

# RTSP를 이용한 인터넷 VCR의 설계와 구현

황석현, 장주욱\*

서강대학교 전자공학과

## Design and Implementation of an Internet VCR using RTSP Protocol

Suk-hyun Hwang, Ju-wook Jang

Dept. of Electronic Engineering, Sogang Univ.

### 요약

본 논문에서는 스트리밍 방식에 기반한 실시간 인터넷 비디오 VCR 서버 및 클라이언트를 설계하고 구현하였다. 구현된 시스템은 음성과 영상이 포함된 MPEG-1 시스템 스트림을 소프트웨어적으로 처리하고 있으며, MPEG 스트림의 실시간 전송과, 재생, 일시 정지, 위치 이동, 재생 속도 조절 등의 VCR 기능을 위해 각각 RTP 프로토콜과 RTSP 프로토콜을 구현하여 사용하고 있다. 또한, 본 시스템에서는 고속 재생의 구현을 위해 MPEG 영상 스트림 중 인트라 프레임만을 추출하여 전송하는 방식을 채택하고 있으며, 이때 고속 재생을 위한 별도의 MPEG 파일 제작과 같은 추가적인 준비 작업을 필요로 하지 않는다는 특징을 가지고 있다. 구현된 시스템은 서버와 클라이언트 모두 PC 상에서 구현되었으며, 랜 환경에서 원활한 MPEG 스트림의 전송, 재생 및 제어가 이루어짐을 확인하였다.

## 1. 서론

인터넷을 통해 영상을 전송하는 방식에는 다운로드 방식과 스트리밍 방식의 두 가지가 있다. 이 중 다운로드 방식은 사용자가 긴 다운로드 시간 동안 기다려야 재생을 시작할 수 있다는 문제점과, 클라이언트 측에 다운로드한 영상 파일 전체를 저장할 수 있는 방대한 저장 공간이 준비되어 있어야 한다는 문제점을 가지고 있다. 스트리밍 방식은 영상을 전송 받으면서 동시에 재생이 이루어지는 방식으로, 이와 같은 재생 전의 긴 다운로드 시간을 필요로 하지 않으며, 재생에 필요한 최소한의 저장 공간만을 필요로 한다는 장점으로 인해 현재 다양한 연구 및 개발이 이루어지고 있는 상황이다.

이와 같은 스트리밍 기술의 발전은 최근 리얼 넷 워크스와 넷스케이프가 주축이 되어 제안한 스트리밍 프로토콜인 RTSP(Real Time Streaming Protocol)의 등장으로 더욱 가속화될 것으로 보인다.

현재, 이 RTSP를 응용한 다양한 스트리밍 연구 및 제품 구현이 이루어지고 있는데, 대표적인 예로는 유럽 입자 물리 연구소인 Cern의 Wrtsp 프로젝트와, WrtspVoD 프로젝트, W3C의 Jigsaw, 그리고 RealNetworks의 RealPlayer를 들 수 있다. Wrtsp 프로젝트는 RTP 화상회의 세션을 원격으로 저장하고 재생하기 위한 프로젝트로서, RTSP를 이용하여 원격 제어를 구현하고 있다. WrtspVoD는 멀티미디어 콘텐츠를 네트워크를 통해 스트리밍 방식으로 제공해주는 프로젝트로서, 마찬가지로 RTSP를 기반으로 하여 원격 제어를 구현하고 있다. Jigsaw는 W3C에서 자바 언어로 개발하고 있는 웹 서버로서 현재 RTSP 지원 모듈이 포함되어 개발되고 있는 상황이다. RealPlayer는 RTSP를 제안한 회사 중의 하나인 RealNetworks에서 개발한 비디

\* 본 연구는 98년도 정보통신부 국책기술 개발 사업의 지원을 받았음

오 스트리밍 프로그램으로서 역시 RTSP를 이용한 원격 제어에 기반하고 있다.

위와 같이 현재 다양한 관련 연구 및 제품들이 나오고 있는 상황이다. 이들 중 대표적인 스트리밍 제품으로는 Microsoft NetShow와 RealNetworks의 RealPlayer를 들 수 있다. 다음은 이들과 본 논문에서 구현한 시스템을 비교한 것이다.

	Microsoft NetShow	RealNetworks RealSystem	본 논문
서버 플랫폼	Windows NT	UNIX, Windows NT	Windows 95/98/NT
클라이언트 플랫폼	Windows 3.1/95/98/NT Mac, UNIX	UNIX, Windows 3.1/95/98/NT	Windows 95/98/NT
비디오 형식	ASF	Real Media	MPEG
스트림 전송	TCP, UDP, HTTP	TCP, UDP, HTTP	RTP
스트림 제어	비표준	표준(RTSP)	표준(RTSP)
재생속도제어	미지원	미지원	지원

표 1. 다른 스트리밍 시스템과 본 논문과의 비교

본 논문에서는 이상과 같은 스트리밍 방식을 채택한 인터넷 VCR을 설계, 구현한 뒤, 동작을 확인해 보았다. 스트리밍 방식은 다운로드 방식과는 달리 다음과 같은 몇 가지 기술적인 문제들을 갖고 있다. 우선, 실시간 전송에 적합한 전송 프로토콜이 필요하게 된다. 웹을 비롯한 기존의 인터넷 응용에서 주로 사용하고 있는 전송 프로토콜인 TCP는 패킷의 재전송과 같은 부하가 큰 과정들로 인해 멀티미디어 전송에 적합하지 않다는 문제점을 가지고 있으며, 부하가 작은 전송 프로토콜인 UDP 역시 패킷들간의 순서나 시간 관계 등을 기술할 수 없다는 문제점을 가지고 있다. 다음으로, 스트리밍 시에 재생, 일시 정지, 위치 이동, 재생 속도의 조절과 같은 사용자의 요구를 서버 측에 전달할 수 있는 방법이 필요하게 된다. 그리고 고속 재생 시에 높아지는 대역폭문제를 어떻게 처리할 것인가의 문제도 있다. 본 논문에서는 이러한 문제에 초점을 맞춘 표준에 기반한 시스템의 설계 및 구현을 목표로 하고 있다

논문의 구성은 다음과 같다. 2장에서는 제안된 VCR 시스템의 전반적인 구조를 먼저 설명한 뒤, 각 세부 모듈들의 특징과 구조에 관하여 설명하겠다. 다음으로 3

장에서는 시스템의 구현 환경 및 구현 결과를 살펴보고, 4장에서는 결론과 향후 방향을 기술하겠다.

## 2. Internet VCR의 설계

본 논문에서 구현한 시스템은 크게 HTTP 모듈, RTSP 모듈, RTP 모듈, 그리고 MPEG 모듈의 4 모듈로 구분할 수 있다. 이를 도시하면 그림 1과 같다.

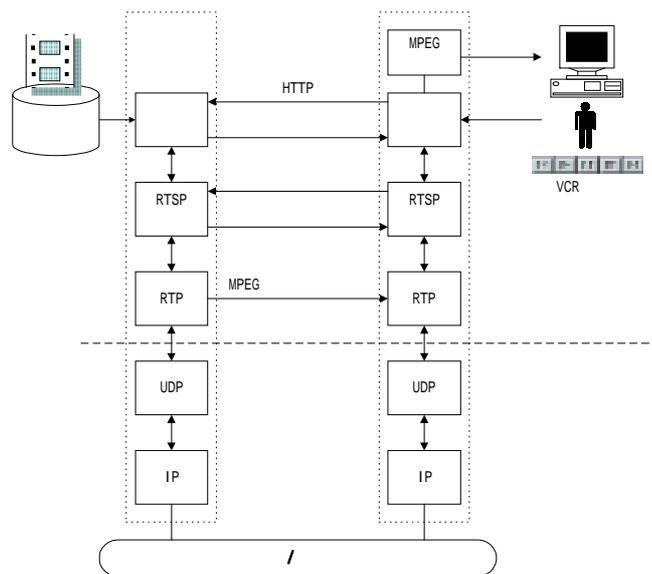


그림 2. 시스템의 전체 구성도

우선, HTTP 모듈은 세션을 초기화하고, RTSP 클라이언트와 서버를 연결시켜주는 역할을 담당하고 있다. 두 번째로, RTSP 모듈은 RTP 모듈을 생성하고 초기화하는 역할과 스트림을 제어하는 역할을 담당하고 있다. 세 번째 모듈인 RTP 모듈은 MPEG 스트림을 서버에서 클라이언트로 전송하는 역할을 담당하고 있으며, 마지막 모듈인 MPEG 모듈은 서버 측에서 MPEG 스트림의 송신 전에 필터 처리를 하는 역할과, 클라이언트 측에서 수신된 MPEG 스트림을 재생해서 사용자에게 보여주는 역할을 담당하고 있다.

### 2.1 RTP를 이용한 비디오 스트림의 전송

TCP/IP 기반의 인터넷 응용에서는 전송 프로토콜로 주로 TCP나 UDP를 사용하고 있다. TCP는 잘 알려져 있는 바와 같이 안정적인 데이터의 전송을 보장해준다. 예를 들어, TCP를 전송 프로토콜로 사용한 데이터 전

송의 경우에는 데이터의 안정적인 전송의 보장을 위해 수신 측에서는 수신된 데이터에 대한 ACK를 송신 측에 보내주며, 송신 측에서는 이를 토대로 손실되었다고 추측되는 패킷을 수신 측에게 재전송 해준다. 하지만, 이와 같이 안정적인 데이터 전송의 보장을 위한 TCP의 일련의 작업들이 큰 전송 부하로 작용하는 반면, 일반적인 비디오 전송 시스템에서는 모든 데이터의 안정적인 전송보다는 부하가 작고 빠른 전송이 더 중요한 것이 보통이다. 이와 같은 점에서, UDP는 TCP보다 이와 같은 용도에 더 적합하다고 할 수 있다. UDP는 TCP의 경우와 달리 모든 데이터의 안정적인 데이터의 전송을 보장하기 위한 작업을 수행하지 않는다. 즉, 손실된 패킷의 재전송이 이루어지지 않는다.

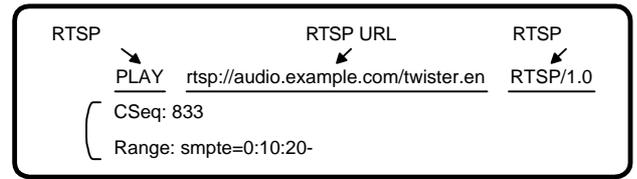
하지만, UDP도 패킷들 간의 순서나 시간 관계를 기술 할 수 없다는 부족함을 가지고 있다. 본 시스템에서는 이 UDP 위에 RTP(RFC 1889)를 구현하여 전송 프로토콜로 사용하고 있다. RTP의 사용에 의해, 패킷들 간의 순서(sequence number)와 패킷들 간의 시간 관계(timestamp)를 기술하는 것이 가능해 진다. 이와 같은 RTP/UDP/IP 계층의 사용을 통해 부하가 작은 MPEG 스트림의 전송을 이룰 수 있었다.

### 2.2 RTSP를 이용한 스트림 전송의 제어

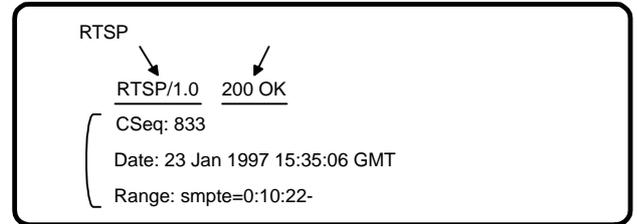
RTSP(RFC 2326)는 스트리밍 응용을 위해 제안된 프로토콜로, 본 시스템에서는 RTP 세션의 초기화와 재생, 일시 정지, 위치 이동, 재생 속도의 조절과 같은 사용자의 요청을 서버 측에 전달하고 수행 여부를 확인하기 위한 용도로 사용되고 있다.

RTSP는 텍스트 기반의 프로토콜로서 ISO 10646 문자셋을 사용한다. RTSP의 구문과 동작 방식은 의도적으로 HTTP/1.1과 유사한 구조를 가지고 있다. 그림 2는 RTSP 메시지의 한 예인데, 클라이언트에서 서버에게 특정 구간의 재생을 요청하는 경우의 메시지 예를 보여주고 있다.

그림 2에서 볼 수 있듯이 RTSP 메시지는 요청 메시지와 그에 대한 응답 메시지로 구성되어 있다. 이와 같은 RTSP를 이용하면 재생, 일시 정지, 위치 이동, 재생 속도의 조절과 같은 사용자의 요청을 서버와 협상하는 것이 가능해진다. RTSP 1.0에서 정의되어 있는 RTSP 메소드는 표 2와 같다.



(a)



(b)

그림 3. RTSP 메시지의 한 예

메소드	설명
DESCRIBE	서비스에 관한 설명을 요청
ANNOUNCE	서비스에 관한 설명을 수정
GET_PARAMETER	특정 파라미터의 값을 요청
OPTIONS	상대방이 지원하는 메소드들의 리스트를 요청
PAUSE	전송의 일시 정지를 요청
PLAY	전송 시작의 요청
RECORD	전송되는 스트림의 저장을 요청
REDIRECT	다른 주소의 지정
SETUP	전송 방식, 포트등의 정보 전달
SET_PARAMETER	특정 파라미터의 값을 수정
TEARDOWN	세션 종료 요청

표 2. RTSP 메소드의 개요

### 2.3 프로토콜 스택

본 시스템의 서버와 클라이언트간에 송수신 되는 패킷은 RTP 패킷과 RTSP 패킷으로 나누어질 수 있다. 이 중 RTP 패킷은 MPEG 스트림 전송을 위한 패킷이며, RTSP 패킷은 RTP 스트림의 제어를 위한 메시지 패킷이다. RTP 패킷은 높은 대역폭으로 인해 낮은 부하의 전송이 요구되며, RTSP 패킷은 메시지의 양은 적으나 안정적인 전송이 필요하다는 특성을 가지고 있다.

이상의 점들을 고려해서, 본 시스템에서는 RTP는 부하가 작은 UDP를 하부 계층 프로토콜로 사용하고 있으며, RTSP는 안정적인 전송을 보장해주는 TCP를 하부 계층 프로토콜로 사용하고 있다. 이상을 도시하면

프로토콜 스택은 그림 3과 같다.

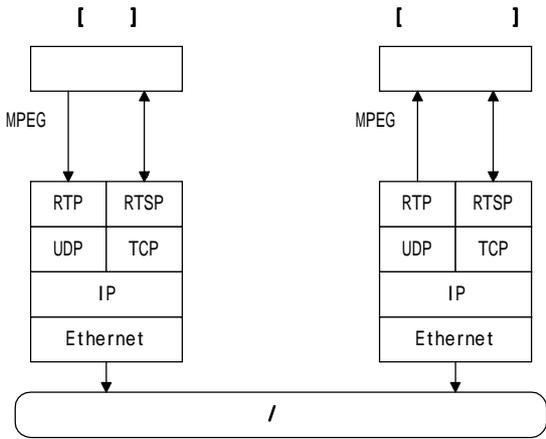


그림 4. 프로토콜 스택

### 2.4 MPEG 스트림 송신 모듈

그림 4는 VCR 서버의 송신 모듈의 구조를 보여주고 있다.

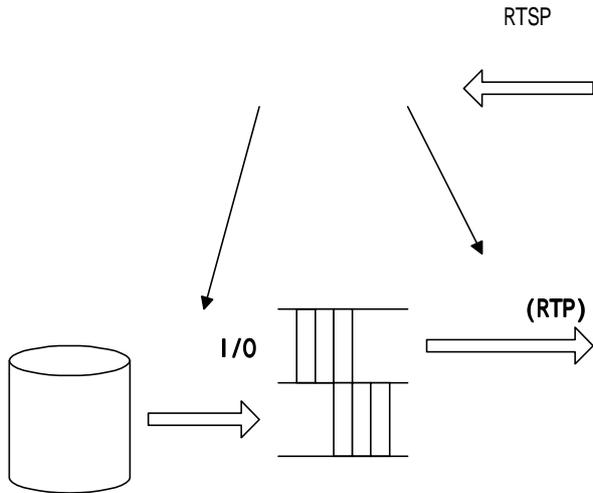


그림 5. 서버의 송신 모듈

VCR 서버는 RTP를 이용한 네트워크 송신 모듈과 MPEG 파일 I/O 모듈, 그리고 RTSP를 이용한 스트림 제어 모듈로 구성되어 있는데, 이 모듈들을 모두 별도의 쓰레드로 구현하였다. 이것은 MPEG 파일 관련 처리가 이루어지는 동안 네트워크 송신이 중단되는 일을 막기 위한 것이다. 버퍼는 네트워크 송신용과 파일 I/O 용의 두 개로 구성되어 있는데, 네트워크 송신 모듈이 한 버퍼에 저장되어 있는 데이터를 송신하는 동안 병렬적으로 파일 I/O 쓰레드는 MPEG 파일 관련 처리 수행

후 다른 버퍼에 채워 넣는 방식으로 동작한다. 네트워크 송신 모듈이 한 버퍼에 저장되어 있는 데이터를 모두 송신하고 나면, 파일 I/O 쓰레드가 MPEG 스트림으로 채워넣은 버퍼로 이동하여 다시 송신을 시작하고, 파일 I/O 쓰레드는 이와 동시에 네트워크 송신 쓰레드가 비워 놓은 버퍼에 다시 데이터를 채워 넣는 방식으로 동작한다.

### 2.5 고속 재생의 구현

본 시스템에서는 고속 재생을 지원하는데, 고속 재생의 경우에는 정상 속도 재생의 경우보다 높은 대역폭이 필요하게 된다. 예를 들어 2배속 재생의 경우에는 1배속 재생의 경우보다 2배의 대역폭이 필요하게 된다. 하지만, 본 시스템과 같이 네트워크 기반의 VCR의 경우에는 고속 재생 시에도 기존의 대역폭을 그대로 유지하는 것이 좋다. 이를 위해 본 시스템에서는 MPEG 영상 스트림 중에서 가장 중요한 정보인 인트라 프레임만을 추출하여 전송하는 방식을 적용시키고 있다.

그림 5는 이와 같은 인트라 프레임 추출 과정을 보여주고 있다.

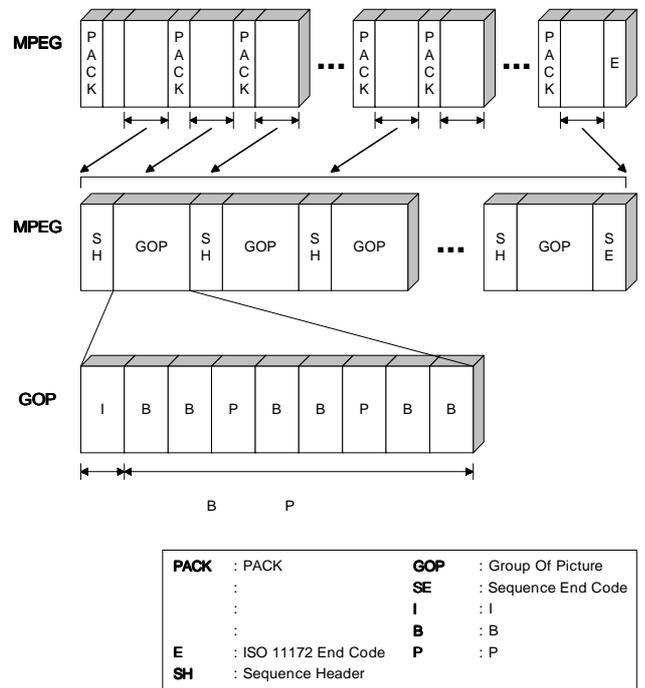


그림 6. MPEG 영상 I 프레임의 추출

본 시스템에서 사용하는 스트림이 음성과 영상이 모두 포함되어 있는 MPEG-1 시스템 스트림이기 때문에, 일차적으로 MPEG 음성 스트림과 영상 스트림의 구분

작업이 이루어진다. 일단 MPEG 영상 스트림만의 추출이 이루어지면, 다음으로 MPEG 영상 프레임들 중 I 프레임만을 통과시키고, B 프레임과 P 프레임을 제거하는 작업이 이루어진다. 본 시스템에서는 이상의 과정을 통해 고속 재생시의 대역폭을 정상 속도 재생시의 대역폭 수준으로 낮추고 있다.

본 시스템의 특징적인 점은 이와 같은 인트라 프레임 추출 과정이 원본 MPEG-1 시스템 스트림에 실시간으로 처리된다는 점에 있다. 따라서, 고속 재생을 위한 별도의 파일의 준비와 같은 사전 작업이 필요하지 않다는 장점을 가지고 있다.

### 3. 구현 결과

비디오 서버는 그림 6에서 볼 수 있듯이 홈페이지를 통해 비디오 서비스의 목록을 보여준다.

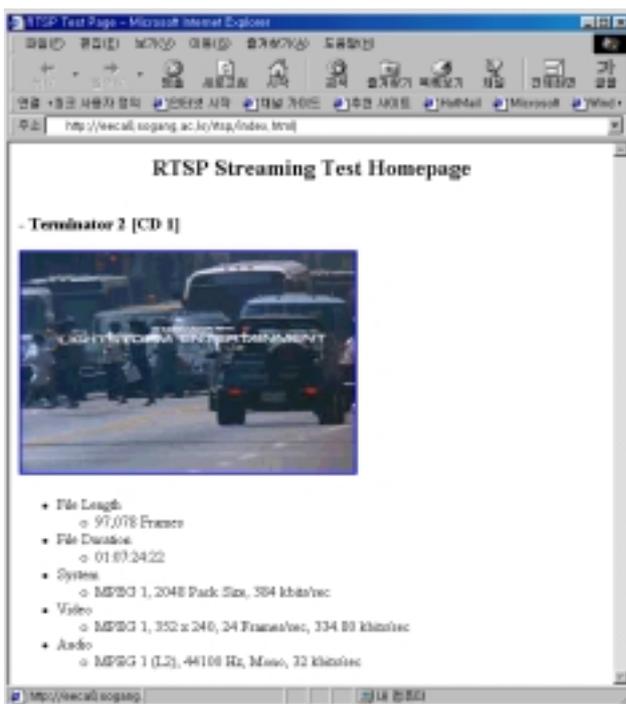


그림 7. 실험을 위해 사용한 웹 홈페이지

마찬가지로 각 비디오 서비스에 관한 설명 등도 간단한 홈페이지의 제작 과정을 통해 클라이언트에게 제공되게 된다.

비디오 서버는 서비스 중인 클라이언트가 없는 경우에는 대기 상태로 있다가, 클라이언트가 웹 페이지내의 특정 비디오 서비스를 클릭 하면 비디오 서버와 클라이

언트간의 연결이 이루어지게 된다. 이처럼 웹 서버의 역할은 VCR 클라이언트와 서버를 연결시켜주는 것이며, 이후의 통신은 이 둘 사이에서 이루어진다.

그림 7은 사용자가 웹 브라우저 상에서 특정 비디오 서비스를 선택한 경우 실행되는 VCR 클라이언트의 실행 모습을 보여주고 있다. VCR 클라이언트는 서버로부터 재생에 필요한 최소한의 MPEG 스트림을 수신하면 바로 재생을 시작한다.



그림 8. 실행된 VCR 클라이언트

전체적인 구성은, 비디오를 재생해서 보여주는 비디오 윈도우와, 재생, 일시 정지, 고속 재생 등을 요청할 수 있는 스트림 제어 톨바, 임의 위치 재생을 요청할 수 있는 스크롤 바, 그리고 현재의 시간 위치를 보여주는 타임 디스플레이로 구성되어 있다.

구현된 시스템은 서버와 클라이언트 모두 PC의 Microsoft Windows 95/98/NT 상에서 C++ 언어로 제작되었다. 서버는 Pentium-II 400MHz PC에서 테스트하였으며, 클라이언트는 일반 Pentium 급 PC 상에서 테스트하였다. 서버의 MPEG 스트림 처리 및 클라이언트의 MPEG 복호화, 그리고 재생 모두 소프트웨어적으로 처리하였다. 모든 PC는 100Mbps 이더넷 카드를 장착하고 있으며, 100Mbps Hub를 통해 연결되었다. 스트림 제어 톨바에 있는 여러 기능들의 테스트를 통해, 일시 정지, 재생 시작, 고속 재생 등의 기능들이 정상적으로 수행됨을 볼 수 있었으며, 스크롤 바를 이용한 비디

오의 임의 위치 재생도 정상적으로 수행됨을 볼 수 있었다.

#### 4. 결론 및 추후과제

본 논문에서는 스트리밍 방식에 기반한 실시간 인터넷 VCR 서버와 클라이언트를 설계하고 구현하였다. 구현된 시스템은 널리 보편화되어 있는 멀티미디어 형식 중의 하나인 MPEG-1 시스템 스트림을 압축 형식으로 채택하고 있으며, 이의 전송을 위해 실시간 전송에 적합한 프로토콜인 RTP를 UDP/IP 상에 구현하여 사용하고 있다. 또한 이 RTP 세션의 초기화 및 RTP 스트림의 제어를 위해 RTSP 실시간 스트리밍 프로토콜을 구현하여 사용하고 있다.

본 논문에서 구현된 시스템의 가장 큰 특징은 비디오의 압축 형식에서부터 전송, 그리고 전송의 제어에 이르기까지 모든 모듈들이 표준에 기반하고 있다는 점에 있다. 현재의 소프트웨어 개발 추세가 컴포넌트 기반으로 나아가고 있다는 점을 고려해보면, 표준에 기반한 시스템의 구축은 더 큰 의미를 가질 것이다.

본 논문에서는 MPEG-1 시스템 스트림만을 대상으로 하고 있지만, 여러 가지 스트림이 지원된다면 다양한 플랫폼에 적용 가능한 시스템이 될 것이다. 예를 들어 음성 스트림의 지원이나 낮은 대역폭의 비디오 스트림의 지원이 이루어진다면, 대역폭이 낮은 네트워크 환경에도 적용 가능한 시스템이 될 것이다. 또한, 네트워크 상태나 수신자 상태에 적응적인 전송 제어가 이루어진다면 다양한 플랫폼에 적용 가능해질 것이다. 본 연구는 이를 위한 기초를 제공하였다는 점에 그 의의가 있다고 하겠다.

#### 참고문헌

[1] Schulzrinne, H., "Real Time Streaming Protocol (RTSP)", RFC 2326, April 1998.  
[2] Schulzrinne, H., "RTP profile for audio and video conferences with minimal control", RFC 1890, January 1996.  
[3] Schulzrinne, Casner, Frederick, Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, February 1996.

[4] Hoffman, D., Fernando G., "RTP Payload Format for MPEG1/MPEG2 Video", RFC 2038, January 1998.  
[5] Stephen A. Thomas, "IPng and the TCP/IP Protocols - Implementing the Next Generation Internet", John Wiley & Sons, 1996.  
[6] Joan L. Mitchell, William B. Pennebaker, Chard E. Fogg, Didier J. LeGall, "MPEG Video Compression Standard", International Thomson Publishing, 1996.  
[7] Fielding, R., Gettys, J., Mogul, J., Nielsen, H., and T. Berners-Lee, "Hypertext transfer protocol - HTTP/1.1", RFC 2068, January 1997.  
[8] W. Richard Stevens, UNIX Networking Programming Volume 1, Prentice Hall, 1998.