

휴대용 Http 라이브 스트리밍 전송기 구현에 관한 연구

A Study on the implementation of a Portable Http Live Streaming Transmitter

조 태 경[†] · 이 재 희^{*}
(Tae-Kyung Cho · Jea-Hee Lee)

Abstract - In this Paper, We proposed the HLS(Http Live Streaming) transmitter which is operated easily and cheap in all networks and client environments compared to the exist video live streaming transmitter. We analyzed the HLS protocol and then implemented for making the HLS transmitter cheaper and portable. After designing the HLS transmitter hardware using the RISC processor of Arm11 core, we ported the Linux Operating System and implemented the HLS protocol using the open source FFmpeg and Segmenter. For the performance evaluation of the developed HLS transmitter, we made the testing environment which is including the notebook, iPhone, and aroid Phone. In this testing environment, we analysed the received video data at the client displayer. As a results of the performance evaluation, we could certify that the proposed HLS transmitter has a higher performance than the Apple company's HLS.

Key Words : Http live streaming, MPEG2-TS, RTSP, m3u8, FFmpeg

1. 서 론

동영상 라이브 스트리밍(Video Live Streaming)이란, 촬영한 동영상 정보를 실시간으로 사용자의 동영상 플레이어로 보내 재생하도록 하는 기술을 말한다. 온 디맨드 스트리밍(On-Demand Streaming)에서는 촬영 이후에 편집이라는 과정을 거쳐 동영상 파일을 제작한 후에 사용자의 요구가 있을 때 동영상을 재생할 수 있도록 하지만, 라이브 스트리밍에서는 비디오와 오디오를 실시간으로 인코딩해 많은 사용자에게 동시에 보낼 수 있어야 한다는 점에서 차이가 있다.

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 갖추어야 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[1][2].

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.

[†] 교신저자, 정회원 : 상명대학교 국방정보공학과 교수 · 공박
E-mail : tkcho@smu.ac.kr

^{*} 비 회원 : 동서울대학교 정보통신공학과 교수 · 공박
접수일자 : 2014년 8월 6일
수정일자 : 2014년 8월 13일
최종완료 : 2014년 8월 18일

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)을 변경하지 않고 그대로 이용할 수 있다는 것이 HLS(Http Live Streaming)의 장점이다.

1.1 HLS(Http Live Streaming)서비스

HLS는 Apple사에서 iOS 3.0과 QuickTime-X를 위해 2009년에 발표한 프로토콜이다. 이 프로토콜에서는 스트리밍 데이터를 MPEG-2 Transport Stream에 담아 시간 단위로 단편화하여 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 타 업체들도 쉽게 HLS 방식을 지원할 수 있게 되었다. 기존의 라이브 스트리밍 방식과 HLS 방식을 그림 1과 그림 2에 나타냈다.

HLS방식과 기존의 스트리밍서버를 사용하는 방식은 동영상을 중단 없이 사용자의 동영상 플레이어에 전달한다는 점에서는 동일한 구조이다.

그림 1에서 Live H.264/ACC 와 그림 2의 서버 부분의 역할은 카메라로 촬영한 영상을 코덱(Codec)을 이용해 압축하여 서버로 보내는 기능을 수행한다. HLS와 Streaming Server방식의 차이점은 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 역할만을 한다. 요청받은 파일을 읽어서 어떠한 변형도 하지 않고, 읽은 그대로 보내기만 한다. 즉

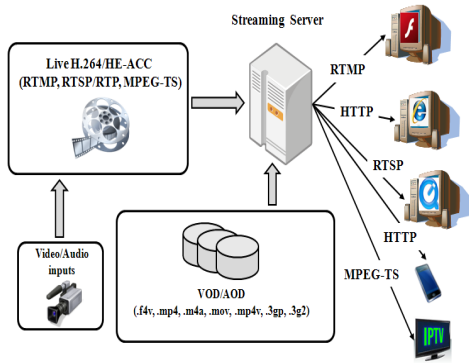


그림 1 기존의 스트리밍서버를 이용한 스트리밍 서비스
Fig. 1 Existed Streaming Service using Streaming Server

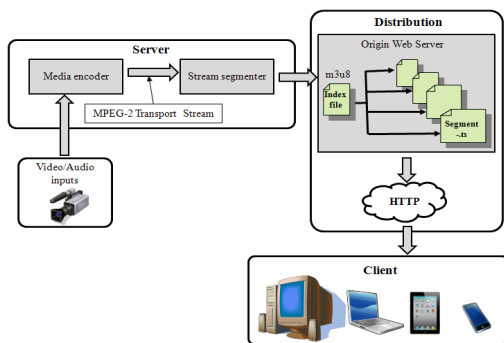


그림 2 HLS 이용한 스트리밍 서비스
Fig. 2 Streaming Service using HLS

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 서버가 데이터를 변형해 보내는 작업을 수행한다. 경우에 따라서는 특정 스트리밍방식을 위해 해당회사의 제품을 구매해야 하는 경우도 발생한다. 이는 웹 서버에 비하여 처리과정의 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 생성한다. Http는 양방향 방식이 아니기 때문에, 클라이언트에서 반드시 서버에 요청을 해야 그에 맞는 응답을 받을 수 있다. 즉, 단편화된 동영상과 다음에 디스플레이할 동영상정보를 함께 클라이언트에 전달한다.

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[1][3].

1.2 m3u8 파일 형식

m3u8 파일형식은 m3u 파일형식으로부터 유래한 것으로 m3u은 연속으로 재생할 mp3 파일 목록을 가진 플레이어

리스트 파일이다. 각 줄마다 재생할 파일의 경로를 기록한 매우 단순한 구조이다. 하지만 m3u에는 Latin-1 문자 집합만 사용할 수 있다. 그리고 단순히 파일 목록을 나열할 뿐이므로 재생할 파일에 대한 정보를 재생하기 전에 미리 알 수 없다. 그래서 나온 확장규격이 m3u8 파일형식이다. m3u8 파일과 TS(Transport Stream) 파일의 구조를 그림 3에 나타냈다.

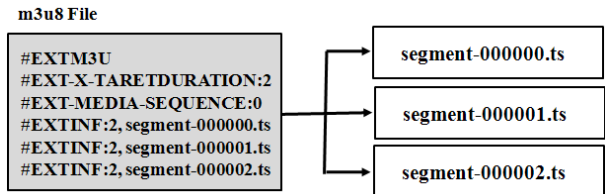


그림 3 m3u8파일과 ts파일

Fig. 3 m3u8 file and ts file

m3u8 포맷에서는 UTF-8 문자 집합을 사용할 수 있고, 여러 가지 지시어로 재생할 파일에 대한 추가적인 정보를 제공할 수도 있다. Http Live Streaming에서 m3u8 포맷을 사용하려면 몇가지 규칙을 따라야 한다. 우선 m3u8 파일의 가장 첫 줄은 #EXTM3U로 시작해야 한다. #EXTM3U라는 지시어로 이 파일은 m3u8 포맷을 사용한다는 것을 사용자의 플레이어에게 알려준다. 둘째는 모든 m3u8 포맷의 지시어는 표 1에 나타내었고, 줄 맨 앞에 #EXT로 시작해야 한다. #EXT로 시작하지 않으면 #이후의 문자열을 주석으로 간주한다.

표 1 m3u8 포맷 지시어

Table 1 The infinitive maker of m3u8

지시어	형식	내용
#EXTM3U	#EXTM3U	파일이 m3u8 포맷임을 명시한다.
#EXTINF	#EXTINF:<재생시간: 초>, <제목>	다음에 명시된 콘텐츠의 재생시간과 제목을 명시한다.
#EXT-X-TARGETDURATION	#EXT-X-TARGETDURATION:<시간: 초>	파일목록에 나열된 각 파일의 최대 재생시간을 명시한다.
#EXT-X-ENDLIST	#EXT-X-ENDLIST	플레이 리스트에서 재생할 콘텐츠가 더 이상 없음을 의미한다.
#EXT-X-DISCONTINUITY	#EXT-X-DISCONTINUITY	이 지시어가 표지된 줄을 기준으로 이전 줄과 이후 줄에서 재생하는 콘텐츠의 정보가 변경되었음을 표시한다.
#EXT-X-MEDIA-SEQUENCE	#EXT-X-MEDIA-SEQUENCE:<첫 파일의 일련번호>	제일 먼저 플레이해야 하는 파일의 일련번호를 명시한다.
#EXT-X-KEY	#EXT-X-KEY:<암호화 방법>[key]	암호화된 파일을 해독하는 키 값을 명시한다.
#EXT-X-STREAM-INF	#EXT-X-STREAM-INF	#EXT-X-STREAM-INF는 다음과 같은 정보를 제공한다. - BANDWIDTH: 10진수로 표시한 bps값 - PROGRAM-ID: 플레이 리스트 파일에 있는 콘텐츠가 갖는 고유값 - CODEC: 해당 콘텐츠에 적용되는 코덱 정보 - RESOLUTION: 해상도

1.3 MPEG-2 TS(Transport Stream)구조

MPEG-2 TS는 오디오나 비디오, 방송의 채널정보 등을 전

송하거나 저장하기 위해 정의한 규격으로, MPEG-2 기본 스트림을 패킷(Packet)으로 만들 때 오류정정 및 동기화 정보 등을 같이 포함할 수 있도록 하는 컨테이너(container)포맷이다. 실제로 TS로 데이터를 전송하는 경우 하나의 연결 내에 동시에 여러 개의 채널 정보를 담아서 전송할 수 있다. TS 패킷 구성은 그림 4에 나타냈다.

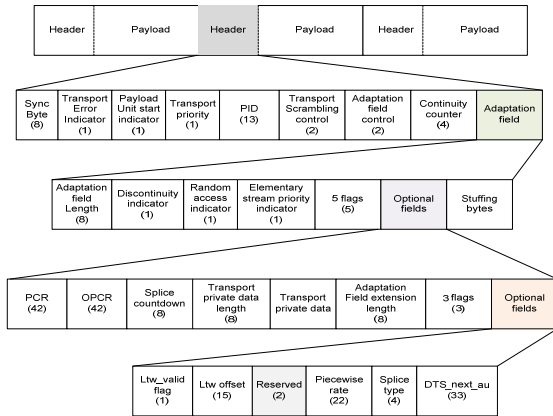


그림 4 TS Packet 구성도
Fig. 4 The block diagram of TS Packet

TS 패킷의 크기는 188bytes로 고정되어 있고, 이보다 큰 데이터는 모두 나누어진다. 동기화를 위한 정보, 시간정보, 오류정정, 스크램블링(Scrambling)정보 등을 헤더에 명시한다. TS의 페이로드에 포함하고 있는 데이터를 표 2에 나타냈다.

표 2 TS의 페이로드에 포함된 데이터

Table 2 The data of TS Payload

페이로드 데이터	설 명
Packetized Elementary Stream(PES)	네트워크전송을 위한 패킷의 크기로 나누어진 Elementary Stream. 분할된 첫 조각을 포함하는 패킷에는 PES헤더라는 정보가 포함되어 전송되며 이후에는 분할된 데이터만 포함된다.
Program Association Table(PAT)	프로그램의 번호와 프로그램 맵 테이블에 담고 있는 패킷의 PID(Packet Identifier)간의 연결 관계를 담고 있다.
Program Map Table(PMT)	프로그램의 Elementary Stream.을 담은 패킷에 대한 연결정보를 담고 있다.
Condition Access Table (CAT)	허가된 사용자만 해당 프로그램을 재생할 수 있게 하는 정보를 담고 있다 Scrambling을 사용할 경우 반드시 있어야 한다.
Network Information Table(NIT)	만드시 구현해야 하는 사항은 아니고, 내용 역시 사용자 임의대로 구성할 수 있다.

PAT, PMT, PES의 관계는 그림 5와 같다. PAT를 통해 어떤 채널이 있는지 목록을 얻고, PMT로 선택한 채널의 비디오/오디오 데이터가 어떤 ID를 갖고 있는지 알아내어 해당 ID를 가진 패킷만 선택하여 재생한다.

1.4 HLS를 위한 ts 파일 생성

HLS에서 사용하기 위한 ts 파일은 MPEG-2 TS를 순서대로 저장한 파일이다. 다만 정한 시간에 따라 분할하여 저장한다. HLS로 화면을 자연스럽게 재생하려면 각 ts 파일이 I-frame을 포함해야 한다. 첫 비디오 데이터가 I-frame인

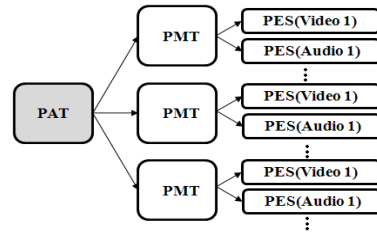


그림 5 PAT, PMT, PES의 상관관계
Fig. 5 The relative relationship of PAT, PMT, PES

것이 좋다. 첫 화면이 I-frame이 아니면 플레이어에서 화면이 나오지 않거나 일시적으로 뭉개진 화면이 나올 수 있다. I-frame은 데이터의 크기가 크기 때문에 ts파일이 포함하는 데이터 재생 시간 간격을 적절하게 선택해야 한다. HLS를 위해 권고하는 미디어의 규격은 표 3과 같다.

표 3 HLS를 위한 미디어 규격 권고안

Table 3 The recommendations of media standard for HLS

	비디오	오디오
코덱(Codec)	-iPhone 3G/iPod 2세대 이상 H.264 Baseline 3.1이하 -iPad, Apple TV, iPhone Baseline profile 3.0/3.1	-HE-AAC 혹은 ACC-LC, 스테레오 -MP3, 스테레오
Frame rate /Bit rate	-200 kbps 미만: 10 fps -200 ~ 300 kbps: 12 ~15 fps -기타 : 29.97	-Sampling Rate: 22.05 khz -비트레이트: 40 kbps

추천하는 각 ts 파일의 길이는 플레이어에서 10초간 재생할 수 있는 분량이다. 재생할 수 있는 분량이 짧아지면 네트워크로부터 받아와야 하는 데이터의 양이 증가한다. 이 경우에는 I-frame의 빈도가 늘어나게 되어 실질적인 데이터 양도 늘어나고 전송 프로토콜의 헤더와 같은 추가적인 전송 데이터 및 갱신되는 index파일은 m3u8 파일을 받기위한 데이터의 양도 같이 늘어났다. 표 4는 Apple사가 권장하는 HLS 서비스의 규격 값이다[4][5][6].

표 4 Apple사의 HLS서비스 권고안

Table 4 The recommendations of Apple company for HLS

통신망	해상도	비디오 rate	키프레임 수
Celluar	480x300	110 kbps	30
WiFi	640x480	600 kbps	90

1.5 HLS의 동작 방식

서버에서 HLS 서비스가 시작되면 m3u8 파일은 분할된 파일이 생길 때마다 계속 주기적으로 바뀐다. 그림 6을 보면 처음에는 2, 3, 4의 일련번호를 갖는 파일이 있다. #EXT-X-TARGETDUARATION이 2로 명시되어 있으므로 2초 후에는 5번 파일이 생긴다. m3u8파일의 뒤에 5번 파일을 가리키는 줄을 추가할 수도 있다. 하지만 줄을 추가하여 m3u8파일에 명시된 콘텐츠가 늘어나면 5번 파일이 생기고 난 뒤에 스트리밍을 요청하는 플레이어는 콘텐츠 재생을 위해 2, 3, 4, 5의 4개의 파일을 요청하게 된다. 이는 초기 시작을 위한 대기시간을 늘리고, 실제 동영상과 플레이어에서

재생하는 동영상의 시간 차이를 늘리게 된다. 그래서 일정한 개수의 파일만을 m3u8파일에 명시하는 것을 권장하고 있다[4][5].

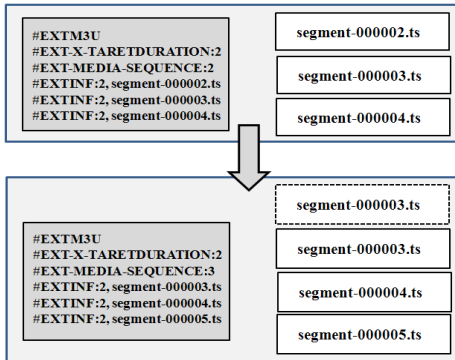


그림 6 시간 흐름에 따른 m3u8파일의 변화
Fig. 6 The variation of m3u8 file according to the flowing of time

2. HLS 전송기 구현

2.1 HLS전송기 하드웨어 구성

본 논문에서는 휴대가 가능하고 가격 경쟁력이 있는 HLS 전송기를 구현하기 위해 RISC 칩인 Arm11 코어 프로세서를 사용하여 하드웨어를 설계하였다. 구현한 HLS 전송기의 하드웨어 규격을 표 5에 나타냈다.

표 5 HLS 전송기 하드웨어 항목규격
Table 5 The Hardware item spec. of HLS transmitter

항목	내용
Video Input	Composite Video NTSC/PAL 3 RCA Jack (Video, Audio Left, Right)
Video Output	Composite Video NTSC/PAL 4 Pole phone jack (optional)
CPU	ARM11 660 Mhz
Flash	64 Mbyte
SDRAM	128 Mbyte
Storage	Micro SD Card Socket (optional)
Port	Ethernet 10/100 Base T
	IR Transmitter Receiver
	USB Host 1.1
	USB OTG 2.0 (optional)
Power	DC 12V

2.2 HLS 전송기의 소프트웨어 구성

본 연구에서는 구현한 하드웨어 상에 Linux-2.6.2를 탑재하였고 FFMpeg와 오픈소스 세그먼트(Open Source Segmenter)를 이용하여 HLS 프로토콜을 탑재하였다[5]. 클라이언트 디스플레이는 iPhone, Android phone, PC 등에서 별도의 앱이나 응용프로그램 없이 동작하도록 설계하였다. HLS 전송기의 소프트웨어 규격을 표 6에 나타냈다.

표 6 HLS 전송기 소프트웨어 구성항목 및 규격
Table 6 The software item and spec. of HLS transmitter

항목	내용	
OS	Linux2.6.2	
Video	codec	H.264
	resolution	VGA, QVGA 30fps
	bitrate	384 kbps, 512 kbps, 1 Mbps, 1.5M bps
Streaming	Audio	2 channel AAC
	Streaming format	HTTP Live Streaming
Client Display Device	Iphone, Android, PC. No need special app. Easy operation using web browser -Setting streamer (Video Type, Quality,Network)	
Network	Ethernet, USB를 이용한 WiFi Upnp (라우터 Dynamic DNS Support (DDNS Client)	

2.2.1 FFMpeg 개발 및 테스트절차

HLS을 구현하기 위한 개발 및 테스트 절차 과정은 그림 7과 같다[3][4].

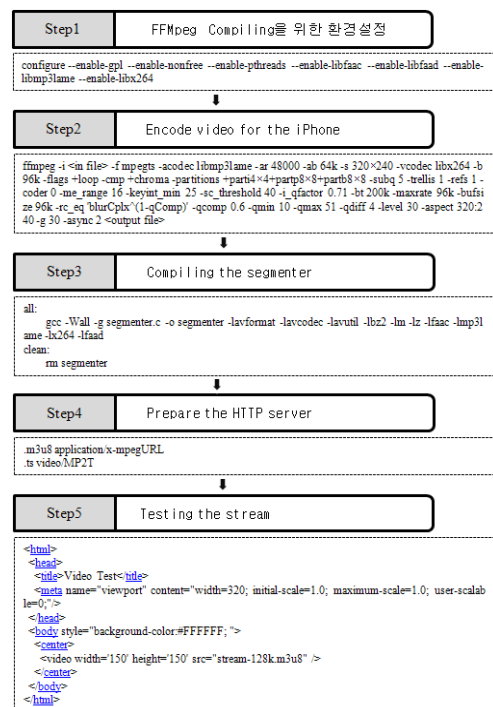


그림 7 HLS 구현을 위한 개발 및 테스트 절차도
Fig. 7 The procedure diagram of HLS implement and testing

3. HLS전송기 동작검증 및 결과분석

3.1 HLS 전송기 하드웨어 구현

본 논문의 표 5에 언급된 항목을 근거로 설계 구현한 휴대용 HLS 전송기의 하드웨어 외형 사진은 그림 8과 같다.

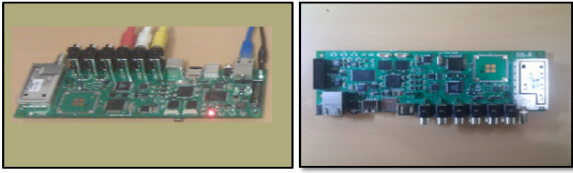


그림 8 HLS 하드웨어 구현 외형
 Fig. 8 The pictures of the HLS Hardware

3.2 HLS 전송기의 성능테스트 환경구성

구현한 휴대용 HLS 전송기의 성능을 검증하기 위한 테스트 환경을 그림 9에 도시하였다.

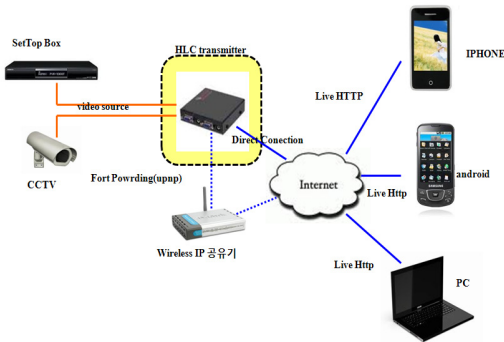


그림 9 HLS 테스트 환경
 Fig. 9 HLS transmitter testing environment

3.3 HLS 전송기의 동작테스트 결과분석 및 검증

본 논문에서 구현한 HLS 전송기의 동작테스트는 Cable TV 방송에서 영상을 수신하여 HLS 전송기의 입력신호로 하고 HLS 전송기에서 무선통신망(3G)과 WiFi 무선망으로 전송하여 수신된 영상을 Notebook과 iPhone에서 동시에 재생하는 것으로 확인하였고 이때 사용한 영상 파라메타는 표 7에 나타냈다.

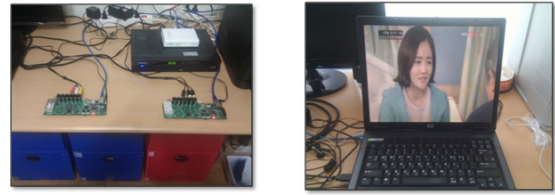
표 7 HLS 전송기 동작테스트에 사용한 영상 변수
 Table 7 Image parameter for testing of HLS Transmitter

통신망	해상도	비디오 rate	키프레임수
3G	480x300	150 kbps	40
WiFi	640x480	800 kbps	120

또한 표 4에 나타난 Apple사에서 권장하는 HLS 서비스를 위한 최소한의 권장규격을 만족하는지를 수신된 영상을 분석하여 개발한 전송기의 성능을 검증하였다.

개발한 HLS 전송기를 통해 전송한 동영상을 사용자단에서 분석한 결과, Apple사의 권고안보다 상회하는 결과를 보였다.

그림 10은 개발된 HLS 전송기의 테스트환경과 Notebook에서 수신된 HLS 전송기의 화면이다.



개발된 HLS전송기의 테스트환경 Notebook 에서 전송된 동영상 수신화면

그림 10 HLS 전송기의 동작 테스트 환경 화면
 Fig. 10 The picture of HLS transmitter operation testing environment

그림 11은 본 논문에서 개발한 휴대용 HLS 전송기의 성능을 iPhone과 안드로이드 폰에서는 초기 화면 설정화면과 전송된 화면이 Notebook과 스마트폰에서 동시에 디스플레이 되는 것으로 HLS 전송기의 성능을 검증하였다.



Android Phon 동작화면 수신된 영상이 DMB 방송이 아니라 Cable TV 방송임을 알 수 있음 iPhone 에서 초기화면 설정 및 수신된 영상



수신된 HLS전송기에서 보낸 영상이 Notebook 과 Smart Phone에서 동시에 수신된 영상

그림 11 HLS 전송기의 동작테스트 결과 화면
 Fig. 11 The picture of HLS transmitter operation result

4. 결 론

본 논문에서는 휴대가 가능한 HLS 전송기를 구현하기 위해 Http Live Streaming 전송방식에 대해 연구, 분석 하였고 휴대용 HLC 전송기를 경량 및 저렴한 비용으로 구현하기 위해 Arm 11프로세서를 이용하여 임베디드 보드를 개발하였다. 개발한 하드웨어 위에 리눅스를 탑재하였고 FFMpeg과 Open Source Segmenter를 이용하여 HLS 전송기의 기능을 구현하였다. 개발된 HLS 전송기 성능을 검증하기 위해 HLS 전송기의 동작테스트 환경을 구축하여 성능을 검증하였다. 테스트 환경은 CATV에서 수신된 영상을 개발한 HLS 전송기에 입력하고 HLS 전송기가 Http 라이브 비디오 스트림을 무선통신망(3G)과 WiFi망으로 전송한 후에 그 스트림을 Notebook, iPhone, android Phone에서 수신하여 디스플레이하는 방식으로 시나리오를 구성하였다.

본 논문에서는 동작 검증 시나리오에 따라 수신된 동영상
을 분석하여 Apple사 HLS 서비스 권고안보다 상회하는 성
능을 얻었고, 전송기와 클라이언트의 수신화면을 제시하여
본 논문의 연구결과를 증명하였다. 향후에는 본 연구를 확
장하여 HD급 화질의 실시간 비디오 스트림 전송기를 개발
하고자 한다.

References

- [1] [http://helloworld.naver.com/helloworld/HTTP Live Streaming.html](http://helloworld.naver.com/helloworld/HTTP%20Live%20Streaming.html)
- [2] Taeho Choi, "Fast reconnection of adaptive HTTP streaming in heterogeneous wireless networks", Hanyang University, 2012
- [3] Seonghui Jeon, "Design and implementation of Smart Agent System based on the adaptive streaming, Seoul Women's University, 2013
- [4] <http://developer.apple.com/library/ios/#documentation/StreamingMediaGuide/Introduction.html>
- [5] carson "iPhone HTTP Streaming with FFMpeg and Open Source Segmenter" June 28, 2009
- [6] carson "iPhone Windowed HTTP Live Streaming Using Amason S3 Cloudfront Proof of Concept, July 2009

저 자 소 개



조 태 경 (趙 兌 璟)

1984년 한양대학교 전자통신공학과 졸업
1986년 한양대학교 전자통신공학과 석사
2001년 한양대학교 전자통신공학과 박사
2003년~현재 상명대학교 국방정보공학과
교수

E-mail : tkcho@smu.ac.kr



이 재 희 (李 在 熙)

1985년 2월 광운대학교 전자통신과 졸업
1987년 2월 광운대학교 전자통신과 석사
2000년 2월 광운대학교 전자통신과 박사
1987~1993년 국방과학연구소
1993년~1999년 대덕대학 정보통신과 교수
1999년~현재 동서대학교 정보통신과
교수

E-mail : ljh7314@dsc.ac.kr