

# RTP 기반 실시간 MPEG2 스트림 전송 시스템

박 대훈(朴 大勳), 홍 윤식(洪 琬植)  
 인천대학교 공과대학 컴퓨터공학과  
 전화 : (032) 770-8495 / 팩스 : (032) 766-6894

## A MPEG2 stream transport systems based on Real-time Transport Protocol

Dae-Hoon Park, Youn-Sik Hong  
 Department of CE and CS, University of Incheon  
 E-mail : parkdh@itic.re.kr, yshong@lion.inchon.ac.kr

### Abstract

Current Internet and Intranet developments are focusing on network multimedia services involving the transport of real-time multimedia streams. In this paper, we present a real-time MPEG2 Encoding/Decoding systems and a MPEG2 stream player running on client systems based on real-time transport protocol(RTP). The MPEG2 player has been implemented by using JMF and java applet. We have tested our prototype systems with one of possible multimedia sources, camcorder, to verify its functional correctness and measure its performance.

### 1. 서론

인터넷환경에서의 MPEG2동영상 데이터전송은 최근 몇 년동안 관련기술이 발전함에 따라 새롭게 부각되고 있다. 일반 사용자들은 케이블모뎀 등을 비롯한 고속전송망의 대역폭이 증가하면서 고속의 전송속도로 동영상 데이터를 송수신할수 있게 되었다. MPEG2 Encoder의 하드웨어 가격이 하락하여 PC에 MPEG2 Encoder를 부착하는 사례가 늘어나는 추세이며, PC의 성능도 동영상 데이터를 압축 전송할 수 있을 정도로 크게 향상되었다. IETF에서 개발된 RTP(Realtime transport protocol)프로토콜은 이와 같은 압축전송을 표준화 할

수 있다. 본 연구에서는 MPEG2 동영상 데이터를 Encoding 및 Decoding할 수 있는 시스템 구현과 RTP/RTCP 방식으로 동영상 스트림을 실시간으로 전송하기 위한 Server-Client 환경을 구축하고자 한다.

### 2. RTP/RTCP Protocol

RTP[1,2]는 실시간으로 패킷을 전송할수 있도록 설계된 프로토콜이다. 또한 TCP기반이 아닌 UDP 기반으로 설계되었기 때문에 신뢰성있는 전송을 보장하지 못한다.

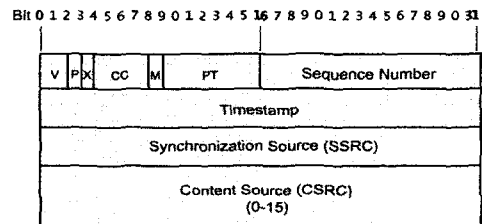


그림 1. RTP 패킷 헤더의 구조[3]

- V : RTP의 버전정보를 나타내며 현재 Version은 2이다
- X : RTP 헤더의 추가정보가 있을 경우 X bit는 set된다

- CC : CSRC의 개수를 표시하는 항목으로 Audio정보가 Mixing되었을 때 RTP Packet의 SSRC 정보를 CSRC에 포함시키는 경우 포함시킨 SSRC의 개수의 확인을 위해 사용된다
- M : market bit의 해석은 payload 형식에 의존한다. 예를 들어 비디오 payload에서는 프레임의 끝을 표시하고, 오디오에서는 talks의 시작을 표시한다.
- PT : 1-byte payload 타입으로 packet에 들어있는 payload의 종류를 식별한다(JPEG video 또는 GSM audio).
- Timestamp : 32-bit Timestamp는 데이터가 패킷에 저장되는 순간을 기록한다. Timestamp 값은 payload에 따라 다르다.
- Sequence Number : 16-bit packet sequence number는 패킷의 분실여부를 발견하는데 사용되고 Timestamp를 가진 패킷들의 순서를 정한다.
- Synchronization source identifier(SSRC) : 섹션 안의 자원을 유일하게 식별하는 불규칙적으로 생성된 32-bit 숫자.

RTCP프로토콜은 RTP의 전송상태를 확인할수 있도록 설계된 프로토콜이다. RTCP에서 제공하는 주요 기능은 다음과 같다.

- Provide information to application(SR, RR) : 데이터의 질을 고려하면서 어플리케이션에 정보를 제공하는 기능을 한다. 각 RTCP 패킷은 전송자, 수신자 보고를 갖는다. 전송된 패킷, 손실된 패킷등의 통계도 갖고 있다.
- Identify RTP source(SDES) : RTP 소스의 트랜스포트 계층의 구별자를 전송한다 CNAME이라고 하는 이 구별자는 RTP 섹션에 참여하는 사용자를 추적한다. 사용자는 RTP 섹션에 관련된 집합에 참여하는 참여자에게 여러 개의 데이터를 연결시킨다.
- Convey minimal session control information(APP) : 선택 기능으로 섹션의 모든 참여자에게 작은 양의 정보를 전송하는데 RTCP를 사용할 수 있다. 예를 들면 사용자 화면 상에서 참여자를 구별하기 위해 개인 이름을 전송할 수 있다.

또한 RTCP Packet은 송수신자 보고 패킷,소스 디스크립션, APP 패킷의 세가지 패킷을 가질수 있다. 송수신자 보고패킷은 송수신된 패킷수와 잃어버린 패킷수를 확인한다. SR과 RR의 정보로 네트워크의 상태와

RTP의 전송량을 확인할수 있다. SDES Packet의 각 항목은 자신을 구별하는 소스를 서술하는 항목들로 구성된다. APP 패킷은 패킷 타입 값을 등록하지 않고 새로운 어플리케이션을 실험할 경우 사용하는 것을 목적으로 하고 있다. 공식 패킷 타입을 갖는 완전한 RTCP 패킷으로 쉽게 변환된다.

### 3. 개발환경

본 연구에서는 MPEG2데이터를 Encoding하기 위하여 iCompression사의 iVAC Chip을 사용하여 Encoding System을 구현하였으며, Decoding을 위해서는 Oak사의 OTI8211 Chip을 사용하였다. 이 시스템의 구동 드라이버는 Windows98에서 Visual C++을 사용하여 구현하였다.한편 동영상 입력원으로는 Camcoder (삼성 SV-K85)를 사용하였다. 실험에 사용된 시스템의 구성도(데이터 흐름도)를 그림 2에 나타내었다.

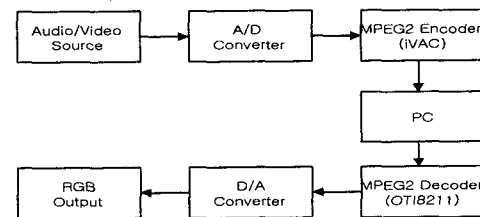


그림 2 Encoding 및 Decoding시스템 구성도

### 4. JMF기반의 RTP Player

그림 3과 같이 실시간으로 Encoding된 데이터를 사용자의 PC에 전송하는 방법으로 Java기반의 JMF(Java Media Framework)[4]를 사용하여 RTP방식으로 전송하였다.

JMF에서는 Capture Device가 추상화되어있어서 사용자가 별도의 프로그램을 하지 않아도 Capture Device를 사용할수 있다. Capture Device에서 선택된 각종 포맷을 JMF에서 사용하기 위하여 DataSource객체를 사용하여 각종 영상원을 추상화한다.

Processor객체는 DataSource객체를 입력원으로 삼아 각종 포맷의 형식변환 및 데이터의 추가 삭제와 Audio와 Video의 분리 및 Mixing이 가능하도록 설계되어 있다. 이 과정에서 DataSource가 RTP데이터 형식으로 변환되어 Network에 전송될수 있는 Packet으로 생성된다. Processor객체에서 Encoding 스트림의 출력은 다시 DataSource객체로 출력된다. 이 객체를 파일로 저장하고 싶을 경우 DataSink객체를 통하여 원

하는 파일 형식으로 저장매체에 저장가능하고 RTP 전송을 원하는 경우 SessionManager를 통하여 RTP 프로토콜로 전송 및 RTCP프로토콜로 전송할수 있다.

Client에서 RTP Packet을 전송받기 위해서는 SessionManager객체를 생성하고 객체에 포함되어 있는 각종 Event를 처리함으로써 RTP Packet을 DataSource객체로 변경 가능하다.

DataSource객체를 화면상에 출력시키기 위하여 Player객체의 입력을 DataSource객체로 지정하면 화면에 표시되는 애플릿의 화면이 구성된다. Player객체는 Visual Component와 Control Component로 구성되어 있는데 Visual Component는 화면상에 출력되는 영상 및 음성을 표시해주는 객체이고 Control Component는 RTP Packet의 상태와 음성과 재생비율등을 설정해 줄수 있는 객체이다.

JMF Client에서는 Web Browser상에서 Java Applet으로 Player를 구현하기 때문에 별도 프로그램을 설치하지 않아도 RTP Player를 실행시킬 수 있다.

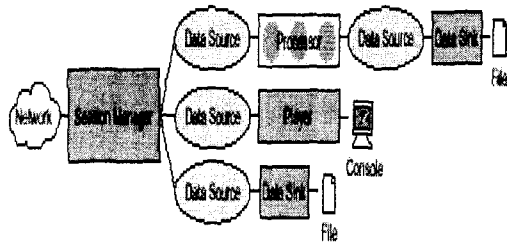


그림 3 JMF기반에서의 RTP전송

본 연구에서는 iVAC Chip을 사용하여 MPEG2 데이터 소스를 획득하였고 이를 바탕으로 RTP Server와 RTP Client를 구현하였다. RTP서버에서 Client의 요청이 있을 때 MPEG 데이터를 RTP Packet으로 변형시켜 전송하였고 Client에서 RTP Packet을 수집하여 Web Browser에 표시하였다.

RTP Server는 Multicasting으로 데이터를 전송하여 네트워크의 부하를 줄일수 있도록 구성하였고 별도로 Point-to-Point 방식의 데이터 전송이 실시될수 있도록 JMF의 Cloned DataSource를 사용하여 구성하였다.

### 5. 실험 결과

iCompression사의 iVAC Chip을 사용한 Window 98용 Driver(소프트웨어)의 개발을 완료하였고 이를 사용한 실시간 Encoding 및 Decoding 결과를 그림 4에 나타내었다. JMF의 RTP방식전송은 MPEG1 스트림만

지원하기 때문에 MPEG2 스트림의 RTP전송을 위한 Multiplexer와 Demultiplexer의 개발은 현재 구현되지 않았다.

RTP서버가 실행되어 Client에 반응되는 화면은 그림 5에 나타내었고 Web Browser에서 표시되는 RTP Client는 그림 6에 표시하였다. RTP Client에서 Packet의 전송상태를 표시하는 RTCP Packet의 상태는 그림 7에서 확인할수 있다.



그림 4 실시간 Encoding 및 Decoding 화면

실시간 Encoding 및 Decoding은 Visual C++을 사용하였고 미들웨어로 Platform SDK를 사용하였다. 이를 이용한 결과화면은 그림4와 같이 나왔고 MPEG2 Data를 실시간으로 Encoding 및 Decoding하였다.

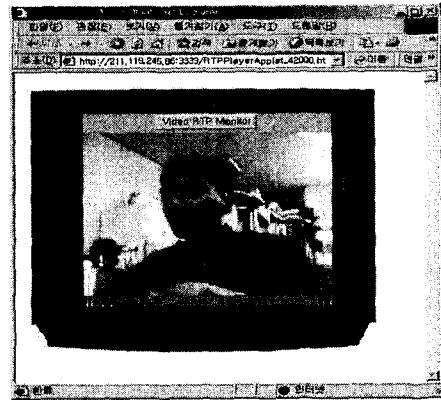


그림 5 RTP Client의 실행화면

Client는 JMF2.0을 이용한 Applet으로 구성되었다. 서버에서 전송해주는 RTP Packet을 화면에 출력해주는 부분을 담당하고 동시에 서버에 RTP Packet의 수신상태를 전달해주는 역할을 담당한다

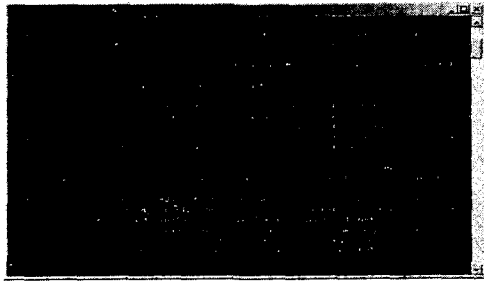


그림 6 RTP Server의 실행화면

RTP Server는 Java Application으로 작성되었다. 각종 포맷의 파일과 Capture Device의 데이터를 RTP 형식으로 전송한다. 클라이언트는 그림 7과 같은 RTCP정보를 수신한다.

General RTP Statistics	
Total Packets Recd	429
Total Bytes Recd	365530
Bad RTP Packets	0
Local Collisions	0
Remote Collisions	0
Packets Looped	0
Failed Transmissions	0
RTCP Packets Recd	53
SR Packets Recd	26
Bad RTCP Packets Recd	0
Unknown RTCP Types	0
Malformed RR	0
Malformed SDES	0
Malformed BYE	0
Malformed SR	0
<input checked="" type="checkbox"/> Always on Top	

그림 7 . RTCP패킷정보

JMF를 이용한 RTP전송결과 몇가지 문제점을 발견할수 있는데 현재 JMF는 MPEG1의 RTP 전송만 지원하여 MPEG2의 RTP전송을 위해서는 Multiplexer와 Demultiplexer를 각각 구현해야 하는 문제점이 있었다. 또한 JMF의 Class가 Heavy Component이기 때문에 Applet만 가지고 화면을 표시할수 없고 Client에 JMF를 수동으로 설치해야 하는 문제점이 발견되었다. 마지막으로 Java로 RTP Packet을 전송시 실행속도가 매우 늦어 5 Frame/Sec의 낮은 전송속도를 측정하여 WebCasting이용시 다소 부적합함을 확인하였다 (320 X 200, 16bit Color).

이의 개선책으로 RTP Server측이 현재 Java Application으로 구현되어 있어 실행속도의 한계를 가지고 있다. 따라서 Server측의 개발환경을 Visual C++을 이용한 환경으로 전환하고 MPEG Data의 H/W

Encoding을 이용하면 속도개선이 될 것이다.

## 6. 향후 연구 방향

인트라넷 환경에서 MPEG2 동영상 데이터의 실시간 Encoding 및 Decoding이 가능한 시스템을 구현하였다. 이 결과를 바탕으로 클라이언트에서 RTP 방식의 동영상 스트림 출력을 위한 MPEG 재생기를 java의 JMF를 사용하여 구현하였다. 실험결과를 바탕으로 Encoder부분을 H/W Codec을 사용한 실시간 Encoding으로 처리하고 RTP Server부분을 Visual C++개발환경의 RTSP서버로 구성하고 Client환경을 JMF를 사용한 Applet으로 구현하여 실시간 실행속도 구현을 목표로 하고 있다.

## 참고문헌

- [1] M.R. Civanlar, G.L. Cash, B.G. Haskell, RTP Payload format for bundled MPEG, RFC2043, I.D. 1998
- [2] H.Schulzrinne, RTP profile for audio and video conferences with minimal control, RFC1890, January 1998
- [3] H.Schulzrinne, S.Casner, R. Frederick,V.Jacobson, RTP: a transport protocol for real-time applications, RFC1889
- [4] Sun, Java Media Framework API Guide, November 1999
- [5] Mark A. Miller, P.E. Voice over IP, M&T Books