ISSN 1598-0170 (Print)
ISSN 2287-1136 (Onine)
http://www.jksii.or.kr

# 동적 단일 버퍼링 기법을 적용한 스마트 HLS의 채널변경 분석☆

# Channel Transition Analysis of Smart HLS with Dynamic Single Buffering Scheme

김 종 일[1*] 강 민 구[2] 김 동 현[2] 김 인 기[3] 한 경 식[3]
Chong-il Kim Min-goo Kang Dong-hyun Kim In-ki Kim Kyung-sik Han

## 요 약

본 논문에서는 OTT(Over The Top)가 채널상태에 따라 동적으로 최적화함으로서, 최고의 비디오 비트율(bit-rate)로 전송하기 위한 스마트 HLS(HTTP Live Stream) 플랫폼을 제안하였다. HLS 플랫폼은 스마트 OTT가 채널변경 시간을 최적화도록 HLS 서버와 OTT 클라이언트 사이의 대역폭을 모니터링 한다. 이를 위해 다중 비트율과 적정한 대역폭으로 비디오 스트림을 조정하도록 설계되었다. 제안한 스마트 OTT는 최적의 비트율로 버퍼링을 함으로서 라이브와 VOD(Video On Demand)비디오를 재생할 수 있다.

이러한 HLS 플랫폼은 세그먼트 형식의 H.265 MPEG-2 TS(Transport Stream) 비디오와 관련 정보를 담고 있는 m3u8 파일과 스마트 OTT가 연동한다. 이로서 설계한 단일 버퍼링 기반의 스마트 OTT는 동작 중인 채널 대역폭 효율과 복호화된 VOD에 대한 적응적 비트율을 활용함으로서 대역폭이 허용하는 최대의 데이터를 버퍼링으로 최적의 비디오를 전송할 수 있다

☞ 주제어 : HLS(HTTP Live Stream), 단일 버퍼링, 비디오 비트율, 대역폭 조절기, 다중 세션

## ABSTRACT

In this paper, we propose a smart HLS(HTTP Live Stream) platform with dynamic single buffering for the best transmission of adaptive video bit-rates. This smart HLS can optimizes the channel transition zapping-time with the monitoring of bandwidth between HLS server and OTT(Over The Top) client. This platform is designed through the control of video stream due to proper multi-bitrates and bandwidths. This proposed OTT can decode the live and VOD(Video On Demand) videos with the buffering of optimumal bitrate. And, the HLS can be cooperated with a smart OTT, and segmented for the m3u8 files of H.265 MPEG-2 TS(Transport Stream) videos.

As a resullt, this single buffer based smart OTT can transmit optimal videos with the maximum data buffering according to the adaptive bit-rate depending on the network bandwidth efficiency and the decoded VOD video, too.

☞ keyword : HLS(HTTP Live Stream), single buffering, video bitrate, bandwidth resolution, multi-session

## 1. Introduction

IP broadcasts are separated by IPTV services using the managed network and OTT (Over The Top) services using an unmanaged network. Adaptive buffering technique is very important to reduce the channel disconnection and improve channel zapping speed at OTT client using unmanaged network which is not guaranteed QoS(Quality of Service) [1].

[1] Catholic Kwandong University, Korea.
[2] Hanshin University, Korea.
[3] Quber Co. TD., Korea.
[*] Corresponding author (cikim@cku.ac.kr)
[Received 10 May 2016, Reviewed 26 May 2016, Accepted 19 September 2016]
☆ A preliminary version of this paper was presented at ICONI 2015 and was selected as an outstanding paper.

Streaming transmission of OTT is HTTP(Hyper Text Transfer Protocol), RTP(Real-time Transport Protocol), RTSP(Real Time Streaming Protocol). HTTP transmission is used for the OTT services, because HTTP ensures the transfer of all data packets that are sent to destination [2].

Media transport protocol based on the HTTP transmission scheme has the MPEG (Moving Picture Experts Group) -DASH (Dynamic Adaptive Streaming over HTTP), and HLS (HTTP Live Streaming) with dynamic single buffering[3].

HLS streaming is designed to transmit audio and video based on general web server to iOS based on mobile devices, android-based mobile devices with adaptive video bit rates [4].

In this paper, we present dynamic single buffering to shorten the zapping time of channels, as a result of optimizing buffering algorithms based on adaptive bandwidth.


http://dx.doi.org/10.7472/jksii.2016.17.6.09

## 2. Analysis of HTTP Live Streaming

HLS standard in 2009 is made to solve the problem of delivering video to mobile devices over varying bandwidths using adaptive streaming with adaptive bit-rate video delivery for the dynamic bit-rate optimization of stream buffering, and for the channel surfing based on smart OTT [5].

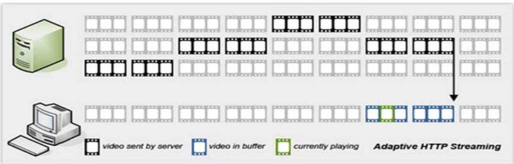


Fig. 1. Adaptive HTTP streaming with dynamic bitrate

### 2.1 Overlaying hybrid IPTV and smart OTT

For reducing the zapping time of HLS for smart OTT, dynamic bitrate optimization can still be challenging, if many different encoding profiles are required. But, even for a smaller standalone OTT deployment, flexibility is required in the encoding solution of HLS-server and OTT-client [6].

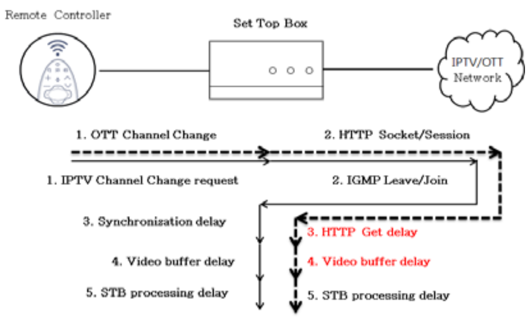


Fig. 2. Design of smart OTT with channel-changing

### 2.2 Zapping time analysis of smart OTT

In traditional TV broadcast and cable technology, zapping(channel-changing) time is instantaneous since it only involves the TV receiver tuning to a specific carrier frequency, demodulating the content, and displaying it on the TV screen with smart OTT.

When a user switches to a new channel, the STB(Set Top Box) has to issue a new channel request towards the network. Since video distribution is done via multicasting, this is translated into leave and join multicast requests. These are handled by a group management protocol IGMP (Internet Group Management Protocol) between HLS and OTT [7].

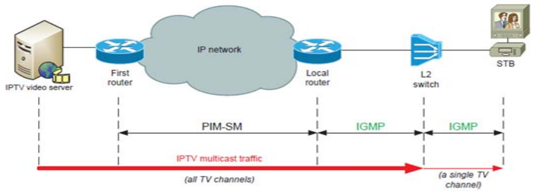


Fig. 3. IGMP analysis for HLS server/smart OTT

Many video services also employ content encryption, so the encryption keys must be acquired and provided to the decryption for decrypting the content, and this also adds to synchronization delay during transmission.

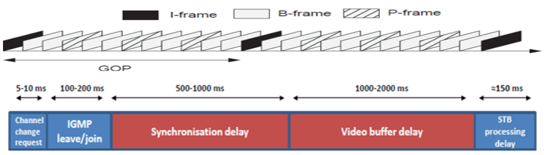


Fig. 4. Design of zapping(channel-changing) time

A key problem of HLS is that it is restricted to one DRM(Digital Right Management), and so is not likely to gain wider adoption in the OTT beyond Apple iOS. The following [table.1] summaries the main differences between the various ABR(Adaptive Bit Rate) solution[8].

Table 1. Analysis of difference among ABR solutions

| Feature | Apple | Microsoft | Adobe | MPEG–DASH |
|---|---|---|---|---|
| Adaptive streaming technology | HLS | Smooth Streaming | HDS | MPEG–DASH |
| Codec used | H.264 | H.264/VC-1 | H.264, VP6 | H.264/AVC or other MPEG codec family (SVC, MVC, HEVC) |
| Open standard | No | No | No | Yes |
| Adopted by industry consortium | No | No | No | HbbTV, YouView published DASH based standards, 3GPP , DECE and DLNA work on a DASH based standard |
| Subtitle support | Partial | Yes | Partial | Yes |
| Multiple audio support | V4 only * | Yes | No | Yes |
| Interop testing | No | No | No | Yes |
| Trick mode support | Partial | Yes | Partial | Yes |
| CDN friendly | Requires chunk carriage optimization | Requires specific IIS-7 server | Requires specific FMS server | Yes |
| Device support | IOS, Mac, Xbox, PlayStation, STB, TV Android. | PC, Xbox, STB, TV | PC, TV | Limited in 2012. TV, Tablet, phone in 2013. |

## 2.3 Dynamic Bit-rate of smart OTT

In order to analyze the channel-changing time, we define the zapping time which is related to channel transition between STB tuner operation(OTT) and stream server-HLS streaming of video presentation. In Fig. 5, HLS based adaptive streaming mode is designed for the bit-rate optimization of smart OTT with media segments of m3u8 files of H.265 MPEG-2 TS [8].

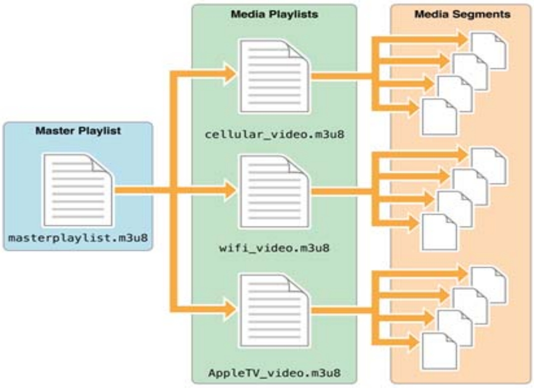


Fig. 5 Design of HLS based adaptive streaming

For the dynamic bit-rate optimization, the analysis of zapping time is proposed with proxy in Fig. 6. There are three other reasons to buffer incoming of packets in the decoder of smart OTT like in Fig. 7 [8][9][10].

In Fig. 8, HTTP traffic is analyzed in Galaxy S4 with LTE-A using YouTube APP. During watching live video, contents are requested, and delivered with HLS and OTT, H.264 CODEC, and 5 second-chunk duration.

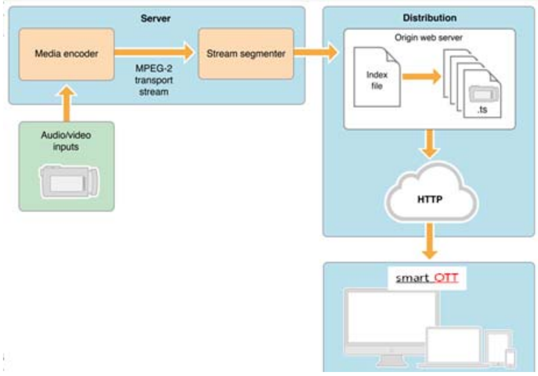


Fig. 6. Zapping analysis of smart OTT with proxy

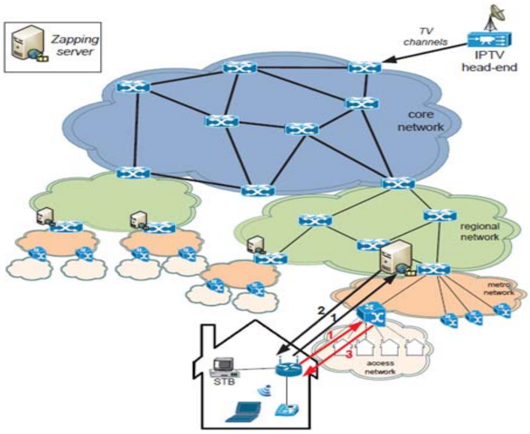


Fig. 7. OTT decoding process of HTTP ABR streaming

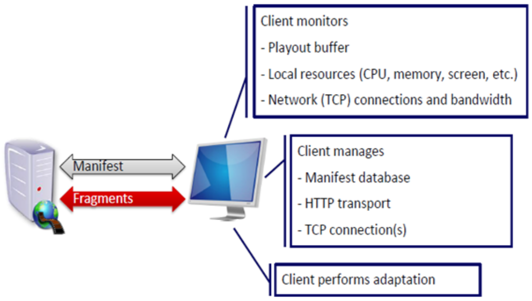


Fig. 8. HTTP traffic analysis diagram with HLS/OTT

In Fig. 9, HTTP adaptive bit rate is analyzed from 64 Kbps(itag 151) to 2.8Mbps (itag 95), and 758Kbps (itag 93), 1.47Mbps (itag94) depending on the chunk quality of LTE-A and Wi-Fi for YouTube Live TV [12].

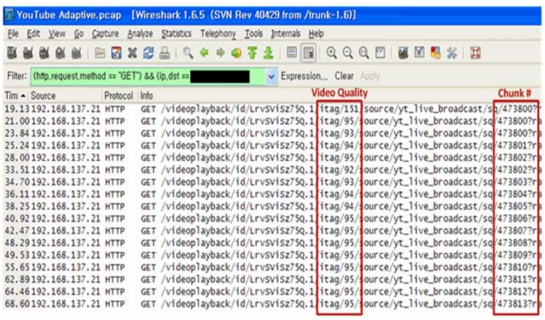


Fig. 9. HTTP analysis of adaptive bit rate videoin Galaxy S4 with LTE-A using YouTube APP

## 3. Design of Dynamic Single Buffering

Single buffering is designed for playing media without disconnection. One HTTP session is connected with HLS server, but this paper presents multiple session with allowed network bandwidth for seamless video transition without delay like Fig. 10-12.

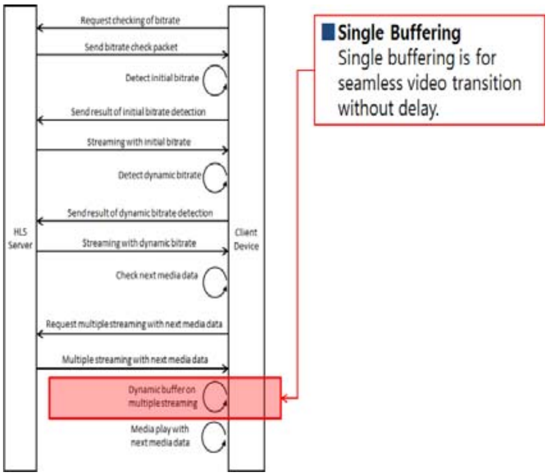


Fig. 10. Design of dynamic single buffering for seamless video transition without delay

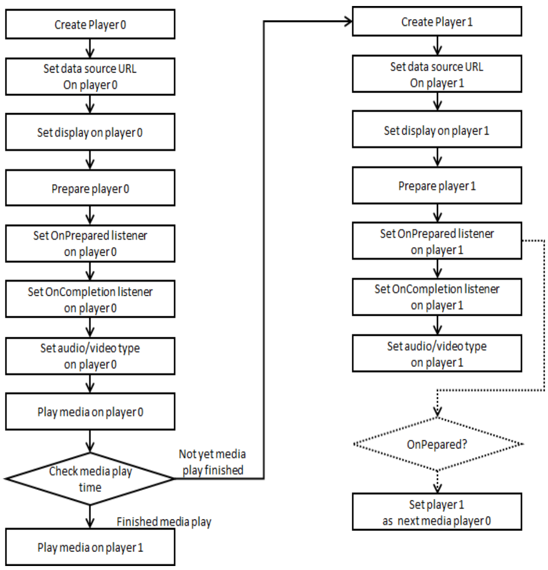


Fig. 11. Flowchart design of dynamic single buffering

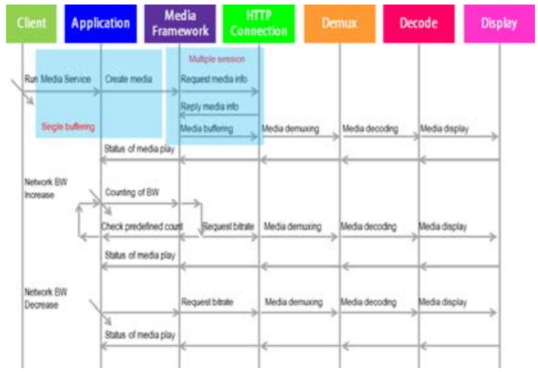


Fig. 12. Protocol of dynamic single buffering

## 4. Simulation & Result Anaysis

In Fig. 13, HLS server and WiFi access point are connected with smart OTT device including OTT application like Table 2 for the analysis of zapping time based on an OTT client from smart HLS [13-15].

Table. 2. Spec. of simulation environment

| Device | Specification | |
|---|---|---|
| Smart OTT/STB | Hisilicon Hi3719M Dual Core CPU based Set Top Box<br>RAM 1Gbyte, Ethernet 10/100-T | |
| Access Pointer | Kaon AR3010 802.11 a/b/g/n/ac based AP<br>Support QoS | |
| HLS Server | Wowza Media Server 3.6.2<br>HLS, RTP, VOD Streaming | |
| Channel Information | Channel #1 | Animation, Running Time : 2m20s<br>Bitrate: 3M, 6M, 10M, 15M, 20Mbps |
| | Channel #2 | Animation, Running Time : 3m27s<br>Bitrate: 3M, 6M, 10M, 15M, 20Mbps |
| | Channel #3 | Flight Video, Running Time: 4m18s<br>Bitrate: 3M, 6M, 10M, 15M, 20Mbps |

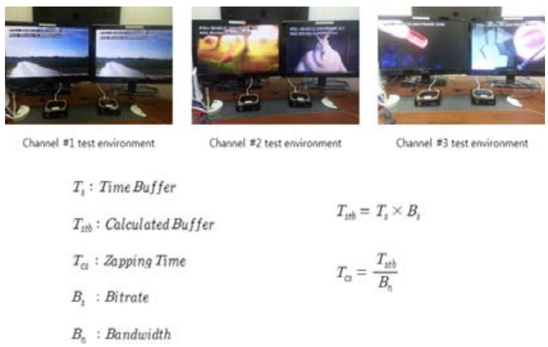


Fig. 13. Experimental analysis of zapping time

Table. 3. Zapping analysis of channel transition

| | channel #1 to #2 | | channel #2 to #3 | | channel #3 to #1 | | Ideal Value |
|---|---|---|---|---|---|---|---|
| AP bandwidth | single | adaptive | single | adaptive | single | adaptive | Ideal Time |
| 1000 | 23.4 | 0.69 | 34.51 | 0.71 | 40.28 | 0.59 | 30 |
| 2000 | 7.45 | 0.69 | 14.93 | 0.73 | 19.42 | 0.74 | 15 |
| 5000 | 3.85 | 0.69 | 4.71 | 0.69 | 7.17 | 0.67 | 6 |
| 8000 | 3.26 | 0.69 | 3.82 | 0.72 | 5.08 | 0.72 | 4 |
| 10000 | 2.66 | 0.65 | 3.95 | 0.63 | 3.68 | 0.67 | 3 |
| 15000 | 1.7 | 0.65 | 2.6 | 0.65 | 2.35 | 0.68 | 2 |
| 20000 | 2.58 | 0.57 | 2.09 | 0.68 | 2.53 | 0.71 | 2 |
| 25000 | 2.1 | 0.56 | 2.33 | 0.58 | 1.86 | 0.62 | 1 |
| 30000 | 1.27 | 0.52 | 1.26 | 0.58 | 6.34 | 0.65 | 1 |
| | channel #1 to #2 | | channel #2 to #3 | | channel #3 to #1 | | Ideal Value |
| AP bandwidth | single | adaptive | single | adaptive | single | adaptive | Ideal Time |
| 35000 | 1.21 | 0.56 | 1.51 | 0.58 | 1.39 | 0.68 | 1 |
| 40000 | 1.29 | 0.56 | 1.25 | 0.64 | 1.58 | 0.72 | 1 |
| 45000 | 1.58 | 0.65 | 1.34 | 0.7 | 1.35 | 0.66 | 1 |
| 50000 | 1.47 | 0.47 | 1.29 | 0.53 | 1.37 | 0.64 | 1 |
| 55000 | 1.38 | 0.64 | 1.2 | 0.57 | 1.38 | 0.64 | 1 |
| 60000 | 1.08 | 0.52 | 1.15 | 0.65 | 1.26 | 0.48 | 1 |
| 65000 | 1.16 | 0.56 | 1.03 | 0.63 | 1.51 | 0.65 | 0 |
| 70000 | 1.34 | 0.69 | 1.25 | 0.74 | 1.2 | 0.65 | 0 |
| 80000 | 1.24 | 0.59 | 1.13 | 0.65 | 1.28 | 0.61 | 0 |
| 90000 | 1.09 | 0.56 | 1.16 | 0.57 | 1.28 | 0.79 | 0 |
| 100000 | 1.12 | 0.56 | 1.08 | 0.67 | 1.19 | 0.54 | 0 |

In Fig. 13, zapping time(channel #1 to #2, channel #2 to #3, and channel #3 to #1) is analyzed by changing of network bandwidth which is from 1Mbps to 100Mbps with fixed video rate(6Mbps) from Tcs.

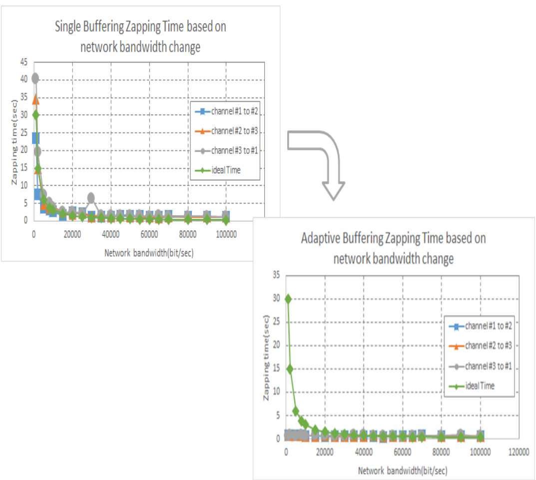


Fig.14. Zapping time analysis for single buffering

## 4. Conclusions

In this paper, channel zapping time is significantly reduced with dynamic single buffering according to network bandwidth and video bit-rate with canning through different television channels to find something interesting to watch, and the number of buffering is reduced with media transfer to others in seamless video transition without delays.

As a result, this smart HLS platform with OTT will be useful in 4K and 8K systems which have high quality of video resolution with dynamic buffering depending on the channel condition for the wide availability of the remote control for the future.

## ACKNOWLEDGMENT

This work was supported by a research grant (Advanced Technology Center[#10045816] of MOTIE/KEIT, the Korean government)

## Reference

[1] In-ki Kim et al, “Analysis of Smart OTT based Zapping Time with Adaptive Buffering for Live Streaming,” ICONI 2015, Dec. 2015. http://www.iconi.org/2015

[2] I.K. Kim, M.G.Kang etal, “Method of processing OTT data based on dynamic adaptive buffering,” Korea Patent (10-2016-0026673), http://www.kipris.or.kr

[3] G.J. Yang, B.W. Choi, and J.H. Kim, “Implementation of HTTP Live Streaming for an IP Camera using an Open Source Multimedia Converter,” International journal of software engineering and its applications, vol. 8, no.6, pp. 39 - 50, 2014. http://dx.doi.org/10.14257/ijseia.2014.8.6.04

[4] How-Sing Lin, “Improving the Availability of Scalable on-demand Streams by Dynamic Buffering on P2P Networks,” KSII transactions on internet and information systems, vol. 4, no. 4, Aug. 2010. http://dx.doi.org/10.3837/tiis.2010.08.003

[5] Hyoung-Gook Kim, “Enhanced Timing Recovery Using Active Jitter Estimation for Voice-Over IP Networks,”

KSII transactions on internet and information systems, vol. 6, no. 4, Apr. 2012.
http://dx.doi.org/0.3837/tiis.2012.04.003

[6] M. K. Jung, "Design of streaming system with live streaming load balance and VOD media file distribution from fixed nodes based on overlay network," Kongju University, Feb. 2015.
http://dlps.nanet.go.kr/SearchDetailList.do

[7] C. M Sung, "QoE-aware DASH Agent for Video Streaming Service in Mobile Cloud Environments," Kyungbuk University, Feb. 2016.
http://www.riss.kr/search/detail/DetailView.do?p_mat_type=be54d9b8bc7cdb09&control_no=097fb38ce507db68ffe0bdc3ef48d419

[8] T. H Ban, "(A)study on adaptive dual-streaming system," Baejae University, Feb. 2016.

[9] Hyun-Sik Kim et al, "An Adaptive Buffering Method for Practical HTTP Live Streaming on Smart OTT STBs," TIIS, Vol. 10, No.3, March 31, 2016.
http://dx.doi.org/10.3837/tiis.2016.03.028

[10] http://www.encoding.com/httplive-streaming-hls, 2015

[11] http://www.avacoustics.in/video/video-streaming, 2016

[12] https://www.hlsplayer.net/

[13] http://www.kaonmedia.com/

[14] http://www.uniqcast.com/blog/blog-article.php?id=21

[15] http://www.squadeo.tv/quickplayer-for-web-white-paper/

## ◐ Authors ◑

**Chongil Kim**
received the B.S. degree in electronic engineering from Sogang University, Seoul, Korea, in 1987, and the M.S. and Ph.D. degrees in electronic engineering from Yonsei University, Seoul, Korea, in 1989 and 1993 respectively. From 1993, he was the assistant and associated Professor of the Department of Electronic Engineering, Catholic Kwandong University. He is currently Professor of the Department of Electronic Engineering, Catholic Kwandong University. His current research interests are in applications of coding and modulation to modern digital communication systems, specifically channel coding, trellis-coded modulation, multicarrier communication and CDMA systems for digital mobile communications.

**Mingoo Kang**
is a professor in the department of IT contents at Hanshin University, Osan South Korea from 2000. He has received the B.S., M.S., and Ph.D. degrees from Yonsei University, Seoul, Korea all in Electronic Engineering in 1986, 1989 and 1994, respectively. He was a research engineer at Samsung Electronics from 1985 to 1997. His research interests include wireless communication algorithm, mobile devices, and Smart UX for Mobile TV & DTV. He also served the ICONI 2014 hosted by the Korean Society of Internet Information, as a conference chair.

**Donghyun Kim**
is a candidate for master course from Hanshin University, Osan South Korea in Telecommunication Engineering. His research interests include embedded platform, mobile devices, and Android Application.

**Inki Kim**
is a candidate for Ph.D course from Hanshin University, Osan South Korea in Telecommunication Engineering. He was a research engineer at Samsung Electronics and CTO in KaonMedia and CEO in InnoDigital. His research interests broadcasting including OTT, smart eco-system, mobile device and service application and UI/UX for STB & mobile devices.

**Kyungsik Han**
is a Senior Research Engineer at QUBER, Pangyo South Korea from 2010. He has received the Ph.D. degrees from Hanshin University, Osan South Korea in Information Science & Telecommunications in 2013. He has received the B.S. degrees from Soongsil University, Seoul, Korea in Electronic-Science Engineering in 2004. His research interests include OTT, wired & wireless networking for STB and network streaming system architecture.