ISSN 1229-800X The Transactions of the Korean Institute of Electrical Engineers Vol. 65P, No. 4, pp. 326~334, 2016
http://dx.doi.org/10.5370/KIEEP.2016.65.4.326

다중 플랫폼 지원 실시간 HD급 영상 전송기 개발에 관한 연구

The Study on the Development of the Realtime HD(High Definition) Level Video Streaming Transmitter Supporting the Multi-platform

이 재 희* · 서 창 진†
(JaeHee Lee · ChangJin Seo)

Abstract - In this paper for developing and implementing the realtime HD level video streaming transmitter which is operated on the multi-platform in all network and client environment compared to the exist video live streaming transmitter. We design the realtime HD level video streaming transmitter supporting the multi-platform using the TMS320DM386 video processor of T.I company and then porting the Linux kernel 2.6.29 and implementing the RTSP(Real Time Streaming Protocol)/RTP(Real Time Transport Protocol), HLS(Http Live Streaming), RTMP(Real Time Messaging Protocol) that can support the multi-platform of video stream protocol of the received equipments (smart phone, tablet PC, notebook etc.). For proving the performance of developed video streaming transmitter, we make the testing environment for testing the performance of streaming transmitter using the notebook, iPad, android Phone, and then analysis the received video in the client displayer. In this paper, we suggest the developed the Realtime HD(High Definition) level Video Streaming transmitter performance data values higher than the exist products.

Key Words : RTSP/RTP, HLS, RTMP, SDP, DDNS, STUN

1. 서 론

최근에 인터넷 이용의 급증으로 네트워크 속도가 증가하고 주변 정보통신 기기들의 급속한 발전으로 인해 인터넷 사용자들이 대용량 멀티미디어 콘텐츠를 통한 실시간 정보 전달 및 표현에 대한 수요가 증가하고 있다. 이러한 현상으로 방송과 통신 융합 형태의 기술이 속속 등장하고 실시간으로 멀티미디어 콘텐츠를 송, 수신할 수 있는 인터넷 방송이 대중화되고 있다. 기존의 웹 기반의 비디오나 오디오와 같은 멀티미디어 전송 스트리밍 기술은 초기 접속 시 버퍼링을 위한 2 ~ 5초 서비스 지연시간이 발생하는 게 사실이다. 그러나 비디오 및 오디오는 어느 정도의 손실이 있어도 QoS를 만족하며 실시간 특성을 가지므로 초기 시간지연을 방지하기 위해서는 UDP/IP기반의 전송 프로토콜을 사용하고 있는 추세이다.

데이터 전송의 신뢰성을 바탕으로 한 TCP/IP 기반의 HTTP 응용프로토콜은 인터넷에서 실시간 특성의 미디어 데이터를 전송하는데 동시 다중 접속자 수용 및 다중포트 구성의 문제점이 있어 적당하지 못하다. 본 논문에서는 방송과 통신의 융합시대를 향한 보다 나은 고품질 영상 전송 서비스를 제공할 수 있도록 실시간 동영상 전송 프로토콜에 대하여 연구, 분석하고 이를 바탕으로 다양한 종류의 클라이언트 단말기에서 실시간 영상수신이 가능하도록 휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기를 설계 및 구현하고자 한다.

† Corresponding Author : Dept. of National Defense Intelligence Engineering, Sangmyung University, Korea
E-mail : cjseo@smu.ac.kr
* Dept. of Information & Telecommunication, Dongseoul University, Korea
접수일자 : 2016년 11월 4일
최종완료 : 2016년 11월 28일

2. 실시간 동영상 전송 프로토콜

2.1 스트리밍

스트리밍(Streaming)은 리얼네트웍스(RealNetworks)사가 개발한 스트리밍 멀티미디어라는 기술에서 유래하는데, 이 기술의 원리는 대용량의 멀티미디어 자료라도 이를 개별적으로 실행할 수 있는 1~2초 분량의 작은 조각으로 나눠 시냇물이 흐르듯이 데이터를 연속적으로 전송하여, 수신하는 측은 전체자료가 모두 수신될 때까지 기다릴 필요 없이 즉석에서 각 조각의 파일들을 재생하는 기술을 의미한다. 응용계층의 연속적인 미디어 데이터를 짧은 토막으로 잘라 패킷화 하여 전송하고 수신측에서는 일정한 단위의 데이터가 수신될 때마다 실시간 특성을 어느 정도 유지하면서 연속적인 복호화를 통해 영상을 재생하는 것으로 전체 데이터를 수신한 다음 복호 재생하는 방식과는 다른 것이다.

현재 스트리밍이라는 용어는 인터넷상에서의 방송을 의미하며 스트리밍은 브로드캐스팅의 한 종류라고 할 수 있다. 한편 스트리밍은 방식으로는 주문형(On Demand)과 라이브(Live)로 나뉘고, 기술적으로는 유니캐스터(Unicast) 와 멀티캐스터(Multicast)로 구분된다.

스트리밍 기술이 사용하는 표준 프로토콜은 RTSP(Real Time Streaming Protocol)가 있다[1]. 첫 번째 방식인 주문형 스트리밍방식은 미리 준비되어 있는 미디어 파일을 스트

리밍해 보는 방식으로 미디어 영상을 보기위해서는 해당 파일을 클릭한 후 그 파일의 구동 신호를 수신할 때까지 기다려야 한다. 두 번째 방식은 라이브 스트리밍이다. 이 방식은 해당하는 곳을 클릭하면 곧바로 스트리밍을 지원하는 플레이어가 나타나면 영상 또는 음향을 보내주게 되는 방식이다. 이 방식에서는 지금까지는 네트워크의 전송속도가 음질이나 화질에 큰 영향을 주었다. 데이터가 필요이상으로 많이 들어오게 되면 일시적으로 버퍼에 저장하고 데이터가 필요한 것 보다 적게 들어오게 되면 음질이나 화질이 떨어지는 현상이 발생했다.

2.2 RTSP(Real Time Streaming Protocol)

RTSP는 인터넷상의 스트리밍서비스에 있어서 서비스에 대한 요구, 응답이나 서비스 연결설정, 스트림재생과 관련된 각종 제어를 담당하는 프로토콜로서 RealNetworks, Netscape, Communications Corporation, Columbia 대학에서 공동으로 개발하여 1998년 4월에 RFC2326으로 표준화 되었다[2].

스트리밍을 위한 프로토콜은 RTSP와 RTP를 사용하게 되고 이를 통해 패킷화하여 스트리밍 서비스를 수행한다. RTSP프로토콜은 유니캐스트 또는 멀티캐스트를 모두 사용하는 멀티포인터에서 멀티미디어 스트리밍을 위한 강력한 프로토콜을 제공하기 위한 목적을 가진 응용계층의 프로토콜로서, 오디오, 비디오와 같은 시간적으로 동기화된 스트림을 생성하고 제어한다. 연속매체 자체는 전송하지 않고 스트리밍 서버를 위한 네트워크 원격제어 역할을 수행한다. RTSP프로토콜의 기본 동작원리를 보면 클라이언트는 서버에게 실시간 특성을 갖는 영상이나 음성 정보를 요청하고, 이 요청에 의해 서버가 정보를 전송하는 방식으로 동작한다. 여기서 스트리밍이란 서버 측에서 압축된 연속적인 메시지를 패킷으로 잘라 전송하면, 수신측에서는 매시지 전체를 수신한 다음 복호화/ 재생하는 것이 아니라, 어떤 일정한 단위의 메시지가 수신될 때마다 복호화를 수행하여 실시간 특성을 어느 정도 유지하면서 연속적인 재생을 가능하게 해주는 기술을 말한다.

RTSP프로토콜의 특징은 유니캐스트(Unicast) 또는 멀티캐스트(Multicast)환경에서 복수개의 미디어정보 스트림을 동시에 제어가능하고 일반적으로 TCP와 UDP를 포함하는 다양한 전송계층 프로토콜 위에서 동작할 수 있으며 RTP와 RTCP프로토콜과 함께 사용한다.

RTSP는 제어 메시지 전송을 위해서 신뢰성 있는 TCP를 사용하여 RTP/RTCP 채널 설정을 한 다음, RTP/RTCP 패킷이 전달되도록 한다. 즉 세션의 설정과 해제는 RTSP에 의해 제어되고, 실제의 A/V정보는 RTP를 통해 전송하고, HTTP와 비슷한 Syntax와 오퍼레이션을 가지나 Server와 Client 모두가 Request를 보내고 Response를 받을 수 있으며 초기, 준비, 재생 상태를 가진다. 그리고 세션의 설정과 해제에 사용되는 명령어에 대한 프로토콜 규정은 SDP(Session Description Protocol)를 통하여 이루어진다. 그림 1과 같이 RTSP 동작과정은 Client의 Player는 접속요청을 SDP를 사용한 RTSP를 통해서 이루어지고 이에 대한 스트리밍 서버의 응답 또한 SDP는 session과 session의 설정하는 정보들을 담고, SDP에 의해 담겨진 정보들은 RTSP에 적절히 할당되기에 적합하도록 사용된다.

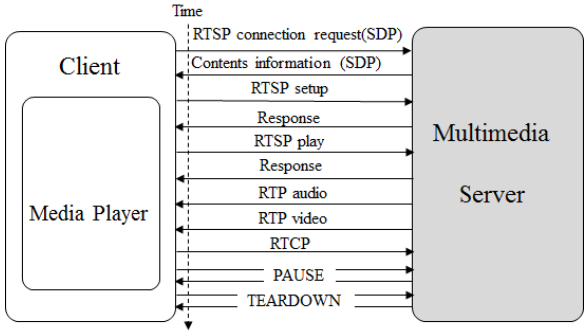

그림 1 RTSP 동작과정
Fig. 1 RTSP operation processing

Media Player의 제어 명령은 RTSP를 통해 전달되고 응답 또한 RTSP를 통해 이루어진다. 이 정보를 이용하여 스트리밍 서버에게 Setup 요구를 보낸다. 서버가 Setup을 허용하면 클라이언트는 Play요구를 보낼 수 있다. Play 요청이 받아들여지면 실제 데이터는 RTP/RTCP 프로토콜을 사용하여 클라이언트에게 전송되어진다. 일시적인 재생중지를 위해서는 PAUSE 메서드를 사용하고 세션을 완전히 닫기 위해서는 TEARDOWN 메시지를 보낸다.

2.3 RTP(Realtime Transport Protocol)

RTP(Realtime Transport Protocol)는 1995년 11월 IESG(Internet Engineering Streering Group)으로부터 인터넷 제안표준으로 승인되었으며 RFC1889(Request For Comments 1889)와 RFC 1890(RTCP : RTP Profile for Audio and Video Conferences with Minimal Control)으로 발표 되었다[3].

RTP는 멀티캐스트 또는 유니캐스트 상에서 음성, 화상, 또는 모의 데이터와 같은 실시간 데이터를 전송하는 응용에 적합한 기능을 제공한다. 그러나 RTP는 자원 예약에 대한 내용은 다루지는 않으며, 특히 적시 데이터 전송 (Timely Delivery), QoS 보장, 뒤바뀐 순서의 전송 방지와 같은 기능을 제공하지 않는다. 따라서 트랜스포트의 의미는 실시간 데이터의 특성에 중점을 두어 제정한 표준이라고 할 수 있다. RTP패킷은 UDP를 이용하여 전달된다. RTP에서의 다중화는 목적지 전송 주소가 제공하며, 여러 미디어를 사용하는 회의에서는 각각의 미디어들이 서로 다른 목적지 주소를 가지는 RTP세션들 내에서 전송된다.

서버에 저장된 오디오 및 비디오 스트림을 RTP 패킷으로 만든 후에 UDP와 IP를 통하여 인터넷으로 전송되는 RTP 프로토클 Stack은 그림 2와 같다. 오디오는 RTP패킷 구성할 시에는 특별한 Parsing 과정이 필요 없고 단순히 일정한 길이로 자르기만 하면 된다. 오디오는 사용자에게 일방적으로 전송만 하면 되므로 접속 요정 및 제어 명령에 대한 응답과는 달리 바인딩 과정이 필요 없다. 하지만 비디오 스트림은 프레임당 나오는 비트의 수가 일정하지 않으므로 오디오와 같이 일정한 길이의 RTP 패킷을 만들 수는 없다. 그러므로 Parsing과정이 필요하다.

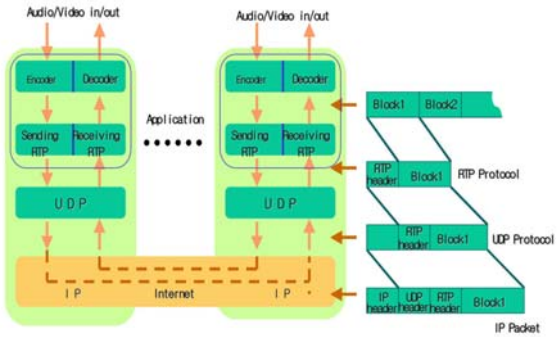

그림 2 RTP 프로토콜 Stack
Fig. 2 RTP protocol stack

2.4 SDP(Session Description Protocol)

SDP 프로토콜은 인터넷에서 멀티미디어 세션에 참여하기 위한 사용자가 필요로 하는 정보를 광고하고 실시간으로 멀티미디어 세션을 정의할 목적으로 IETF의 MMUSIC (Multiparty Multi-media Session Control)워킹그룹에 의해 RFC2327로 표준화된 프로토콜이다[4]. SDP는 멀티미디어 세션을 정의하기 위해 세션의 생성, 세션의 초대, 세션기술에 관한 정보를 담고 있으며 이러한 정보로는 미디어 제어서버 에 대한 주소와 포트, 미디어 유형 및 미디어서버 주소 등을 포함하게 된다. 세션을 기술하기 위해서는 SDP 프로토콜을 이용한다. SDP 프로토콜은 멀티미디어 세션들을 기술하고 다양한 형식의 세션을 초기화 하는 데 사용되고 있다. IP 멀티캐스트 기능을 이용하여 인터넷 위에 구축된 MBone은 다자간 멀티미디어 응용 서비스에 널리 이용되고 있다. 하지만 사용자가 현재 MBone에 개설되어 있는 세션에 참가하려면 세션의 내용이나 각 세션이 사용하는 멀티캐스트 주소와 포트번호를 알아야 한다.

SDP 프로토콜은 멀티미디어 세션을 기술하는 데 사용하는 프로토콜로서 이미 정해진 멀티캐스트 주소와 포트로 패킷을 주기적으로 보내어 세션을 알릴 때 페이로드 부분을 기술하는 목적으로 주로 사용된다. SDP에서는 이 멀티미디어 세션을 일정 기간 동안 존재하는 미디어 스트림의 집합으로 정의하고 있다. 세션의 존재를 광고하고 그 세션에 참가할 수 있게 충분한 정보를 전달하는 기능이 있다. 인터넷 환경에서 SDP는 중요한 두 가지 기능을 제공하고 있다. 하나는 세션의 존재를 알리는 것이고, 다른 하나는 그 세션에 참가할 수 있게 충분한 정보를 전달하는 기능이다. 이를 위해 SDP는 다음 정보를 포함하고 있다. 세션 이름과 목적, 세션이 개설되는 시간, 세션을 구성하는 미디어, 미디어 수신정보(주소, 포트, 포맷), 미디어 유형(비디오, 오디오), 전송 프로토콜, 미디어 포맷, 미디어에 대한 멀티캐스트주소, 미디어에 대한 전송포트, 대역폭 정보를 제공한다. 또한 클라이언트가 세션에 참가하는데 필요한 충분한 대역폭 정보와 세션을 책임지고 있는 사람의 연락 정보를 제공한다.

2.5 Hinted for On-instant & Skip-Protection 기법

Hinted for On-instant & Skip-Protection 기법이란 RTSP/RTP를 사용하여 한 번에 한 프레임의 데이터를 모두 보낼 수 있는 기반을 마련하였으나 Server가 즉각적으로 응답하기 위하여 동영상 파일을 페이로딩 하지 않고 헤더의 작은 데이터 (원본 미디어 데이터의 6%)만을 로딩 하여 전송할 수 있도록 하는 기법이다. 여기서 Hinted기법은 Client가 Server에 동영상 요청시 Streaming Server는 동영상 파일을 모두 페이로딩 하지 않고 데이터를 Client에 전송방식으로 동영상데이터의 정보(미디어 속성)를 파일의 Head부분에 추가 하여 Streaming Server가 헤더 데이터만을 페이로딩하여 Client에 바로 전송하는 기법으로 영상과 음성 압축 단계부터 적용한다. 이 기법은 RTSP 프로토콜을 사용하는 VOD Server에 적용된 기법이다. On-Instant 기법은 Client가 Server에 동영상 요청시 즉각적인 반응을 보이기 위해서 모노리틱 커널(Monolithic Kernel) 기반의 리눅스 운영체제를 선택하여 RTP port(6,950-6,990) 50개를 사용할 수 있는 특징을 활용하여 미리 대기 프로세스를 띄워놓아 Client 요청시 Hinted미디어 데이터를 바로 전송하는 기법으로 RTSP 프로토콜을 사용하는 VOD Server에 적용된 기법이다. Skip-Protection 기법은 RTP(6,950~ 6,990)를 통해 동영상을 전송하면서 RTSP(554, 7070)사용하여 Server와 Client가 지속적인 정보를 주고받을 수 있게 구현이 됨으로 인하여 Client의 모든 상황을 Server가 즉각적으로 인지할 수 있음으로 인터넷 속도가 저속으로 떨어졌을 때 Server에서 미디어 데이터의 패킷을 감소시킴으로 끊어짐 없는 동영상을 전송하는 기법으로 RTP와 RTSP 프로토콜을 사용하는 실시간 영상 전송시스템 즉 VOD, Live 생방송에 적용된 기법이다.

2.5.1 Hinted 구성 Role

RTP 페이로드 구성요소는 RTP payload meta-information과 RTP data로 구성된다. 즉 RTP payload meta-information format은 그림 4과 같다. RTP 데이터 형식과 하위필드 값은 표 1과 같고 이것으로 정의되는 RTP

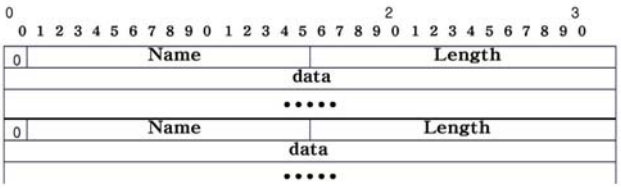

그림 3 RTP payload meta-information format
Fig. 3 RTP payload meta-information format

표 1 Defined Name subfield values
Table 1 Defined Name subfield values

RTP data type	Name subfield value
Transmit time	Tt
Frame type	Ft
Packet number	Pn
Packet position	Pp
Media data	Md
Sequence number	Sn

data 표준 format은 그림 4와 같다[5].

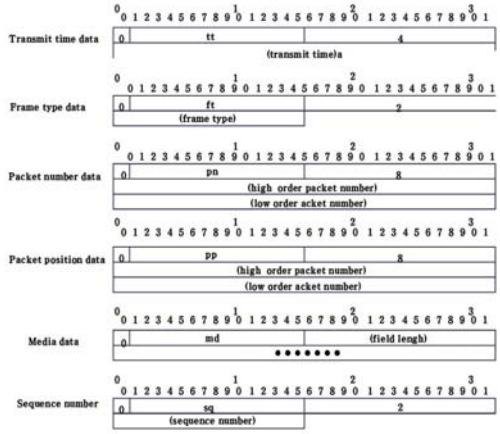

그림 4 RTP data in standard format
Fig. 4 RTP data in standard format

2.6 HLS(Http Live Streaming)서비스

라이브 스트리밍을 위한 전통적인 방식은 RTSP(Real-Time Streaming Protocol)/RTP(Real-Time Transport Protocol), RTMP(Real-Time Messaging Protocol) 등이 있다. 이 방식을 사용하는 스트리밍 서버는 영상데이터의 전송뿐만 아니라, 동영상에 대한 정보 분석이나 전송규격에 맞도록 동영상 파일을 읽어서 변형하는 기능도 갖춰야 한다. 기능이 많은 만큼 웹 서버에 비해 도입 비용이 상대적으로 높을 뿐만 아니라 RTSP/RTP의 경우, RTSP와 RTP가 서로 다른 네트워크 연결을 통해 데이터를 교환하기 때문에 방화벽(Firewall)이나 NAT(Network Address Translator)를 많이 쓰고 있는 환경에서는 서비스가 원활하게 동작하지 않은 문제점이 있다. 그래서 대안으로 나온 것이 HTTP 프로토콜을 전송 채널로 사용하는 것이다. HTTP 방식은 양방(Full-duplex)방식이 아니기 때문에 라이브 스트리밍을 위해서는 단점을 극복할 별도의 방식이 필요로 하지만, 방화벽에서 HTTP서버로의 요청만 통과시키면 되기 때문에 방화벽의 설정이 단순해진다. 요청과 응답이 1:1로 대응되므로 NAT환경에서도 서버와 통신하는 것이 쉽다. 방화벽 문제뿐만 아니라 웹 서비스를 위한 캐시(Cache)구조를 그대로 사용할 수 있고, 기존에 구축되어 있는 CDN(Content Delivery Network)도 특별히 변경하지 않고 그대로 이용할 수 있다는 것이 HLS(HTTP Live Streaming)의 장점이다. HLS는 Apple사에서 iOS 3.0과 Quicktime-X 를 위해 2009년에 내놓은 프로토콜이다[6, 7]. 이 프로토콜에서는 스트리밍 데이터를 MPEG-2 Transport Stream에 담아 시간 단위로 잘게 쪼개서 전송하는 방식이다. 그리고 어떤 파일을 재생해야 하는 지에 대한 정보는 m3u8파일을 이용하여 사용자의 플레이어에 전달한다. HLS는 iPhone 사용자의 증가와 더불어 자연스럽게 그 수요가 증가하고 있고 또한 규격 자체의 단순함과 IETF(Internet Engineering Task Force)를 통한 표준화 작업 등을 통해 다른 업체들도 쉽게 HLS를 지원할 수 있게 되었다[8]. HLS방식과 기존의 스트리밍서버를 사용하는 방식 모두 동영상을 중단 없이 사용자의 동영상 플레이어에 전달한다는 점에서는 동일한 구조이다.

HLS와 Streaming Server방식의 차이는 크게 두 가지이다. 첫 번째는 동영상 정보를 전달하는 방식이고 두 번째는 HLS에서 만든 스트림 세그먼트(Stream segment)이다. HLS에서 서버는 HTTP로 요청을 받아서 사용자의 플레이어에 응답을 주는 역할만 한다. 요청받은 파일을 읽어서 어떠한 변형도 하지 않고, 읽은 그대로 응답에 포함해 보내기만 한다. 저장되어 있는 파일을 읽어서 HTTP 응답에 데이터를 실어서 보낼 수 있는 웹 서버이면 어떤 웹 서버든 사용할 수 있다. 그러나 스트리밍 서버방식은 사용자의 플레이어의 스트리밍 규격에 맞게 다시 서버가 데이터를 변형해 보내는 작업을 수행한다. 경우에 따라서는 특정 스트리밍방식을 위해 해당회사의 제품을 구매해야 할 때도 있다.

웹 서버에 비하여 처리과정이 더 많기 때문에 서버 한 대당 처리할 수 있는 클라이언트의 수도 웹 서버보다 적다. 스트림 세그먼트는 일정한 시간 간격마다 입력받은 미디어 데이터를 분할해 파일을 만들고, 그 분할한 파일에 접근할 수 있는 메타데이터인 m3u8파일을 생성하는 일을 한다[8]. HTTP는 양방향 방식이 아니기 때문에, 클라이언트에서 반드시 서버에 요청을 해야 그에 맞는 응답을 받을 수 있다. 즉, 잘게 쪼갠 동영상과 다음에 볼 동영상정보를 함께 클라이언트에 전달하고 동영상이 끊어짐 없이 때에 맞춰 다음 동영상 정보를 요청한다.

기존의 스트리밍 서버에 의한 서비스 구조는 "원본 → 인코더 → 스트리밍서버→ 플레이어" 라는 데이터 흐름에서 HLS방식은 "원본 → 인코더 → 스트림 세그먼트 → 웹서버 →플레이어" 라는 흐름으로 변화된다. 대규모 서비스를 위한 부하 분산을 고려할 때 HLS방식은 기존의 스트리밍 서버 방식보다 간단하고 효율적이라 할 수 있다.

2.7 RTMP(Real Time Messaging Protocol)

리얼 타임 메시징 프로토콜(Real Time Messaging Protocol, 흔히 줄여서 RTMP)은 어도비 시스템즈사의 독점 컴퓨터 통신 규약이다. RTMP는 오디오, 비디오 및 기타 데이터를 인터넷을 통해 스트리밍할 때 쓰인다. RTMP는 어도비 플래시 플레이어와 서버 사이의 통신에 이용된다[11]. RTMP 종류는 표 2와 같다.

2.7.1 RTMP 동작

TCP 기반의 RTMP 프로토콜은 접속을 지속적으로 유지하는 데 기여한다. 또, 실시간 통신을 한다. 더 큰 덩어리의 정보를 보낼 수 있는 능력을 유지하는 동안, 부가적으로 비디오 및 오디오 스트림을 부드럽게 전달하기 위해, 이 프로토콜은 비디오 및 데이터를 여러 조각들(fragments)로 나누기도 한다. 이 조각들의 크기는 클라이언트와 서버 간에 유동적으로 결정된다. 동적 크기 조절은 비활성화 될 수 있다. 비디오 및 기타 데이터에 대한 스트림 조각들의 기본 크기는 128 바이트이다. 오디오에 대한 스트림 조각들의 기본 크기는 64 바이트이다. 여러 개의 스트림이 있을 때, 각각의 스트림으로부터 꺼내온 조각들은 인터리빙(interleaving)되며, 한 접속 내에서 다중화 된다. 데이터 덩어리(chunk) 크

표 2 RTMP 종류
Table 2 RTMP type

RTMP 종류	내 용
RTMP (기본)	1935번 포트 사용, 암호화되지 않은 RTMP, 혹시나 1935번 포트로 시도해서 실패하면 433 포트(RTMPS)나 80포트(RTMPT)로 재시도함.
RTMPT (RTMP Tunneled)	RTMP 데이터를 HTTP로 감싼 것. 기본 포트는 80번. HTTP 헤더 때문에 RTMP보다는 크기가 큼.
RTMPS (RTMP Secure)	RTMP 데이터를 HTTPS로 감싼 것. 플래시 재생기는 SSL 입출력을 지원하므로 그 기능을 사용함.
RTMPE (Encrypted RTMP)	128비트로 암호화된 RTMP. SSL보다는 가볍지만 SSL 인증 같은게 없음. 암호화 채널을 사용하기 때문에 기본 RTMP보다 약간 성능에 영향을 줄 수 있음.
RTMPTE (Encrypted RTMP Tunneled)	80번 포트 사용. RTMPT, RTMPE 섞어 놓은 형태. 플래시 플레이어 9,0,115,0 필요. 서버 성능에 영향을 줌.
RTMFP (Real Time Media Flow Protocol)	UDP에서 동작. 기본 RTMP는 TCP에서 동작. 항상 암호화 된 상태로 데이터를 전송.

기가 클 경우, RTMP 프로토콜은 조각 당 1 바이트의 헤더만을 실어 보내기도 한다. 그렇게 함으로써 부하를 상당 부분 줄일 수 있다. 그러나 실제로는 조각들은 보통 인터리빙되지 않는다. 대신, 인터리빙과 다중화는 패킷 수준에서 수행된다. 활성화된 여러 다른 채널에 실린 RTMP 패킷들은 각각의 채널의 대역폭 및 레이턴시(latency), 그리고 기타 QoS 요구에 맞게 인터리빙 된다. 이렇게 인터리빙 된 패킷들은 다시 나눌 수 없으며 조각 수준에서 다시 인터리빙 되지 않는다. RTMP 프로토콜은 패킷들을 주고받을 수 있는 여러 개의 채널들을 정의한다. 각 채널들은 다른 채널에 대해 독립적으로 동작한다.

3. 다중 플랫폼 지원 실시간 HD급 영상 전송기 구현을 위한 설계

3.1 다중 플랫폼 지원 실시간 HD급 전송기 하드웨어 구성

본 논문에서는 휴대가 가능하고 가격 경쟁력이 있는 다중 플랫폼지원 실시간 HD급 영상전송기를 구현하기 위해 RISC 칩인 T.I사의 DM368(432MHz) 프로세서를 이용하여 하드웨어를 설계하였다[12]. 개발 구현한 다중 플랫폼 지원 실시간 HD급 영상전송기의 하드웨어의 규격은 표 3과 같다.

3.2 다중 플랫폼 지원 실시간 HD급 전송기의 소프트웨어 구성

표 3에 따라 설계한 하드웨어 위에 Linux-2.6.29 를 탑재하였고 FFMpeg와 오픈소스 세그먼트(Open Source Segmenter)을 이용하여 HLS 프로토콜을 탑재하였고, RTSP서버와 RTMP, TS 스트리머를 탑재하였다. 클라이언트 디스플레이는 다중 플랫폼을 지원하기 위해 iPhone, Android phone, PC 등에서 별도의 앱이나 어플리케이션 프로그램 없이 동작하도록 설계하였다.

표 3 다중 플랫폼 지원 실시간 HD급 영상전송기의 하드웨어의 규격
Table 3 Hardware specification of multi-platform supporting real time HD level video streaming transmitter

항 목	규 격
CPU	T.I TMS320DM368 (432MHz)
RAM	128MB
NAND	64MB
Network	Ethernet 100M (RJ45)
Video Input	HDMI(1080i60, 720P30)
I/O	USB Host, Digital Input, Digital Output, GPS
Audio	AC97 CODEC, Input 1, Output 1, 44.1Khz
CODEC	H.264 high, mian, baseline profile
Power	5V, Rechargeable Battery

표 4 다중 플랫폼 지원 실시간 HD급 영상전송기의 소프트웨어구성
Table 4 Software specification of multi-platform supporting real time HD level video streaming transmitter

항 목	내 용
OS	Linux 2.6.29
File System	UBIFS, busybox
Device Driver	Ethernet, ALSA Audio, MFC H.264 codec 3G Modem Driver, GPS(UART), Digital I/O Camera, SD Card
Frame works	Avencoder, AACEncoder (Fixed Point AAC-LC) RTSP Server, HLS Server,RTMP Streamer, TS Streamer
Tool chain	Armv6 gcc compiler (eabi, armv6, vfp), Alsa Library

3.3 휴대용 다중 플랫폼 지원 실시간 HD급 전송기의 시스템 구성

휴대용 다중 플랫폼 지원 실시간 HD급 전송기 시스템의 그림 5과 같다. ARM9을 코어 프로세서로 하면서 H.264 인코더 기능을 하드웨어 내장한 T.I사 DM368 비디오프로세서를 중심으로 DDR2 SRAM, NAND FLASH가 메모리기능을 수행한다. Text LCD, Ethernet chip, USB Interface등이 주변장치로 기능을 수행한다. 비디오/오디오 실시간 전송을 수행하기 위한 비디오, 오디오 입/출력기는 HDMI 수신기와 디지털 오디오를 처리하기 위한 AC97 코덱을 사용하였다. 휴대용 다중 플랫폼 지원 실시간 HD급 전송기 시스템을 휴대가 가능하도록 전력공급 시스템을 DC 5V Battery와 이를 충전하기 위한 충전시스템으로 구성하였다.

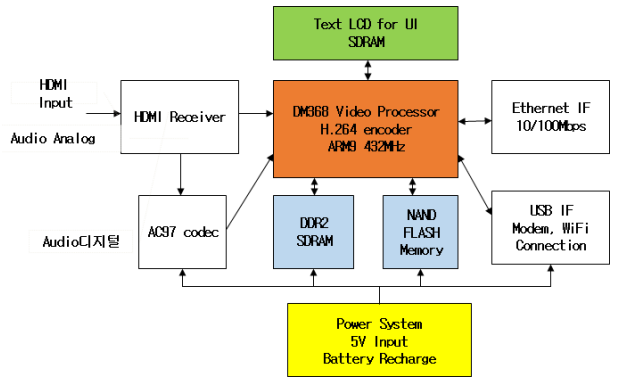

그림 5 다중 플랫폼 지원 실시간 HD급 전송기 시스템 구성도
Fig. 5 Block diagram of multi-platform supporting real time HD level video streaming transmitter

표 5 다중 플랫폼 지원 실시간 HD급 영상전송기의 개발툴 구축
Table 5 Development tool construction of multi-platform supporting real time HD level video streaming transmitter

Directory or file			내 용
Alsa			Alsa Library, Head file
Core		AACEncoder	Fixed point FFT using aac encoder
		Avencoder	Audio, Video capture Audio, Video encoding Writing to M-box format
		Middle ware	FFMpeg
		Porting Server	RTSP Server, HLS Server, FLV Server RTMP
Target	mytv	Lib	Modem driver module
		Bin	Avencoder, usb_mode_switch binary file
		Etc	Modem_switch data file
Start.sh			Booting executing script
Modem.sh			Modem(CHU-629S) start script

3.4 휴대용 다중 플랫폼 지원 실시간 HD급 영상전송기의 개발환경 구축

휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기를 개발하기 위한 개발툴 구축은 표 5와 같다.

3.5 FLV 서비스 Proxy화

비디오 전송장치는 관리를 위한 HTTP 서버가 있게 된다. 또한 HTTP live streaming 을 제외한 동영상 포맷은 각자의 서버가 필요하다. 동영상 전송장치가 공인IP 에 직접 접속되지 않고 NAT를 통과 하여하는 경우 일부 NAT 장치가 UPnP를 지원하지 못할 경우, port forwarding을 수동으로 하여야 하고, HTTP 서버 이외에 FLV 서버의 포트도 forwarding 해야 하므로, 사용자 불편 사항이 증가 하게 된다[13]. 이를 위해, HTTP 서버가 FLV 서버로 요청되는 서비스를 대행하는 모듈을 추가하여 HTTP 서버가 Flash 요구 시 FLV 서버에 재차 Flash를 요구하여 이를 사용자에게 전달해 주는 FLV proxy 기능을 수행 하도록 시스템을 설계 하였다. 일반적인 HTTP 문서요청과 HTTP Live 파일요청은 기존의 HTTP 서버의 기능을 이용하여 서비스 하도록 설계 하였다.

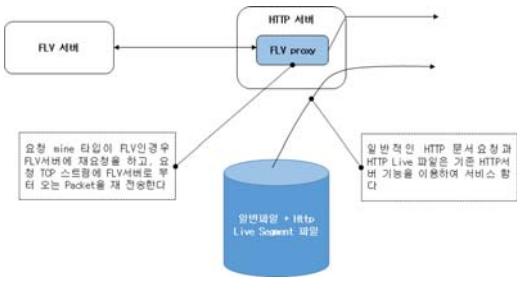

그림 6 HTTP서버의 FLV 프록시화
Fig. 6 FLV proxy of HTTP server

3.6 네트워크 구성

NAT 장치와 연결되어 있는 동영상 전송장치를 외부에서 접근하기 위해서는 다중 플랫폼 지원 실시간 HD급 영상전송기와 연결된 외부 IP 즉, 공인 IP를 알아야 한다[14]. 외부 IP는 NAT 장치를 검색하여 정보를 읽어 오거나, 외부 인터넷 망에 있는 STUN서버 접속을 통해서 알 수 있고, DDNS 서비스를 이용하여도 가능하다. NAT 장치 탐색 및 설정 절차는 그림 7과 같다.

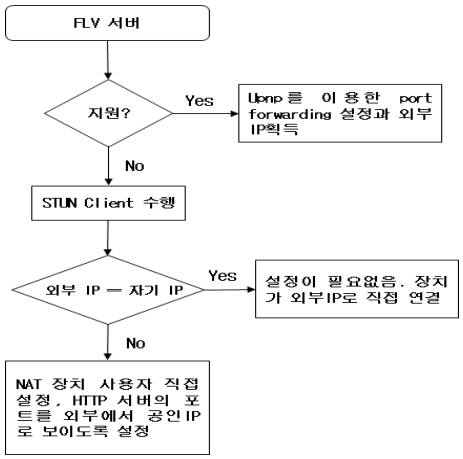

그림 7 네트워크 구성을 위한 NAT장치탐색 및 설정 절차도
Fig. 7 Processing block diagram of NAT device searching for network construction

4. 휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기 구현

휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기의 하드웨어 주요 회로도는 그림 8와 같다

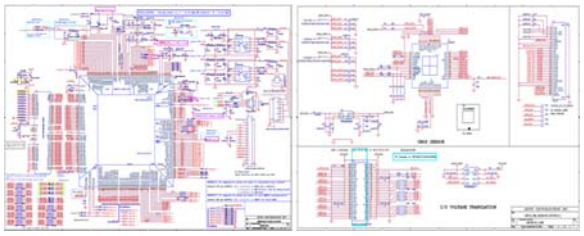

그림 8 휴대용 다중 플랫폼지원 실시간 HD급 영상전송기의 주요 회로도

Fig. 8 Main circuit of multi-platform supporting real time HD level video streaming transmitter

구현된 휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기의 외형은 그림 9과 같으며 다중 플랫폼 지원 실시간 HD급 영상 전송기의 소프트웨어 구성도는 그림 10과 같다.

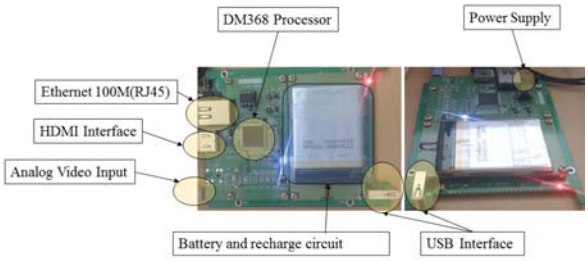

그림 9 다중 플랫폼 지원 실시간 HD급 영상 전송기의 외형

Fig. 9 Look of multi-platform supporting real time HD level video streaming transmitter

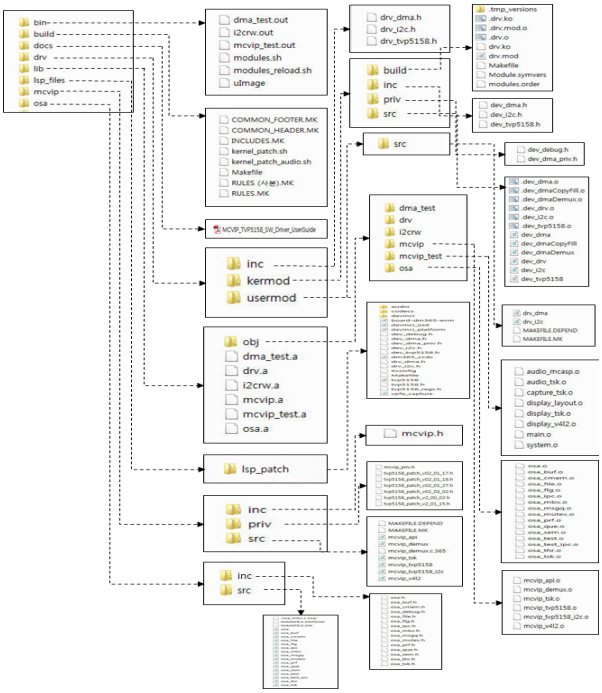

그림 10 다중 플랫폼 지원 실시간 HD급 영상 전송기의 소프트웨어 구성도

Fig. 10 Software block diagram of multi-platform supporting real time HD level video streaming transmitter

5. 휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기 테스트 환경구축 및 구현결과

5.1 휴대용 다중 플랫폼 지원 실시간 HD급 성능 테스트 환경구축

휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기 테스트 환경은 그림 11과 같은 블록도로 구성하였다. CATV Set-top Box의 HD 방송 출력신호를 휴대용 다중 플랫폼 지원 실시간 HD급 영상 전송기의 입력신호로 하였다. 휴대용 다중 플랫폼 지원 실시간 HD급 영상전송기의 출력신호인 실시간 스트림데이터를 상용 미디어서버에 전송하여 미디어서버 접속을 통해 PC(Personal Computer), Notework, Android Phone, iPhone, iPad/Android Pad에서 수신하도록 성능 테스트 환경을 구축하였다. 통신매체는 Wi-Fi, Wibro, LTE 통신모듈을 휴대용 다중 플랫폼 지원 실시간 HD급 영상전송기의 USB 포트에 장착하여 무선 통신기능을 수행 하였다.

5.2 휴대용 다중 플랫폼 지원 실시간 HD급 구현 결과

개발한 제품과 기존제품과의 성능비교를 위해 HD급 동영상을 방송하는 케이블 TV 신호와 HD급 영상재생기의 출력신호를 실시간 영상 전송기에 입력신호로 입력하였다. 실시간 영상전송기가 전송한 신호를 수신 단말기(smart phone, tablet PC, notebook)에서 수신한 영상을 각 프로토콜별로 영상데이터 프레임당 패킷손실비율을 통해 비교평가를 수행하였고 결과는 표 6과 같으며 HD급 영상전송기의 구현은 그림 12와 같다.

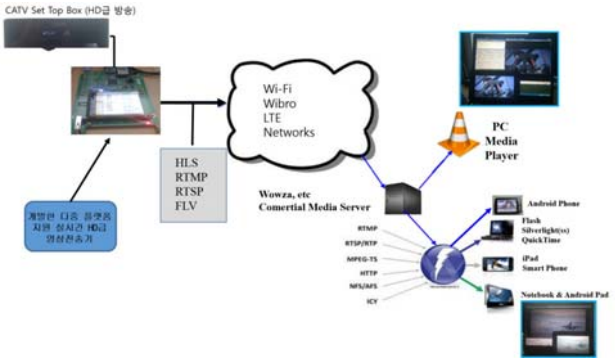

그림 11 다중 플랫폼 지원 실시간 HD급 영상전송기의 성능 테스트 환경 블록도

Fig. 11 Performance test environment block diagram of multi-platform supporting realtime HD level video streaming transmitter

6. 결 론

본 논문에서는 저가격, 저 전력소모의 다중 플랫폼을 지원하는 실시간 HD급 영상 전송기를 구현하기 위해 하드웨어를 설계하고 그 위에 실시간 비디오전송 프로토콜인 RTSP, RTMP, HLS 프로토콜의 탑재하였다. 하드웨어부분

그림 12 다중 플랫폼 지원 실시간 HD급 영상전송기의 구현화면

Fig. 12 Implement picture of multi-platform supporting real time HD level video streaming transmitter

표 6 기존제품과 개발제품의 성능비교

Table 6 Comparison of development product and exist product

프로토콜 / 수신기	RTSP/RTP		HLS		RTMP	
	기존 제품	개발 제품	기존 제품	개발 제품	기존 제품	개발 제품
Smart phone	2.02%	2.01%	1.53%	1.34%	1.62%	1.52%
Tablet PC	1.98%	1.97%	1.05%	1.01%	1.22%	1.17%
Notebook	1.05%	1.05%	0.99%	0.99%	1.03%	1.02%

에서는 실제로 경량화 및 비용절감을 위해 T.I사의 DM386 비디오 프로세서를 사용하였다. 소프트웨어 개발비용을 최소화하기 위해 운영체제로는 리눅스를 적용하였다. 다중의 플랫폼에서 동작하기 위해 수신자의 환경에 따라 비디오 스트림을 변형하여 전송하도록 구현하였다. 개발한 다중 플랫폼 지원 실시간 HD급 영상전송기의 성능을 검증하기 위해 연구실에서 성능평가 실험환경을 구축하였다. 성능실험을 위한 입력신호는 HD급 영상 재생기와 HD급 영상신호를 방송하는 케이블 TV, 두 가지 방식을 사용하였다. 입력 단에서 실시간으로 출력되는 HD급 영상신호를 개발한 다중 플랫폼 지원 실시간 HD급 영상 전송기에 입력하고 다중 플랫폼 지원 실시간 HD급 영상전송기가 RTSP/RTMP/HLS 프로토콜 규격에 맞는 비디오 스트림을 변환하여 유무선 인터넷망으로 전송한 후에 수신단에서 스마트폰, 태블릿 PC, 노트북을 이용하여 수신한 영상을 영상데이터 프레임당 패킷 손실을 비교한 결과 개발한 제품의 성능이 기존제품에 비해 우수함을 알 수 있었다. 본 연구에서는 다중 플랫폼 지원 실시간 HD급 영상전송기의 하드웨어 설계도면의 일부와 다중 플랫폼 지원 실시간 HD급 영상전송기의 소프트웨어의 계층 구조도를 제공하여 향후 다른 실시간 영상전송기 개발 시에 참고자료로 활용될 수 있을 것으로 기대된다. 개발한 제품의 활용분야는 가정 내의 감시 카메라, TV 영상의 실시간 전송 및 뿐만 아니라 실시간으로 현장의 위급상황을 현지 지휘소나 원거리 사령부에 전달해야하는 경우, 각 현장 참여 인원이 개발한 다중 플랫폼 지원 실시간 HD급 영상전송기를 몸에 착용하여 이동하면서 실시간으로 현장 상황영상을 전송하는 분야, 생방송에서 상황변화가 심한 방송의 경우 카메라맨이 실시간 HD급 영상 전송기를 착용하여 이동하면서 영상을 전송하는데 에 활용될 수 있을 것이다.

감사의 글

본 연구는 2015년도 동서울대학교 교내연구비 지원에 의해 수행되었습니다.

References

[1] R. Frederick, J. Geagan, M. Kellner, A. Periyann-an, Caching Support in RTSP/ RTP Servers, Draft-periyannan-rtsp-caching-01.txt, 2002.

[2] Schulzrinne. H., Real Time Streaming Protocol (RTSP), RFC 2326, 2004.

[3] chulzrinne. H., RTP: ATransport Protocol for Real-Time Applications, RFC 1889, 2002.

[4] M. Handley, V. Jacobson. “SDP: Session Description Protocol”, RFC2327..2004

[5] S. Casner, R. Frederick, and V. Jacobson, "RTP:A Transport Protocol for Real-Time Application," RFC1890, 1996.

[6] QuickTime Streaming Server http://www.apple.com-/quicktime/products/qtss

[7] http://helloworld.naver.com/helloworld/HTTPLive-Streaming.html

[8] Carson "iPhone HTTP Streaming with FFMpeg and Open Source Segmenter" 2009.

[9] Text of ISO/IEC FDIS 23002-3 "Information technology- MPEG video technologies - Part3: Representation of auxiliary video and supple -mental information" , 2007

[10] Carson " iPhone Windowed HTTP Live Streaming Using Amason S3 Cloudfront Proof of Concept, 2009.

[11] IA. Charles Beddoe and Ray Shapiro." Using Macromedia Flash as an embedded device UI“, Jul. 2005

[12] http://www.ti.com/lit/ds/symlink/tms320dm368.pdf-TMS320DM368 Digital Media System-on-Chip (DMSoC)

[13] S. Bradner, J. McQuaid, "Benchmarking Methodology for Network Interconnect Devices", RFC2544, Mar. 1999

[14] C. Demichelis, P. Chimento, "IP Packet Delay Variation Metric for IP Performance Metrics (IPPM)", IETF RFC 3393, Nov. 2002

저 자 소 개

이 재 희 (Lee, Jae Hee)

광운대학교 전자통신과 석사
광운대학교 전자통신과 박사
1987~1993년 : 국방과학연구소 연구원
1999년 3월 ~ 현재 : 동서울대학교 정보통신과 교수
관심분야 : Embedded System, Ubiquitous Network, Mobile IPTV, 영상신호처리
E-mail : ljh7314@dsc.ac.kr

서 창 진 (Seo, Chang Jin)

부산대학교 멀티미디어 석사
부산대학교 멀티미디어 박사
2013년 3월 ~ 현재 : 상명대학교 국방정보공학과 교수
관심분야 : Object Detection, Target Tracking, Artificial Vision, Multimedia, E-Learning, Security
E-mail : cjseo@smu.ac.kr