

임베디드 리눅스 기반의 휴대용 Http 라이브 스트리밍 전송기 구현에 관한 연구

이재희*, 조태경**

동서대학교 정보통신공학과*, 상명대학교 국방정보공학과**

The Study on the implementation of Portable Http Live Streaming Transmitter based the Embedded Linux

Jea-Hee Lee*, Tae-Kyung Cho**

Dept. of Information and Communication Engineering, Dong-Seoul Univ.*

Dept. of National Defense Intelligence Engineering, Sang-Myung Univ.**

요 약 기존 비디오 라이브 스트리머에 비해서 매우 저렴하고 저전력으로 휴대용으로 쉽게 동작하는 HLS(Http Live Streaming) 전송기를 임베디드 리눅스 기반으로 개발 및 구현하였다. HLS 전송기를 저렴하게 구현하기 위해 Arm11 프로세서 위에 임베디드 리눅스를 운영체제로 포팅 하였고 오픈 소스인 FFmpeg와 Segmenter을 이용하여 HLS 프로토콜을 구현하였다. 노트북, iPhone, android phone, 노트북을 수신단말기로 이용하여 개발한 임베디드 리눅스 기반의 휴대용 HLS 전송기의 성능을 평가하였다. 본 논문에서 개발한 HLS 전송기는 애플사가 제안한 HLS 권고 규격 이상의 성능을 구현하였음을 성능평가 실험을 통해 확인하였다. 본 논문에서 개발한 제품의 활용분야는 실시간으로 현장의 위급상황을 현지 지휘소나 원거리 사령부에 전달해야하는 경우 HLS를 개인이 착용하여 이동하면서 실시간으로 상황영상을 전달하는 분야에 활용될 수 있을 것으로 생각된다.

주제어 : 임베디드 리눅스, Http 라이브 스트리밍, 실시간 영상전송기, m3u8, FFmpeg

Abstract In this paper, for developing and implementing the HLS(Http Live Streaming) transmitter based embedded linux which is operated easily and cheap and lower power, portable in all networks and client environments compared to exist video live streaming transmitters. We design the developed HLS transmitter hardware using the Arm11 core and then porting the Embedded Linux OS(Operating System) and implementing the HLS protocol using the open source FFmpeg and Segmenter. For proving the performance of developed HLS transmitter, we make the testing environment for testing the performance of HLS transmitter including the notebook, iPhone, android Phone, Notebook and then analysis the received video in the client displayer. In this paper, we suggest the developed HLS transmitter performance data values higher than the Apple company's HLS recommended specification values and the picture of developed HLS transmitter operation in the testing environment. The application field of this paper result is that the man who works in the emergency situation take HLS and transmit the live emergency situation to the head quarter using this portable HLS.

Key Words : Embedded Linux, Http Live Streaming, Real time video transmitter, m3u8, FFmpeg

Received 7 September 2015, Revised 30 October 2015

Accepted 20 November 2015

Corresponding Author: Jea-Hee Lee(Dept. of Information and Communication Engineering, Dong-Seoul University)

Email : ljh7314@du.ac.kr

ISSN: 1738-1916

© The Society of Digital Policy & Management. All rights reserved. This is an open-access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>), which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. 서론

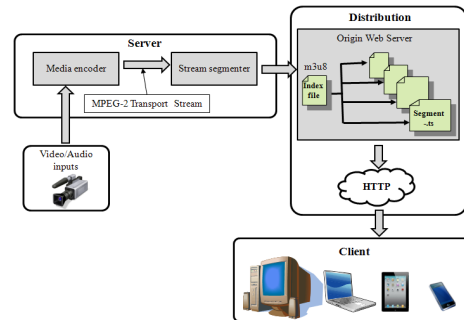
비디오 라이브 스트리밍(Video Live Streaming)이란 촬영한 동영상상을 실시간으로 사용자에게 전송하여, 사용자 단말에서 재생할 수 있도록 하는 것을 의미한다. 온디맨드 스트리밍(On-Demand Streaming) 방식은 촬영한 영상을 편집 과정을 거친 후 사용자가 요구했을 경우에만 동영상상을 전송하지만 라이브 스트리밍에서는 비디오와 오디오 정보를 실시간으로 인코딩하여 많은 사용자에게 동시에 보내야 한다. 라이브 스트리밍을 위한 기존의 방식은 RTSP(Real-Time Streaming Protocol)/RTP(Real-Time Transport Protocol), RTMP(Real-Time Messaging Protocol) 등이 있다[1,2,3].

이와 같은 방식을 사용하는 스트리밍 서버는 동영상상 영상 데이터의 전송 이외에도 동영상의 정보 값을 보내야 하며 또한 전송규격에 맞추어 동영상 파일을 변형하는 기능도 제공하여야 한다. 이러한 기능을 구현하려면 일반 웹 서버에 비해 도입 비용이 상대적으로 높고 RTSP/RTP의 경우에는 RTSP와 RTP가 서로 다른 통신망을 통해 정보를 교환하기 때문에 방화벽 또는 NAT(Network Address Translator)를 사용하고 있는 통신망에서는 서비스가 원활하게 동작하지 않는다는 문제점이 있다[4,5]. 따라서 이러한 문제를 해결하기 위한 방법은 Http 프로토콜을 전송 채널로 이용하는 것이다. Http 방식은 양방향 전송방식이 아니므로 라이브 스트리밍을 위해서는 추가적인 기술이 요구되지만, 방화벽에서 Http 서버로의 요청만 통과시키면 되므로 방화벽의 설정이 매우 간단하다. 또한 Http 방식은 웹 서비스를 위한 캐시 구조를 변경 없이 사용할 수 있고, 기 구축되어 있는 CDN(Content Delivery Network)을 변경하지 않고 그대로 이용할 수 있다는 장점이 있다[6,7].

1.1 HLS(Http Live Streaming)서비스

HLS는 2009년에 Apple사에서 iOS 3.0과 QuickTime-X를 위해 발표한 프로토콜이다. 이 프로토콜은 스트리밍 데이터를 MPEG-2 Transport Stream에 담고, 이를 시간 단위로 잘게 쪼개어 전송하는 방식이다. 그리고 파일 재생에 관한 정보는 m3u8 파일을 이용하여 사용자에게 전달한다. 이러한 특징을 갖고 있는 HLS는 iPhone 사용자의 많아짐에 따라 그 수요가 증가하고 있으며 또한 규격

자체의 단순함과 IETF를 통한 표준화 작업에 힘입어 많은 업체들이 쉽게 HLS를 지원할 수 있게 되었다[8,9,10].



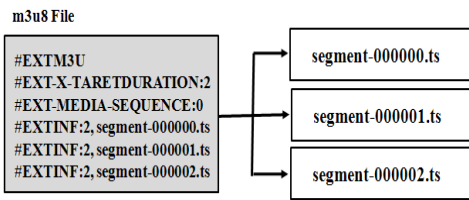
[Fig. 1] The streaming service using HLS

[Fig. 1]에 나타낸 HLS를 사용한 스트리밍 서비스에 서 서버는 Http로 요청을 받아서 사용자의 플레이어에 응답을 주는 역할만 한다. 즉 수신자가 요구한 파일을 읽어서 어떠한 변경도 하지 않고, 읽은 그대로 응답에 포함해 보낸다. 즉 저장하고 있는 파일을 Http 응답을 통해 보낼 수 있는 웹 서버이면 HLS를 사용한 스트리밍 서비스를 제공할 수 있다. 그러나 스트리밍 서버 방식의 경우에는 수신 장비의 규격에 맞게 데이터를 수정하여 보내야 하며, 경우에 따라서는 특정 스트리밍 방식을 수용하기 위해 해당 회사의 제품을 구매해야만 한다.

스트림 세그먼트는 입력받은 미디어 데이터를 일정 시간 간격에 따라 분할해서 파일을 만들고, 그 분할한 파일에 접근할 수 있는 메타 데이터인 m3u8 파일을 만든다. Http는 양방향 통신방식이 아니기 때문에, 클라이언트가 서버에게 데이터를 요청을 해야만 그에 상응하는 응답 데이터를 받을 수 있다.

1.2 m3u8 파일형식

m3u8 파일의 형식은 m3u 파일형식을 기반으로 형식화한 것이다. m3u는 연속 재생할 mp3 파일의 목록을 갖는 플레이어 리스트 파일이며, 각 줄마다 재생할 파일의 경로를 나타낸 단순한 구조이다. 하지만 m3u에는 Latin-1 문자 집합만 사용할 수 있으며, 단순히 파일 목록만을 나열할 수 있기 때문에 재생할 파일에 대한 정보를 재생 전에 미리 알 수가 없다. 이러한 문제를 보완한 확장 규격이 m3u8 파일형식 이다[11,12].



[Fig. 2] The file format of m3u8 and ts file

m3u8 포맷의 경우에는 UTF-8 문자 집합을 사용하며, 지시어를 이용하여 재생할 파일에 대한 추가 정보를 제공한다. m3u8 포맷에서 사용하는 대표적인 지시어를 <Table 1>에 나타냈다.

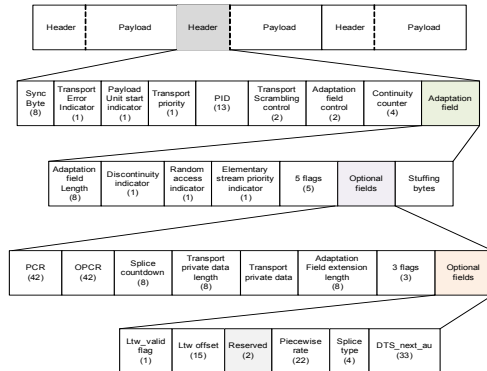
<Table 1> The reference instruction command of m3u8 format

Reference Instruction	Format	Meaning
#EXTM3U	#EXTM3U	Notify the m3u8 file and express the first line
#EXTINF	#EXTINF:<display time:second>, <name>	Notify the display time and name of contents
#EXT-X-TARGETDURATION	#EXT-X-TARGETDURATION:<time:second>	Notify the maximum play time of each file in file list
#EXT-X-ENDLIST	#EXT-X-ENDLIST	Notify that is no more the play contents in the play list
#EXT-X-DISCONTINUITY	#EXT-X-DISCONTINUITY	Notify that the play information is changed after this reference Ins.
#EXT-X-MEDIA-SEQUENCE	#EXT-X-MEDIA-SEQUENCE:<the list number of first file>	Notify that the list number of file must play in advance
#EXT-X-KEY	#EXT-X-KEY:<Encryption method >[key]	Notify that the decoding key value of encryption file
#EXT-X-STREAM-INF	#EXT-X-STREAM-INF	#EXT-X-STREAM-INF suggest information as follow -BANDWIDTH : bps values based on decimal -PROGRAM-ID : the id of contents file in the play list -CODEC : the information of codec applied content - RESOLUTION: the play content resolution

1.3 MPEG-2 TS(Transport Stream)구조

MPEG-2 TS는 MPEG-2 Part1, System(ISO/IEC

13818-1, ITU-T H.222.0)로 오디오나 비디오, 방송의 채널정보 등을 전송하거나 저장하기 위해 정의한 규격이며, MPEG-2 기본스트림을 패킷으로 만들 때 오류 정정 및 동기정보 등을 포함하는 컨테이너(Container)포맷이다. 실제 TS를 이용하여 데이터를 전송할 경우에는 하나의 연결 내에 다수 개의 채널 정보를 포함하여 전송할 수 있다. TS 패킷 구성은 [Fig. 3]과 같다[13,14,15].



[Fig. 3] The form of TS Packet

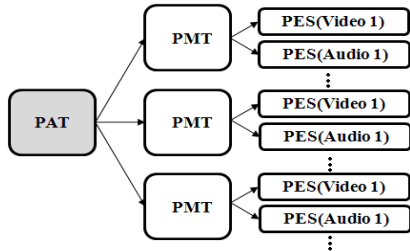
TS 패킷 하나의 크기는 188 byte이며, 이보다 큰 데이터는 모두 나누어 전송한다. 헤더에는 동기정보, 시간정보, 오류정정 정보, 스크램블링 정보 등을 명시한다.

<Table 2> The data of payload TS

Payload Data	Meaning
Packetized Elementary Stream (PES)	The packet include the divided first elementary involve the PES header information and then involve the divided data
Program Association Table(PAT)	The connection relationship between PID(Packet Identifier)s that involve the information of program number and program map table
Program Map Table (PMT)	The connection information of packet included that PES(Packet Elementary Stream) of program
Condition Access Table (CAT)	The table information of program can be played by the admitted user. It must be declared at the scrambling
Network Information Table (NIT)	This information table is not necessary item and can be chosen and used by user's own discretion

PAT, PMT, PES의 대략적인 관계는 [Fig. 4]와 같다. PAT를 통해 채널의 목록을 얻을 수 있으며, PMT로부터

선택한 채널의 비디오/오디오 데이터가 갖는 ID를 인식한 후 해당 ID를 갖는 패킷만을 선택하여 재생한다.



[Fig. 4] The correlation of PAT, PMT, PES

1.4 HLS를 위한 ts 파일 생성

HLS에서 사용하는 ts 파일은 MPEG-2 TS를 순차적으로 저장한 파일이며, 이는 정한 시간에 따라 분할하여 저장한다. HLS를 사용하여 동영상 재생을 하려면, 각 ts 파일이 I-frame을 포함하고 있어야 한다. 일반적으로 첫 번째 비디오 데이터가 I-frame이어야 좋으며, 만약 첫 번째 화면이 I-frame이 아닌 경우에는 영상이 보이지 않거나 일시적으로 영상이 뭉개져 보일 수 있다. I-frame은 데이터의 양이 크기 때문에 ts 파일이 포함하는 데이터 재생시간 간격을 적절히 조절해야 한다. HLS를 위해 권고하는 미디어의 규격은 <Table 3>과 같다.

<Table 3> The recommend media specification for HLS

	Video	Audio
Codec	-iPhone 3G/iPod 2G H.264 Baseline 3.1G -iPad, Apple TV, iPhone Baseline profile 3.0/3.1	-HE-AAC, ACC-LC, Stereo -MP3, Stereo
Frame rate / Bit rate	-200kbps below : 10 fps -200 ~ 300kbps : 12 ~15 fps -etc : 29.97	-sampling rate : 22.05 khz -bit rate : 40 kbps

일반적으로 사용하는 ts 파일의 길이는 플레이어에서 10초간 재생할 수 있는 분량이다. 재생할 수 있는 분량이 짧아지면 I-frame이 자주 발생하게 되어 전송 데이터양이 커지게 된다.

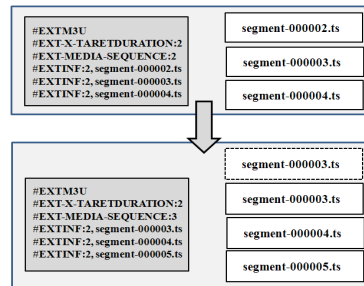
<Table 4>는 Apple사가 권장하는 HLS 서비스의 규격 값이다[10].

<Table 4 > The recommend value of Apple's HLS service

Network	Resolution	Video rate	Key frame number
Celluar	480x300	110kbps	30
WiFi	640x480	600kbps	90

1.5 HLS의 동작

서버에서 HLS 서비스가 시작되면 m3u8 파일은 분할된 파일이 발생할 때마다 주기적으로 바뀐다. [Fig. 5]을 보면 처음에는 2, 3, 4의 일련번호를 갖는 파일이 있다. #EXT-X-TARGETDURATION이 2로 명기되고 이는 2초 후에는 5번 파일이 생성된다는 것을 의미한다. m3u8 파일의 뒤에 5번 파일을 가리키는 줄을 추가할 수도 있다. 그러나 줄을 더하여 m3u8 파일에 명시된 콘텐츠가 증가하면 5번 파일이 생성되고, 스트리밍을 요청하는 플레이어는 콘텐츠 재생을 위해 2, 3, 4, 5의 4개 파일을 요청한다. 이러한 절차는 초기 시작을 위한 대기시간을 크게 하며, 실제 동영상과 플레이어에서 재생하는 동영상의 시간 차이를 크게 한다. 이러한 문제점을 해결하기 위해 m3u8 파일에는 제한된 개수의 파일만을 명시하라고 권장한다[16].



[Fig. 5] The variance of m3u8 file according to the time flowing

2. HLS 전송기 구현

2.1 HLS 전송기 하드웨어 구성

본 논문에서는 휴대가 가능하고 가격 경쟁력이 있고 전력소모를 최소화 하는 HLS 전송기를 구현하기 위해 ARM사의 RISC 칩인 Arm11 코어 프로세서를 사용하였으며, 하드웨어의 상세한 규격은 <Table 5>에 나타났다.

<Table 5> The hardware item specification of HLS transmitter

Item	Specification
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 660Mhz
Flash	64Mbyte
SDRAM	128Mbyte
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

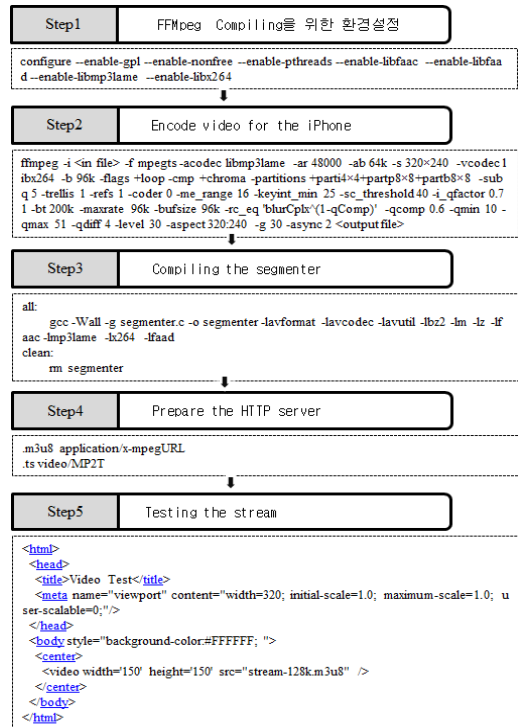
<Table 5>에 따라 설계한 하드웨어에 임베디드 리눅스 커널 2.6.2를 탑재하였고 FFMpeg와 오픈소스 세그먼트 (Open Source Segmenter)를 사용하여 HLS 프로토콜을 구현하였다. 클라이언트 디스플레이는 iPhone, Android phone, 노트북 등에서 별도의 앱 또는 응용 프로그램 없이도 동작할 수 있도록 설계하였다.

<Table 6> The software item specification of HLS transmitter

Item	Specification	
OS	Embedded Linux 2.6.2	
Streaming	Video	codec H.264 resolution VGA, QVGA 30fps bitrate 384Kbps, 512Kbps, 1Mbps, 1.5Mbps
	Audio	2channel 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, USB using WiFi Upnp (Router Dynamic DNS Support (DDNS Client)	

2.3 FFMpeg 개발 및 테스트

HLS을 구현하기 위한 개발 및 테스트는 [Fig.6]과 같이 진행하였다[8].

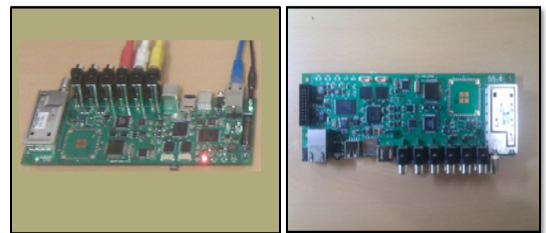


[Fig. 6] The testing and developing procedure diagram for implementing HLS

3. HLS전송기 동작 검증 및 분석

3.1 HLS 전송기 하드웨어 구현

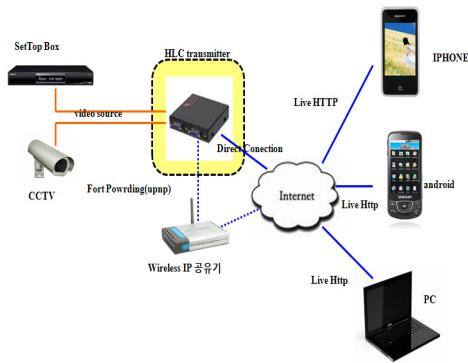
본 논문에서 구현한 휴대용 HLS 전송기의 H/W를 [Fig. 7]에 나타냈다.



[Fig. 7] The picture of developing HLS hardware

3.2 HLS 전송기의 성능검증 환경

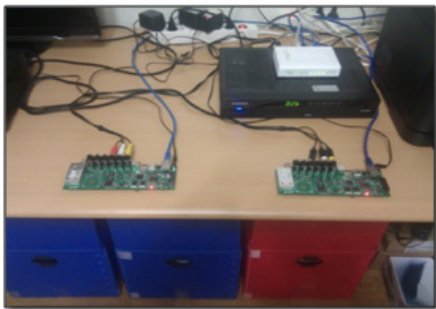
구현한 휴대용 HLS 전송기의 성능을 검증하기 위한 실험 환경은 [Fig. 8]와 같다.



[Fig. 8] The block diagram of testing environment of HLS

3.3 HLS 전송기의 동작평가 결과분석

HLS 전송기의 성능평가는 Cable TV 방송을 수신세 탑박스를 통해 실시간으로 수신하여 AV출력단자를 바로 본 논문에서 개발한 HLS 전송기의 입력신호로 입력하고 HLS 전송기에서 무유선 인터넷망을 통해 영상데이터를 전송하였고 수신단에서는 iPhone, 노트북, 아드에서 동시에 재생하여 HLS 전송기의 성능을 확인하였다. [Fig. 10]은 개발한 HLS 전송기의 테스트환경 화면이다. 케이블 TV 수신 셋탑박스에서 출력을 바로 본 논문에서 개발한 두 대의 HLS의 AV입력으로 연결하여 유무선 인터넷망을 이용하여 영상데이터를 전송하는 실험장면 이다.



[Fig. 9] The operation testing picture of HLS transmitter

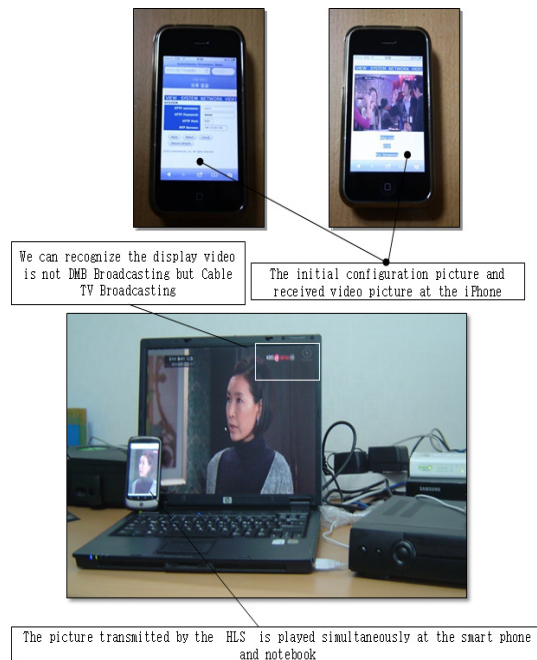
본 논문에서 구현한 HLS 전송기의 성능평가의 기준 규격은 <Table 4>의 Apple사에서 권장하는 HLS 서비스를 위한 최소한의 권장규격을 만족하기를 검증하기 위해 수신단에서 3G망과 WiFi망을 통해 수신된 영상을 분

석하여 확인하였다. 본 논문에서 개발한 HLS의 성능 결과를 <Table 7>에 나타내었다.

<Table 7> The received video analysis of developing mobile HLS transmitter

Network	Resolution	Video rate	Key frame number
3G	480x300	150kbps	40
WiFi	640x480	800kbps	120

<Table 7>에 나타낸 바와 같이 본 논문에서 개발한 HLS 전송기의 성능이 Apple사의 최소 권고안을 충분히 만족하는 결과를 보였다. [Fig. 11]은 [Fig. 9], [Fig. 10]에서 설명한 실험환경에서 개발한 HLS 전송기를 이용하여 CATV방송을 실시간으로 전송하여 iPhone에서 수신 설정값을 표시하는 화면과 iPhone, 노트북, android폰으로 수신한 영상을 표시하는 결과입니다.



[Fig. 10] The received picture of HLS transmitter operation testing result

4. 결론

본 논문에서는 저가격, 저 전력소모 이면서도 휴대 가

능한 HLS 전송기를 구현하기 위해 Http Live Streaming 전송방식에 대해 연구·분석하였고, 실제로 경량화 및 저비용화를 위해 Arm11 프로세서를 사용하여 하드웨어를 개발하였다. 또한 소프트웨어 비용을 최소화 하기 위해 운영체제로는 임베디드 리눅스를 적용하였고 오픈 소스인 FFMpeg와 Open Source Segmenter을 이용하여 HLS 전송기를 구현하였다.

개발한 HLS 전송기의 성능을 검증하기 위해 성능평가 실험환경을 [Fig. 8]같이 구축하였고, CATV에서 수신한 실시간 영상을 개발한 HLS 전송기에 입력하고 HLS전송기가 Http 라이브 비디오 스트림을 유무선 인터넷망으로 전송한 후에 수신단에서 WiFi 무선망과 3G 통신망으로 iPhone, 안드로이드 폰, 노트북을 이용하여 수신한 영상을 재생하여 영상의 동작 및 품질을 평가 및 확인하였다.

즉, 본 논문에서 개발한 HLS 전송기를 사용해 전송한 동영상 수신단에서 분석한 결과, Apple사의 HLS 서비스 권고안을 충분히 만족하는 성능을 얻었다. 그리고 실제 전송기와 다양한 클라이언트에서의 동작 화면을 제시하였다. 본 논문에서 개발한 제품의 활용분야는 실시간으로 현장의 위급상황을 현지 지휘소나 원거리 사령부에 전달해야하는 경우 각 현장참여 인원이 개발한 HLS를 개인이 착용하여 이동하면서 실시간으로 상황영상을 전달하는 분야에 활용될 수 있을 것으로 생각된다.

REFERENCES

- [1] R.Frederick, J. Geagan, M. Kellner, A. Periyannan, 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] Schulzrinne. H., RTP: ATransport Protocol for Real-Time Applications, RFC 1889, 1996.
- [4] Squid Web Proxy Cache, <http://www.squid-cache.org>
- [5] Asit Dan, Daniel M. Dias, Rajat Mukherejee, Dinkar Sitaram and Renu Tewari, "Buffering and Caching in Large-Sacle Video Servers," Proceeding of IEEE CompCon, Los Alamitor, pp.217-224, 2002
- [6] QuickTime Streaming Server <http://www.apple.com/~quicktime/products/qtss>
- [7] <http://helloworld.naver.com/helloworld/HTTPLive-Streaming.html>
- [8]<http://developer.apple.com/library/ios/#documentation/StreamingMediaGuide/Introduction.html>
- [9] carson "iPhone HTTP Streaming with FFMpeg and Open Source Segmenter" 2009.
- [10] carson " iPhone Windowed HTTP Live Streaming Using Amason S3 Cloudfront Proof of Concept, 2009.
- [11] <http://www.solvusoft.com/ko/file-extensions/-file-extension-m3u8/>
- [12] <http://www.gravlab.com/2013/10/26/example-sample-http-live-streaming-m3u8-embedded-player/>
- [13] International Standard ISO/IEC 13818-1 Information Technology - General Coding of Moving Pictures and Associated Audio Information: Systems, 2000.
- [14] Text of ISO/IEC FDIS 23002-3" Information technology- MPEG video technologies - Part3: Representation of auxiliary video and supplemental information" , 2007.
- [15] Text of ISO/IEC 13818-1:2007/FDAM 4 - Transport of Multiview Video over ITU-T Rec H.222.0 | ISO/IEC 1381-1, ISO/IEC/JTC1/SC29 -/WGL1/N10745, London, UK, 2009.
- [16] https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/StreamingMediaGuide/UsingHTTPLiveStreaming/UsingHTTPLiveStreaming.html#//apple_ref/doc/uid/TP40008332-CHI02-SW1
- [17] Jea-Hee, L. Chang-Jin, S. The Study on the video Quality Estimation of IPTV using the European Digital Video Quality Standard Estimate method ETSI TR 101 209. Journal of Digital Convergence v.10, no.6, 197-203, 2012
- [18] Jang-Nak, C. Sang-Hee, L. A Multi-Antenna Mobile Measurement System for DTV Coverage Measurement Journal of Digital Convergence v.11, no.11, 85-94, 2013

이 재 희(Lee, Jea Hee)



- 1985년2월 : 평운대학교전자통신과 (학사)
- 1987년2월 : 평운대학교전자통신과 (석사)
- 2000년2월 : 평운대학교전자통신과 (박사)
- 1999년 ~ 현재 : 동서울대학교 정보통신과 교수

· 관심분야 : 임베디드 시스템
· E-Mail : ljh7314@dsc.ac.kr

조 태 경(Cho, Tae Kyung)



- 1984년 2월 : 한양대학교전자통신 공학과(학사)
- 1986년 2월 : 한양대학교전자통신 공학과(석사)
- 2001년 2월 : 한양대학교전자통신 공학과(박사)
- 2003년 2월 ~ 현재 : 상명대학교 국방정보 공학과 교수

· 관심분야 : 멀티미디어통신
· E-Mail : tkcho@smu.ac.kr