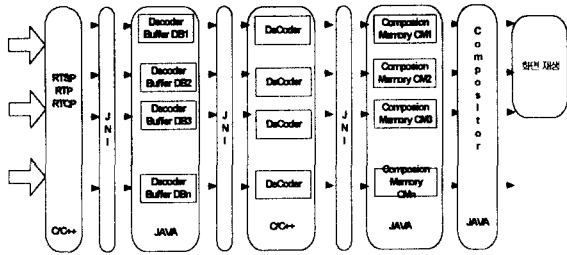


그림 3 클라이언트 시스템 구성도

그림 3의 시스템 구성도는 서버로부터 클라이언트까지 멀티미디어 데이터가 전송 후 재생되는 흐름을 보여준다. 클라이언트에서 멀티미디어 서버에게 RTSP 프로토콜을 이용하여 서비스를 요청하면 서버로부터 MPEG-4 데이터가 DAI(DMIF Application Interface)를 통해 전송된다. 그리고 전송된 RTP Streaming 패킷은 동기화 계층의 데이터 채널을 통해 디코딩 버퍼에 저장된다. 동기화 계층에서는 동기화 계층 처리 쓰레드가 클라이언트의 UDP 버퍼에 저장된 RTP 패킷을 가져와서 분석한 후 프레임단위로 디코딩 버퍼에 저장한다. 그 후 미디어 데이터는 AU(Access Unit)로 변환되어 디코더에 의해 프레임 단위로 디코딩 되어진 후 CS(Composition Unit)로 변환되어 컴포지션 메모리에 저장된다. 마지막으로 저장된 CS는 컴포지터에 의해 렌더링되어 화면에 재생된다.

2.2 클라이언트 쓰레드 구조

클라이언트에서 쓰레드를 적용한 부분은 RTP 패킷을 받는 부분과 동기화 계층부분, 디코딩 부분, 컴포지션 부분이다. 각각의 계층(layer)마다 쓰레드로 두고, 계층사이에는 버퍼를 두었다. 이렇게 하여, 한 계층에서 프로세스 시간이 오래 걸리더라도 다른 계층의 처리를 하지 못하고 블록 되는 현상을 방지하였다. 이처럼 계층간에 멀티 쓰레드 모델링을 적용함으로써 분석시간, 디코딩 시간 그리고 재생시간을 효율적으로 할당하여 성능을 최적화할 수 있게 되었다.

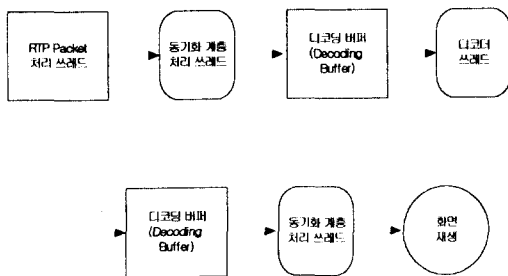


그림 4 쓰레드 모델링

그림 4는 클라이언트 시스템에서 쓰레드 구조를 나타낸다. RTP 패킷 쓰레드는 UDP/TCP 소켓과 패킷을 받는 수신 버퍼를 생성하여 스트리밍 서버로부터 RTP 패킷을 전송받아 버퍼에 저장한다. 동기화 계층 쓰레드는 수신 버퍼로부터 패킷을 얻어와서 패킷에 헤더 정보만을 빼고 프레임 단위로 디코딩 버퍼에 저장한다. 이 프로세스 중, 패킷 헤더 정보중에서 CTS정보와 DTS정보는 비디오/오디오의 동기화를 위해 디코더와 컴포지터에 넘겨지게 된다. 동기화 계층 프로세스 작업이 끝나면 디코딩 쓰레드는 저장된 미디어 데이터를 MPEG-4 표준 복원 코덱을 이용하여 프레임 단위로 디코딩한 후 이를 컴포지션 버퍼에 저장한 후에 대기한다. 컴포지션 쓰레드는 컴포지션 버퍼에서 객체단위로 컴포지터에게 넘겨주어서 화면에 재생 시킨다. 쓰레드간의 동기화는 자바언어에서 제공되는 Synchronized라는 키워드를 이용하여 행해진다. 디코딩과 컴포지션 쓰레드의 작업은 프레임 단위로 이루어진다[1].

2.3 비디오/오디오 동기화 기법

MPEG-4 미디어 데이터가 서버로부터 인코딩 되어 전송될 때 모든 패킷들은 헤더 정보에 DTS와 CTS 정보를 가지고 있다. DTS(Decoding Time Stamp)는 디코딩 되어야 할 시간 정보를 저장하고 있으며, CTS(Composition Time Stamp)는 컴포지션 되어야 할 시간 정보를 의미한다.

디코딩 쓰레드는 현재 시간과 전송된 프레임의 DTS 정보를 비교하여 디코딩 프레임을 적절하게 제거한다. 컴포지션 쓰레드는 CTS 정보와 현재 시간을 비교하여 프레임을 적절하게 제거(Drop)한다.

MPEG-4 IP[9]의 경우, 디스플레이 시간이 현재 시간보다 10프레임 이상 나중에 있으면 1프레임으로 동기화를 이루고 3프레임 이상 10프레임 미만인 경우에는 8프레임만 제거한다. 화면 재생부분은 더블 버퍼링을 통한 렌더링 기법을 사용하여 디스플레이 시간을 최적화해서 성능을 향상시킨다.

3. JAVA 와 네이티브 코드와 연동

본 논문에서 제안된 클라이언트에서 데이터 송,수신(Network Transport)부분, 이벤트 핸들링 메시지(Play, Stop, Pause 그리고 여러 가지 QoS 인자)를 전송하는 부분은 C/C++로 구현되어져 있다. 클라이언트의 코덱 부분도 C/C++로 구현되어져 있다. 이러한 네이티브 코드와 클라이언트의 개발 언어인 자바와 연동시키기 위해 JNI(Java Native Interface)가 이용되었다. JNI는 서

로 자료구조가 다른 언어와 자바가 연동하기 위한 인터페이스를 제공해주는 것으로서 기본적으로 원시코드를 통한 C 함수를 호출한다. 먼저 자바 메소드를 선언하고 원시코드를 포함하고 있는 라이브러리를 메모리에 적재(Loading)하고 네이티브 코드를 이용할 수 있다. 네이티브 선언은 자바의 가상 머신 내에서 원시 함수를 호출할 수 있는 교차역할을 제공해 준다[2,6]. 이 기술을 가지고 클라이언트에서의 RTP 전송 부분과 코덱 부분을 구현하였다.

4. 성능적 측면의 고려

SMART 클라이언트는 플랫폼에 독립적인 클라이언트를 통해 스트리밍 서비스를 받을 수 있도록 클라이언트의 UI와 시스템 부분의 쓰레드를 자바로 구현하였다. 또한 QoS 관리를 위해 클라이언트는 리소스 정보를 가지고 있는 QoS 인자를 스트리밍 서버에 전송한다. 클라이언트는 다양한 대역폭(Bandwidth)을 스트리밍 서버에게 요청하거나 가변적인 대역폭으로 스트리밍 서비스를 받을 수 있다. 클라이언트 시스템의 모든 계층에 쓰레드를 사용하여 성능 최적화를 이루었다. 네이티브 코드와 JAVA와의 연동에서 서로 다른 자료형태로 인해 자료 변환과정에서 성능을 저하 시키는 요인은 앞으로 더 연구 되어져야 할 과제이다.

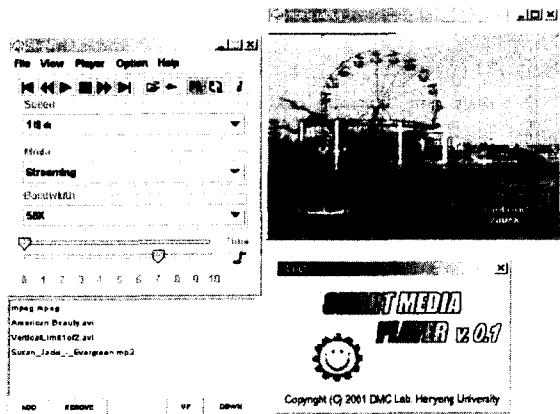


그림 5 클라이언트 구현 화면

그림 5는 윈도우 환경의 클라이언트가 스트리밍 서비스를 받고있는 화면이다.

5. 결론 및 향후 과제

인터넷 방송이나 주문형 비디오 등과 같은 멀티미디어 서비스를 위한 기술에서 효율적인 서버 기술, 네트워크 기술과 함께 다양한 기능을 가진 클라이언트 기술이 필요 되어 지고 있다.

본 논문에서는 플랫폼 독립적인 클라이언트 모델을 제시하였다. 앞으로 MPEG-4 스트리밍 서비스를 받을 수 있는 클라이언트에서 발전되어야 할 몇 가지 과제가 남아 있다. 2차원 비디오나 오디오 뿐만 아니라 3차원 영상 데이터나 facial 이미지 등을 클라이언트에서 재생 할 수 있는 시스템과 유선 환경이 아닌 무선 환경에서도 유선 환경과 동일한 멀티미디어 서비스를 받을 수 있는 클라이언트의 개발이 앞으로 요구되어진다.

6. 참고문헌

[1] Robert Cohen and Radha " Streaming Fine-Grained Scalable Video over Packet-Based Networks" Global Telecommunications Conference, 2000. LOBECOM '00. IEEE, Volume:1, 2000 Page(s): 288 -292 vol.1

[2] <http://java.sun.com/products/jdk/1.2/docs/guide/jni/spec/jniTOC.doc.html>

[3] <http://java.sun.com/products/java-edia/jmf/index.html>

[4] Aggeliki Prayati, Chun Wong, Paul Marchal, Nathalie Cossement, Francky Catthoor, Rudy Lauwereins, Diederik Verkest Hugo De Man " Task concurrency Management experiment for power-efficient speed-up of embedded MPEG4 IM1 Player" Parallel Processing, 2000. Proceedings, 2000 International Workshop on, 2000 Page(s):453-460

[5] Viswanathan Swaminathan Gerard Fernando " MPEG-J: Java Application Engine in MPEG-4" Circuits and Systems, 2000. Proceedings, ISCAS 2000 Geneva, The 2000 IEEE International Symposium on , Volume: 1 2000 Page(s): 44 -47 vol.1

[6] <http://java.sun.com/docs/books/tutorial/native1.1/TOC.html>

[7] A Puri, RL Schmidt, and BG Haskell. " Scene description, composition, and playback systems for MPEG-4" . proceeding Visual Communications and Image Processing, San Jose, January, 1999

[8] ISO/IEC JTC1/SC29/WG11 N2523. IMI Tools, October 1998.

[9] <http://mpeg4ip.sourceforge.net/>