Received June 6, 2017, accepted July 25, 2017, date of publication August 9, 2017, date of current version September 19, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2737557

Performance Evaluation of the Emerging Media-Transport Technologies for the Next-Generation Digital Broadcasting Systems

KYUNGMO PARK[1], (Member, IEEE), NAMGI KIM[2], AND BYOUNG-DAI LEE[2]
[1]Standards and Technology Enabling Team, Digital Media and Communications Research and Development Center, Samsung Electronics Co., Ltd., Seoul 06765, South Korea
[2]Department of Computer Science, Kyonggi University, Suwon 16227, South Korea

Corresponding author: Byoung-Dai Lee (blee@kgu.ac.kr)

This work was supported in part by the Basic Science Research Program through the National Research Foundation of Korea, funded by the Ministry of Education under Grant NRF-2016R1D1A1B03932696 and Grant NRF-2017R1D1A1B04027874 and in part by the Ministry of Science, ICT, and Future Planning under Grant 2014R1A2A2A0100694.

ABSTRACT Moving pictures experts group (MPEG) media transport (MMT) and MPEG-dynamic adaptive streaming over hypertext transfer protocol (MPEG-DASH) are emerging as the content-delivery technologies for the next-generation broadcasting systems, as they seamlessly utilize various underlying delivery networks to provide hybrid multimedia services in any location. In this paper, we present an empirical analysis of the channel-zapping time, which is an important metric in the measurement of the quality of experience (QoE) of broadcast services, regarding the MMT and MPEG-DASH technologies. During this process, we also clarify the important factors that determine the channel-zapping time of the individual technologies. In addition, we propose a simple yet effective method that enables fast channel zapping for the MMT technology.

INDEX TERMS Channel zapping time, MMT, MPEG-DASH, QoE.

I. INTRODUCTION

Today's digital-TV broadcasting system was developed in the 1990s and has been followed by a significant changing of the environment surrounding the content delivery. The changes include the growing diversity of the delivery channels for media content, the emergence of Ultra-High-Definition (UHD) videos, and the widespread use of personal multimedia devices such as smartphones and tablets, to name only a few. In particular, as Internet-protocol (IP) networks have become ubiquitous, the coordination of the hybrid delivery of media content over heterogeneous networks is essential to enable the new services and distribution models of the next-generation broadcasting systems. Overall, these changes present new opportunities for broadcasters, but at the same time, they pose many technical challenges.

However, the media transport and delivery technologies that are used in today's digital-broadcasting systems are limited in their capability to accommodate the previously mentioned changes. For instance, the small and fixed packet size of the Moving Pictures Experts Group (MPEG)-2 Transport Stream (TS) [1], which is the de facto standard for the container format of the broadcast programs, is not efficient for the delivery of UHD videos that require a high bitrate. With UHD videos, the maximum bitrate that needs to be supported is now almost 100 Mb/s, and for these bitrates, the 188-B fixed-size TS packet is simply not suitable. Additionally, since the MPEG-TS has been designed for legacy digital-broadcasting networks, which are not IP-based, it is very challenging to use the MPEG-2 TS for IP-network content delivery.

Recognizing the new technical challenges and the shortcomings of the existing solutions, the MPEG has developed the following new content-delivery standards for the next-generation broadcasting systems that take into account the emerging convergence of the digital broadcasting networks and the Internet: MPEG Media Transport (MMT) [2] and MPEG-Dynamic Adaptive Streaming over the Hypertext Transfer Protocol (MPEG-DASH) [3]. MMT is the MPEG's latest technology for the transport and delivery of multimedia content over heterogeneous packet-switched networks including bidirectional IP networks and unidirectional broadcast networks. Its aim is the serving of the next-generation multimedia services with new features such as UHD-video content, multiple audio tracks, and multidevice presentation. MPEG-DASH, however, is an adaptive-bitrate-streaming technology that enables high-quality media-content streaming through the use of the Hypertext Transfer Protocol (HTTP).

Due to their flexibility and extensibility, MMT and MPEG-DASH have been adopted as the service-delivery protocols for the next-generation broadcasting systems, as well as for the Internet live-streaming services [4]–[12]. However, to the authors' best knowledge, none of the existing works present indepth discussions of the MMT and MPEG-DASH performances from the Quality of Experience (QoE) perspective when they are used for broadcasting services. Several aspects such as the video and audio quality and the artifacts that are caused by packet loss are used to measure the QoE, but the channel-zapping time is also considered as a key metric for the validation of the QoE of broadcast services [13], [14]. The channel-zapping time refers to the time delay between the moment when the user selects a multimedia content (e.g., TV service) and the moment when the content is rendered on the terminal [15]. Therefore, it is generally the sum of several of the delays that occur throughout the channel-zapping operation such as the tuning, physical-layer forward-error-correction (PHY FEC) decoding, dejittering, service-information acquisition, Stream Access Point (SAP) [3] delay, audio/video decoding and rendering, and conditional access (CA)/Digital Rights Management (DRM) descrambling. In this paper, an empirical analysis of the channel-zapping performances of the MMT and the MPEG-DASH technologies is presented, wherein the important factors that determine the channel-zapping time of each technology are clarified. Also, a simple yet effective method that enables fast channel zapping for the MMT standard is proposed. It should be noted that the focus of the present paper is the reduction of the delays that are directly related to the protocol operations.

The organization of this paper is as follows. Section II is a brief introduction of the MMT and MPEG-DASH standards, and Section III presents the channel-zapping processes of these two technologies. Section IV describes the experiment results using prototypic implementations. Lastly, Section V provides a summary and a proposal for a future work.

II. OVERVIEW OF MMT AND MPEG-DASH

A. MMT

MMT provides state-of-the-art technologies in the following three functional areas for an efficient and effective delivery and consumption of the multimedia contents of various formats in heterogeneous network environments (see Fig. 1): *encapsulation*, *delivery*, and *signaling*. The encapsulation functional area defines new logical-data structures for the aggregation of various types of data for the delivery preparation. An *MMT Package* is a collection of *MMT Assets* and the associated metadata such as the compositional information and the transport-characteristics information. An MMT Asset includes media data and is composed of one or more *Media Processing Units* (MPUs). The MPU is an ISO Base Media File Format (ISOBMFF)-based generic container for

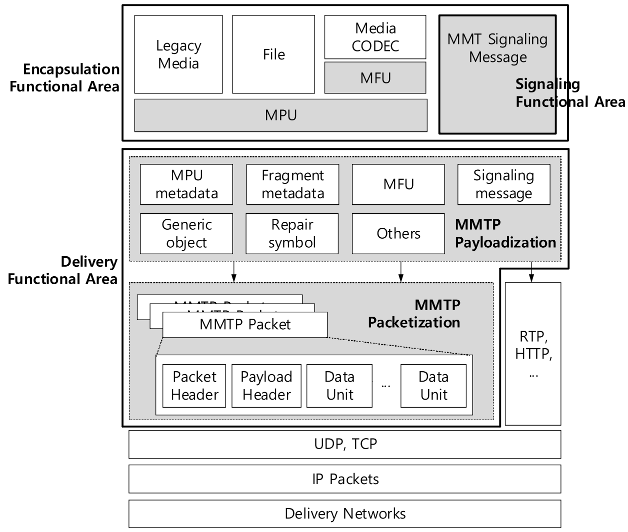


FIGURE 1. **MPEG-media-transport (MMT) protocol stack (gray boxes are the MMT scope).**

timed or nontimed media data that is independently decodable and for which the referencing of other MPUs is not required. For example, one or more of the complete group of pictures (GOP) of a video clip can be encapsulated in a single MPU [16]. Each MPU constitutes a nonoverlapping piece of an asset. MPUs provide information on the internal structure of media data to support the media-aware fragmentation of an MPU for transportation purposes. This small MPU fragment is called the *Media Fragment Unit* (MFU), and it enables an adaptive packetization of the MPUs according to the constraints of the underlying delivery networks, e.g., the maximum size of the transmission units.

The delivery functional area defines an application-layer transport protocol, named the *MMT Protocol* (MMTP), and the payload format for an efficient and reliable packetized delivery of the MMT Packages. The MMTP standard provides enhanced features for the delivery of packages such as protocol-level multiplexing that enables various assets to be delivered over a single MMTP-packet flow, and a delivery-timing model that is independent of the presentation time to enable an adaptability to a wide range of network jitters [2]. The MMTP-based delivery of the MPUs requires the occurrence of packetization and depacketization procedures at the MMT sending entity and the MMT receiving entity, respectively. The MMTP-packetization procedure transforms an MPU into a set of MMTP payloads that are then carried in the MMTP packets, while the depacketization procedure recovers the original MPU data. Note that the MMT defines two packetization/depacketization modes. In the MPU-format agnostic mode, an MPU is packetized into data units of an equal size or a predefined size according to the size of the MPU of the underlying delivery network. In the MPU-format aware mode, however, the boundaries of the different types of data in the MPU are considered in the generation of the packets for the packetization procedure.

For instance, in the case of an MPU of timed media such as video and audio, the MMT packets carry the delivery-data units of either MPU metadata, fragment metadata, or MFUs (see Fig. 2). Therefore, the MPU-format aware packetization enables the underlying network layers or the intermediate-network entities to identify the nature and the priority of the data units that are to be carried, and to perform various Quality-of-Service (QoS) functions such as the discarding of the less-important MMT packets during a network congestion.

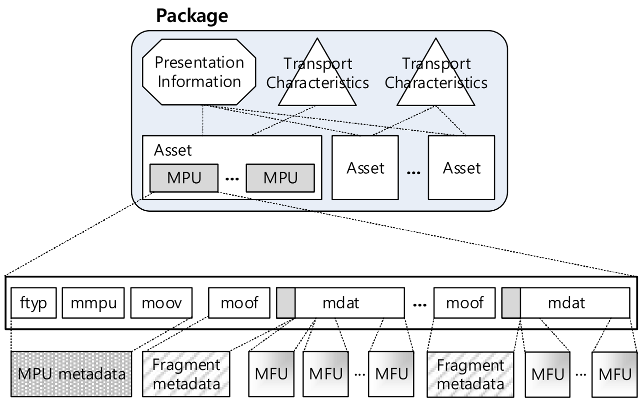


FIGURE 2. **Payload generation for timed media.**

MMT signaling messages provide the MMT clients with the necessary information for the delivery and consumption of the MMT packages, including the structural relationships among the MFU, MPU, asset, and package, which are the configuration information of the MMTP- and MMT-payload formats. The signaling messages are carried by the MMT payload, and the MMTP supports the multiplexing of the media data and the signaling messages within a single MMT-packet flow.

B. MPEG-DASH

Since its release in 2012, MPEG-DASH has been widely deployed in commercialized products ranging from connected TVs to smartphones. MPEG-DASH defines the following two basic formats: *Media Presentation Description* (MPD) and *Segments*. The MPD provides a manifest of the available content, program timing, media characteristics such as the video resolution and the bitrates, the media-component locations on the network, the existence of various encoded alternatives, and other content characteristics, while Segments can contain any type of media data in the form of chunks. Figure 3 shows the hierarchical structure of the MPD. The MPD consists of one or more *Periods*, each of which represents a program interval along the temporal axis. Each period has a starting time and a duration, and they contain one or more *Adaptation Sets*. An Adaptation Set provides the information about a set of the interchangeable encoded versions of one or several media-content components using *Representations*. Therefore, the media contents that are described by Representations within the

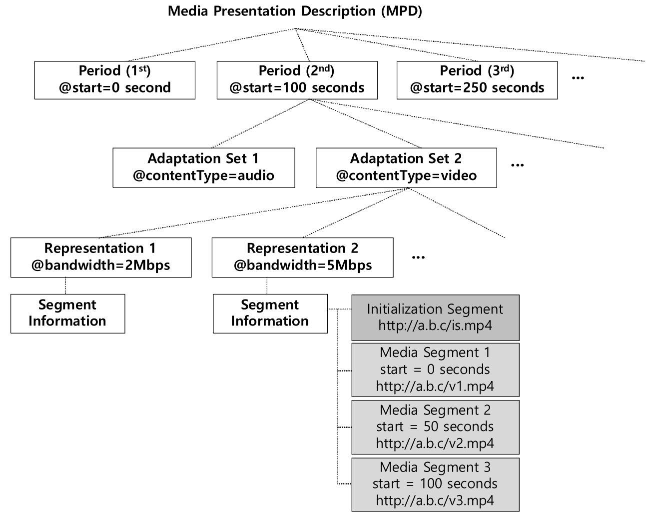


FIGURE 3. **Media-presentation-description (MPD) data model.**

same Adaptation Set are different from one another according to the bitrates, resolutions, and other characteristics. At most, a Representation consists of one *Initialization Segment* for the initialization of the DASH client's media decoder, and one or more *Media Segments* for the delivery of the actual media data. Each Segment comprises a *Uniform Resource Identifier* (URI), so it is the largest unit of data that can be retrieved using a single HTTP request.

To play the media content, the DASH client must first obtain the MPD. Using the information in the MPD, the DASH client then selects the appropriate encoded alternative based on the current network condition and starts its stream by fetching the corresponding segments. While the content is being played, the client automatically selects the next segment from the alternatives for the downloading and playback procedures, thereby facilitating a seamless adaptation to the network conditions or other factors.

MPEG-DASH can be deployed over non-HTTP/TCP networks without a modification of the DASH clients. For instance, in unidirectional network environments such as conventional digital broadcasting systems, the *File Delivery over Unidirectional Transport* (FLUTE) protocol [17] can replace HTTP as the underlying delivery protocol. Then, the Segments are delivered as FLUTE objects in such a way that an HTTP-*Uniform Resource Locator* (URL) is assigned to each delivered object through the *File Delivery Table* (FDT), and the HTTP-URL maps the Segment URLs in the MPD [4].

III. CHANNEL ZAPPING PROCESS

Hybrid broadcasting services allow the operation of digital broadcasting services over dedicated broadcast networks and the seamless integration and convergence of IP-based multimedia services over the Internet. The support of hybrid broadcasting services has become an important requirement of the next-generation broadcasting systems. For hybrid broadcasting services, the channel-zapping time, which is the delay

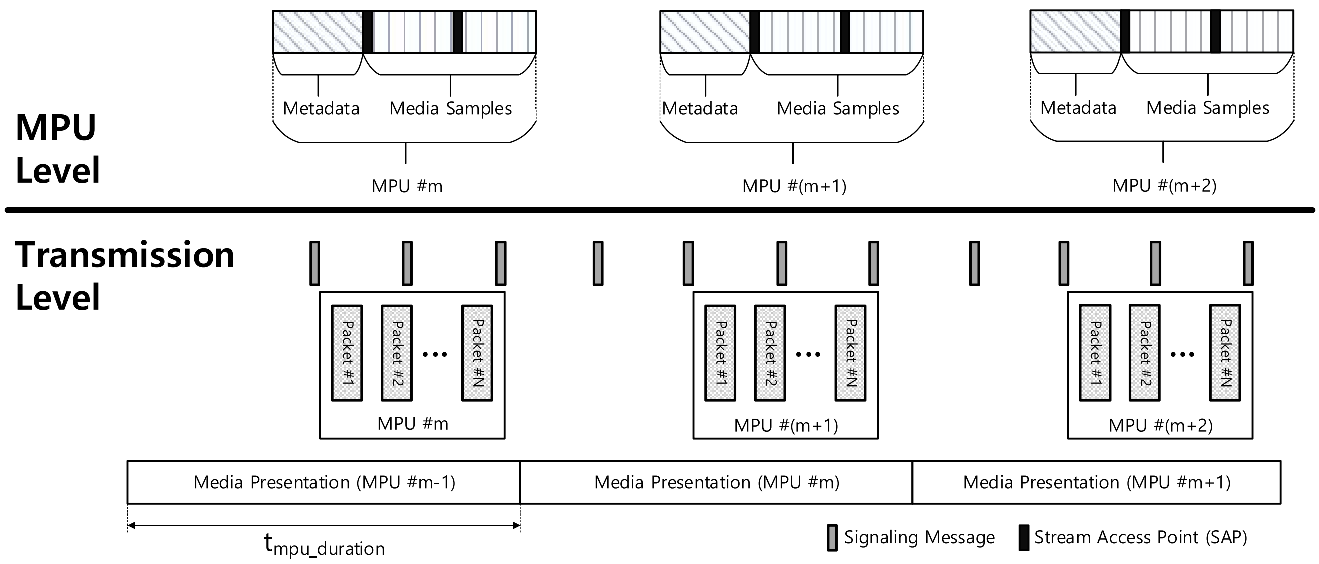


FIGURE 4. **Transmission of the MPEG-media-transport (MMT) services.**

that occurs when the user switches between TV channels, should not fluctuate significantly due to the delays in the heterogeneous underlying delivery networks; therefore, it is an important QoE metric of the next-generation broadcasting systems. In the next section, the channel-zapping processes of the MMT and MPEG-DASH standards are compared in the broadband-network environment.

A. MMT

To play a media content, an MMT client must first obtain the necessary signaling messages. By using the information in the signaling message, the MMT client acquires the location information for the MMTP-packet flows that are delivering the assets comprising the selected package (e.g., a TV program) and starts receiving the corresponding MPUs. An MPU carries not only the audiovisual (AV) data but also the metadata for the initializing of the MMT receiving entities, such as the MPU sequence number, decoder-configuration information, and presentation timestamp. Therefore, the acquisition of the service and initialization information from the signaling messages and the first-received MPU is the first mandatory operation that needs to be performed by an MMT client to play back the selected media content.

Figure 4 shows an example of an MMT service. The first access unit of an MPU must be a SAP, and multiple SAPs can be contained in the MPU so that the media samples can be decoded independently without the receipt of a complete set of MPU samples. For the transmission, the MMTP payloads containing the signaling messages or the MPUs are eventually packetized into an appropriate format according to the underlying transport protocol (e.g., UDP). For the sake of simplicity, it is assumed in this study that the number of packets that is generated for the same MMTP-payload type is the same. For instance, an MPU is packetized into N packets in Fig. 4. Once the packets for an MPU are received and reassembled, the MMT client must wait until it is the presentation time of the received MPU before it can reproduce the MPU contents.

As explained previously, the first operation of channel zapping is the receipt of a signaling message. Unlike on-demand AV streaming services, where the clients pull the signaling messages, MMT is based on the push model, and therefore, the broadcasting systems must transmit the signaling messages periodically, and the time interval between the consecutive signaling messages must be frequent enough to meet the tradeoff between the marginal channel-zapping time and the bandwidth consumption. Note that the number of MPUs between two consecutive signaling messages can be more than one; however, it is clear that the longer the signaling-message interval is, the longer the channel-zapping time is, and only the case where the channel-zapping time can be the shortest was considered for this study, as shown in Fig. 4. In addition, a time interval was also set between two consecutive MPU transmissions to ensure the synchronization of different MMT clients that are accessing the media content at different time points, and in theory, the time interval needs to be set as equal to that of the previous MPU presentation time [18], which is denoted by $t_{mpu\_duration}$ in Fig. 4.

Figure 5 shows the channel-zapping times of the MMT services for two different access points. As the metadata of an MPU is typically sent in the early part of the MPU, as shown in Fig. 2, it is supposed here that the MPU metadata are carried in the first packet. In Case 1, upon the switching to a service, the MMT client was able to receive the service information and the mth MPU without additional delays, and therefore, the channel-zapping time (e.g., $t_{mmt\_zapping}$) was affected by only the MPU-receipt time (e.g., t_{mpu}) and the elapsed time from the decoding to the rendering of

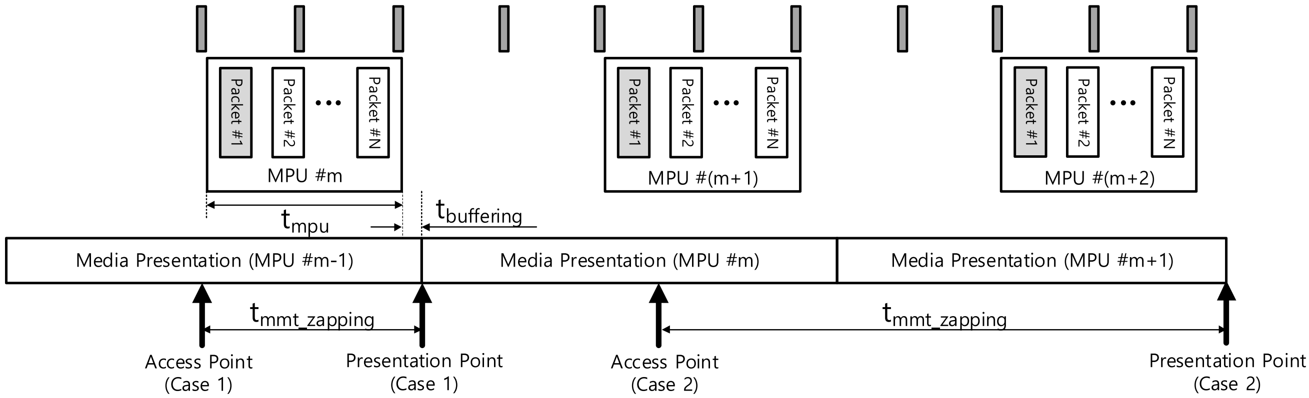


FIGURE 5. Timing model of the channel zapping for the MPEG-media-transport (MMT) services.

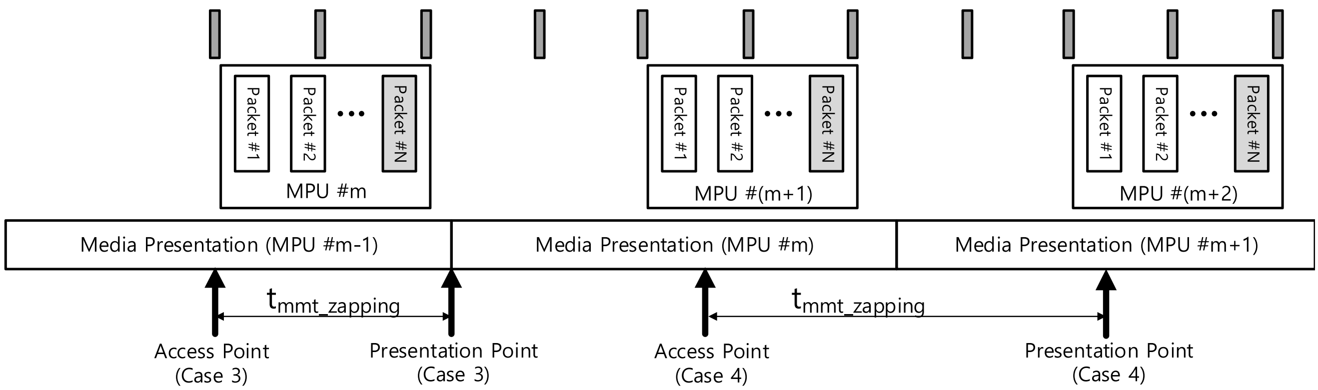


FIGURE 6. Timing model of the channel zapping for the MPEG-media-transport (MMT) services when reordering is applied.

the mth MPU (e.g., $t_{buffering}$). Note that the $t_{buffering}$ could be affected by the status of the delivery networks. For instance, when the network performance is poor, the t_{mpu} can be long and the MMT client must play back the MPU immediately after the MPU is assembled and decoded to meet the presentation time of the MPU. In Case 2, the MMT client failed to receive the signaling message before the arrival of the first packet of the $(m + 1)^{th}$ MPU. Although the client can receive a signaling message in the middle of the MPU, as well as the remaining packets of the $(m + 1)^{th}$ MPU, the MMT client cannot play back the contents in the received MPU because it failed to receive the metadata. This results in the addition of the $t_{mpu\_duration}$ to the overall channel-zapping time, and the MMT client can play back only the contents in the $(m + 2)^{th}$ MPU. For the MMT services, the range of the channel-zapping time ($t_{mmt\_zapping}$) can be simplified as follows:

$$\begin{aligned} t_{min} &< t_{mmt\_zapping} < t_{max} \\ t_{min} &= t_{mpu} + t_{buffering} \\ t_{max} &= t_{mpu} + t_{buffering} + t_{mpu\_duration} \end{aligned} \quad (1)$$

In order to improve the channel-zapping time of the worst case, as shown in Case 2 of Fig. 5, a new method that is based on a reordering of the transmission packets that belong to the same MPU is proposed in this paper. In the proposed method, the packets delivering the metadata, called *key packets* in the present study, are placed into the back of the transmission sequence, thereby enabling the MMT client to receive the key packets with higher probabilities. Figure 6 shows the channel-zapping times of the MMT services when the transmission-packet reordering was applied. In this figure, Case 3 is exactly the same as Case 1, and the transmission of the packets in the reverse order does not influence the overall channel-zapping time. Contrary to Case 3, however, in Case 4, the MMT client is able to perform an initialization even if it accesses the service in the middle of the $(m + 1)^{th}$ MPU, because the key packets are sent in the end of the sequence. In addition, if the MPU contains multiple SAPs and some of the SAPs that are contained in the packet subsets are received, the MMT client can play back the content of the $(m + 1)^{th}$ MPU from that SAP. On average, the channel-zapping time can be further reduced by ($t_{mpu\_duration}/2$) compared with the nonreordering method, as shown in Fig. 5. When the reordering is applied, the channel-zapping time for the MMT services can be illustrated as follows:

$$\begin{aligned} t_{min} &< t_{mmt\_zapping} < t_{max} \\ t_{min} &= t_{mpu} + t_{buffering} \\ t_{max} &= t_{mpu} + t_{buffering} + \frac{t_{mpu\_duration}}{2} \end{aligned} \quad (2)$$

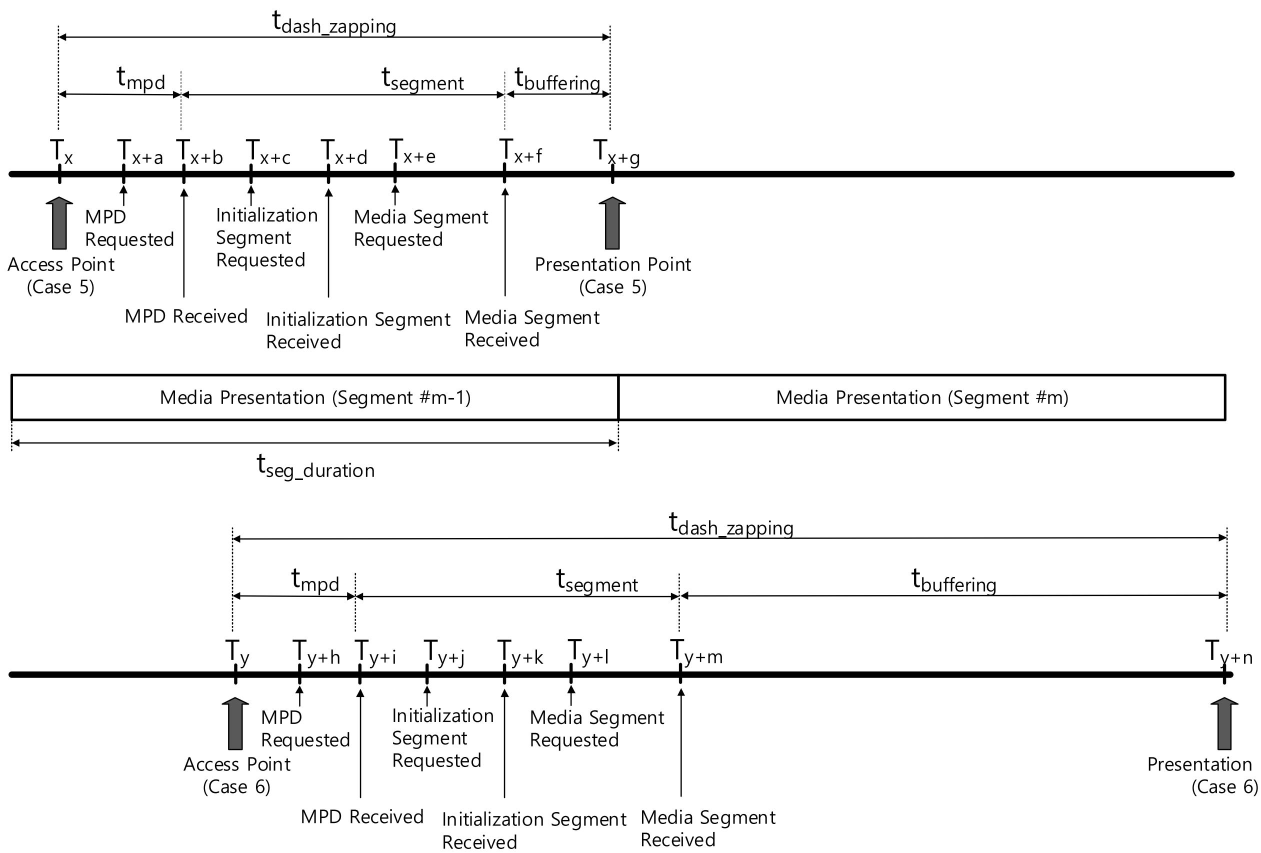


FIGURE 7. Timing model of the channel zapping for the dynamic adaptive streaming over HTTP (DASH) services.

B. MPEG-DASH

Figure 7 shows the channel-zapping times of the DASH model. Similar to the MMT, a DASH client must first request the MPD from a DASH server and obtain a list of the available media-chunk URLs and the other content characteristics. The DASH client can then sequentially request the media chunks as required to receive an uninterrupted playback of the media content. For DASH services, the channel-zapping time ($t_{dash\_zapping}$) can be simplified, as follows:

$$t_{dash\_zapping} = t_{mpd} + t_{segment} + t_{buffering} \quad (3)$$

where t_{mpd} represents the elapsed time between the MPD request and the MPD reception. Whereas the $t_{segment}$ is the amount of time for the receipt of the Initialization Segment and the first-media Segment. The timing model of DASH services is similar to those of the MMT services, but the transmission-packet reordering is not implemented. The difference here is as follows: Unlike MMT, DASH is based on the pull model, and therefore, a DASH client must explicitly request the signaling information (e.g., MPD) before it plays the media content. This behavior eliminates the signaling-information-acquisition delay that occurs in the MMT services, in which the MMT clients must wait for the next signaling message to arrive when they access the channel in the middle of or after the current signaling message.

The worst-case DASH channel-zapping time occurs when the DASH clients are unable to receive the Segment before the next Segment duration starts, as shown in Case 6 of Fig. 7. A DASH client requests and receives the $(m + 1)^{th}$ segment because it cannot make the m^{th} segment playback-ready before its presentation time (e.g., T_{y+m}). In DASH, when the presentation timestamp of the received segment is not within the presentation period of the next segment duration, the DASH client must wait until the presentation time of the received segment. This delay can be influenced by the segment duration (e.g., $t_{seg\_duration}$) because DASH allows only segment-level processing.

IV. EMPIRICAL ANALYSIS

A. PROTOTYPE IMPLEMENTATIONS

To investigate the channel-zapping times of MMT and MPEG-DASH, prototypic implementations were developed for both technologies, and a significant amount of experiments for which the configurable parameters were varied were conducted. Figures 8 (a) and (b) show the architectures of the prototypic implementations for MMT-based and DASH-based media services, respectively. The *Asset Generator*, *Asset Parser*, and *Asset Database* blocks in the MMT Server are responsible for the generation of the asset-related metadata, and the *CI Parser* specifies the spatial and temporal relationships among the assets for the consumption.

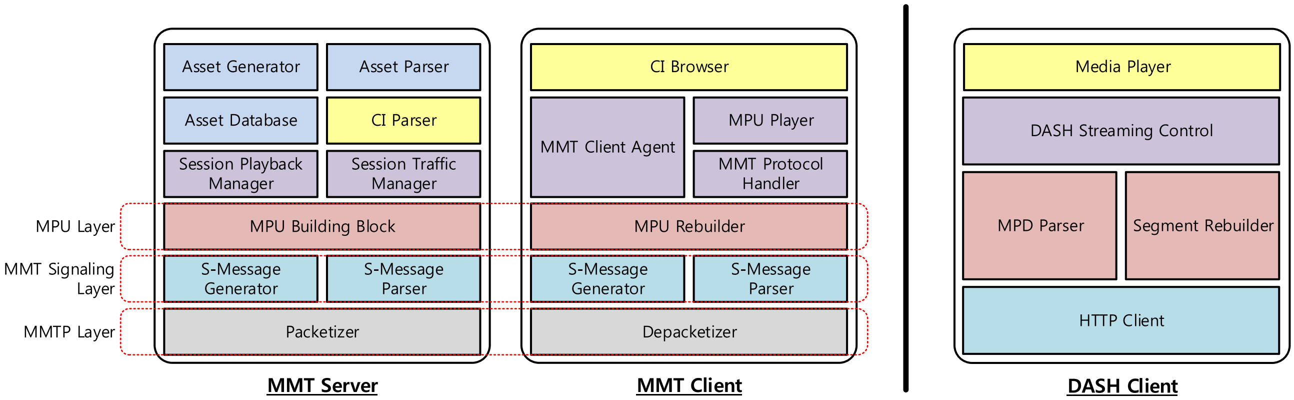


FIGURE 8. Architectures for the prototypic implementations. (a) MMT. (b) MPEG-DASH.

The *Session Playback Manager* and the *Session Traffic Manager* provide the session control for the MMT-packet flows. The *MPU Building Block* generates the MPU according to the *Session Traffic Manager*. The *S-MessageGenerator* and the *S-Message Parser* are responsible for the generation and updating of the signaling messages. The *Packetizer* generates the transport packets according to the underlying transport protocol. The *MMT Client Agent* plays the role of the controlling of the overall MMT-Client operations; for instance, when a service is selected, it informs the *Depacketizer* of the MMTP-packet flows that should be used to receive the media data. The *MMTP Protocol Handler* reassembles the MPUs and transfers them to the *MPU Player* that plays the MPUs according to their formats. The *CI Browser* presents the assets belonging to the same package. In the MPU, MMT-Signaling, and MMTP layers, the MMT client comprises functional blocks that are equivalent to those of the MMT server.

The *HTTP Client* of the DASH client interacts with the HTTP servers and receives the MPD and the Segments. The *MPD Parser* is responsible for parsing the received MPD, whereas the *Segment Rebuilder* handles the downloaded segments and encapsulates the coding formats. *DASH Streaming Control* runs the adaptation logic and determines the downloaded segment based on a given client context. The *Media Player* is responsible for the final decoding and rendering of the media presentation.

B. EXPERIMENTAL RESULTS

For the experiments, two media servers that provide full high-definition (HD) video content with a frame rate of 25 fps, a resolution of 1920 × 1080, and media clients that are connected to the media servers via a gigabit Ethernet interface were deployed (see Fig. 9). In the experimental configuration, when a channel change is requested, a media client is supposed to switch to the other media server running on a different machine to start the processing for the media presentation. The encoding of the video sequences ensured the establishment of the following aspects: i) the GOP size was set as 1000 ms, ii) each GOP starts with *Instantaneous Decoder Refresh* (IDR) frames, and iii) each GOP contains

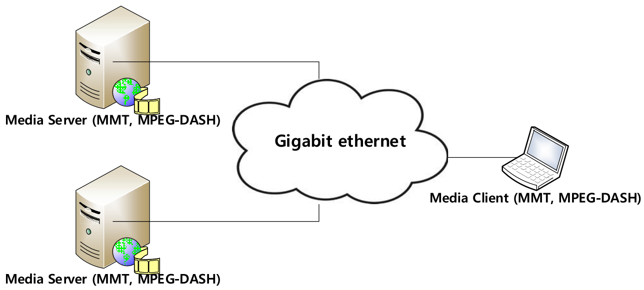


FIGURE 9. The testbed setup for the experiments.

two SAPs including an IDR. The MPU duration and the segment duration were controlled by the numbers of GOPs that were contained in the MPU and the segment, respectively. For instance, for the experimental configuration of the MPU duration of 2000 ms, each MPU contains two GOPs. The frequency of the signaling message was set as 500 ms throughout the experiments.

The comparative channel-zapping times for different MPU duration sizes are shown in Fig. 10. Irrespective of the application of the MPU-transmission-packet reordering, the longer the duration of the MPU, the longer the duration of the channel-zapping time; this is because the average waiting time increases in proportion to the MPU duration size when the MMT client fails to receive a complete set of the packets of the corresponding MPU before the commencement of the next MPU duration. However, when the reordering technique was applied, the MMT client did not need to wait until the cessation of the next MPU duration to start the playback of the selected service. In this case, even if the MMT client accesses a service in the middle of the MPU transmission for the next duration, it is highly likely that the client will receive the key packets and a SAP before the next duration starts, and this is owing to the transmission-packet reordering; therefore, it can play back the latter part of the media contents that were delivered in the partially received MPU. In the experiments of the present study, the channel-zapping time can be reduced by up to 20 % with the reordering of the MPU transmission packets. Figure 11 is another graph that shows the positive

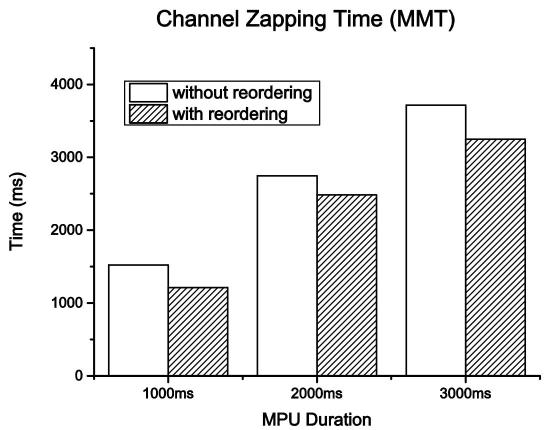


FIGURE 10. Comparative channel-zapping times for the MPEG-media-transport (MMT) services according to the media-processing-unit (MPU) durations.

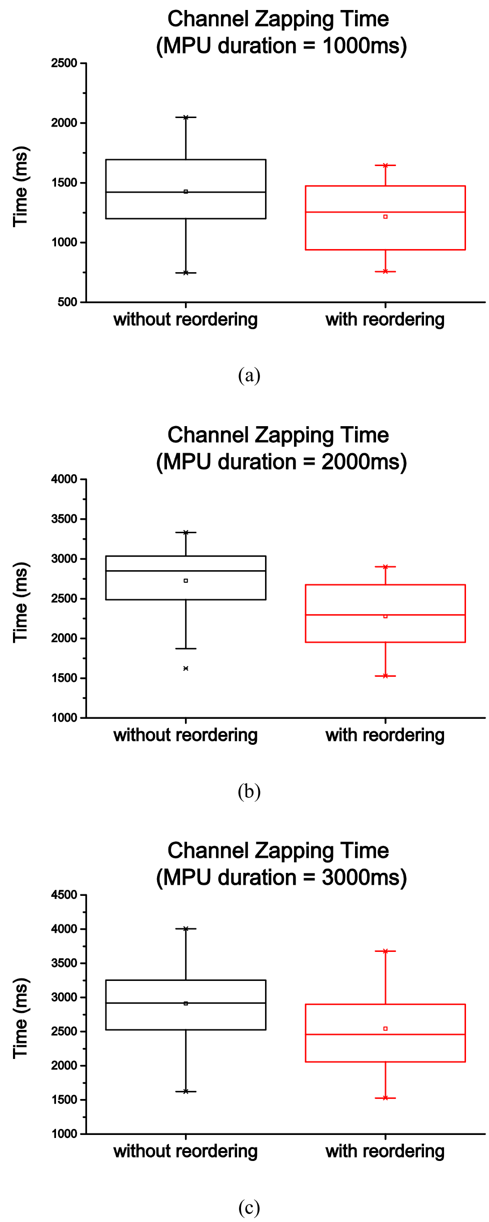


FIGURE 11. The effect of the transmission-packet reordering. (a) Media-processing-unit (MPU) duration = 1000 ms. (b) MPU duration = 2000 ms. (c) MPU duration = 3000 ms.

effect of the reordering of the MPU transmission packets. The experiment results clearly show that the channel-zapping time in the case without the reordering is higher than that in the case wherein the reordering was applied. This is due to the fact that the reordering allows the MMT client to play back the media contents in the partially received MPU as long as the received MPU contains the key packets and an SAP.

Figure 12 shows the performances of the primary components comprising the overall channel-zapping time when the MPU-transmission-packet reordering was applied. In this figure, $t_{reception}$ denotes the total elapsed time for the receipt of a complete or partial MPU for a media presentation. Therefore, it includes the elapsed time for the receipt of a signaling message from the access point and the elapsed time for the receipt of the key packets, as well as the SAPs of the MPU for the next MPU duration. The $t_{reception}$ can also include the waiting time until the key packets of the next MPU are received in a case where the MMT client fails to receive the key packets from the current MPU. In theory, the waiting time is less than the MPU duration. As the MPU-duration size was increased, the $t_{reception}$ was also increased, and this is because the average waiting time for the receipt of the next key packets in the case where the MMT client failed to receive the key packets of the current MPU was increased in proportion to the MPU-duration size. Alternatively, the $t_{buffering}$ does not fluctuate significantly depending on the MPU-duration size. The decoding time of the received MPU and the waiting time before the presentation of the decoded MPU are the primary components of the $t_{buffering}$. The decoding process requires only a few milliseconds, and although the larger MPU durations require a longer decoding time, the difference is not significant when the MPU duration is between 1000 ms and 3000 ms. In addition, the transmission-network condition that was used for the experiments is stable, and therefore, a significant change of the waiting time did not occur.

The comparative channel-zapping times of the DASH services for which different segment-duration sizes were utilized are shown in Fig. 13. As with the MMT services, the segment-duration size affects the channel-zapping time. The reason is that, depending on the presentation time of the received segment, the DASH client can play back the media content in the segment from the next segment duration, or it must wait until the next segment duration ends and then start the playing back of the media content in the received segment. In the latter case, the $t_{buffering}$ was increased significantly. Interestingly, unlike the MMT services where the $t_{reception}$ is the primary component that affects the channel-zapping time, the performances of the components that correspond to the $t_{reception}$ in the DASH services (e.g., t_{mpd} and $t_{segment}$) are relatively stable, and the $t_{buffering}$ is the primary component that determined the channel-zapping time of the DASH services.

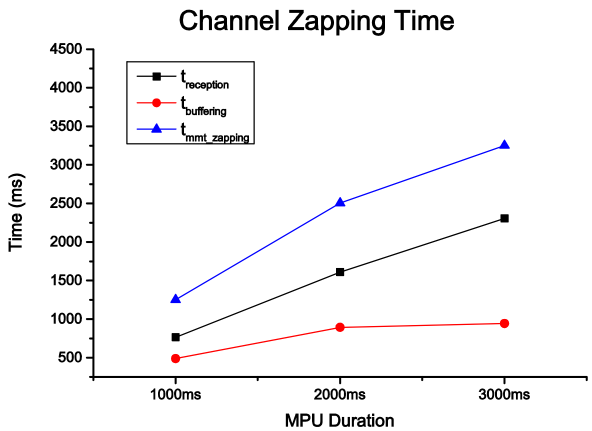


FIGURE 12. Performances of the individual components comprising the channel-zapping time for the MPEG-media-transport (MMT) services.

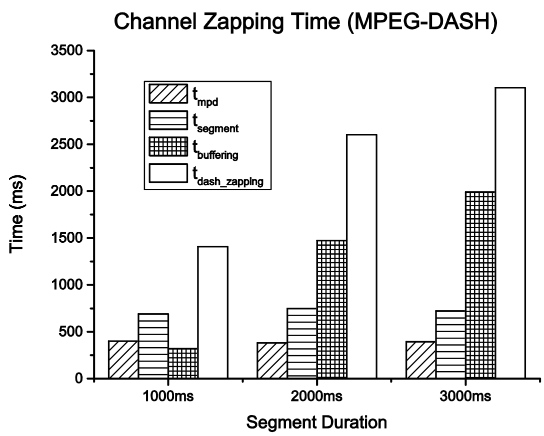


FIGURE 13. Comparative channel-zapping times for the dynamic adaptive streaming over HTTP (DASH) services according to the segment durations.

Since MPEG-DASH is based on the pull model, the DASH client is able to acquire the necessary MPDs and segments without additional delays, with the exception of the transmission time. In addition, as the underlying delivery network that was used for the experiment is consistent in terms of the network-delay performance, the transmission time for the MPDs and segments did not oscillate significantly according to the time and segment sizes. The $t_{buffering}$, however, can vary depending on the presentation time of the received segment, and it is highly influenced by the segment duration and the service access point.

Figure 14 shows a comparison of the average channel-zapping time of the MMT and DASH services with variations of the MPU and the segment-duration sizes. The channel-zapping time of the MMT service with the MPU-transmission-packet reordering and that of the DASH service are similar, and it is not clear which of the technologies is comparatively superior. The difference in the performances is negligible, and both of the technologies achieved acceptable channel-zapping times through a controlling of the MPU, segment sizes, and frequency of the signaling messages.

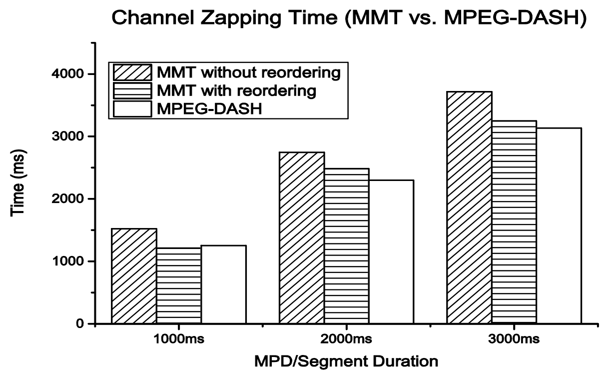


FIGURE 14. Comparative channel-zapping times for the MPEG-media-transport (MMT) and dynamic adaptive streaming over HTTP (DASH) services.

However, the bandwidth costs are different; for example, for a client to be able to channel-change at every single SAP, the required service information needs to be transmitted at least once every SAP. The minimum service information that is required for the DASH services consist of the "ftyp" and "moov" boxes in the initialization segment and the MPD [19]. For the MMT services, the ftyp and moov boxes in the MPU metadata and the minimum signaling tables, e.g., the MP, PA, DCI, and MPI tables in a single PA message, are required, as specified in [2]. The typical sizes of the minimum service information and the bandwidth costs for the achievement of the maximum 1-s channel-zapping time are shown in Table I.

TABLE 1. Minimum set of service information.

Technology	Service Information	Typical Sizes	Bandwidth Costs
DASH	Initialization segment	3.8 KB	60.8 kb/s
	MPD	1.6 KB	25.6 kb/s
MMT	MPU metadata	3.8 KB	60.8 kb/s
	Signaling tables	156 bytes	0.25 kb/s

To achieve the 1-s channel-zapping time, the SAPs must be sent at least once every 0.5 s, and this requires the segment and MPU durations, as well as the signaling-information frequency, to be set as 0.5 s. Therefore, the required bandwidth cost for the MPU metadata, for example, is 3.8 KB $\times$ 2 $\times$ 8 = 60.8 kb/s, while that for the signaling table is 0.156 KB $\times$ 2 $\times$ 8 = 0.25 kb/s.

V. CONCLUSION

MMT and MPEG-DASH are the latest content-delivery technologies that have been developed by the MPEG. Due to their flexibility and extensibility, they have been adopted as the service delivery and synchronization protocols for the next-generation broadcasting systems, as well as for many Internet live-streaming services. In this paper, the channel-zapping time, which is an important metric for the measurement of the QoE for broadcast services, is discussed with respect to both technologies. In addition, a simple yet effective method that enables a fast channel zapping for the MMT services through a MPU-transmission-packet reordering is proposed. The channel-zapping times of both technologies were compared while the import-configuration parameters were varied, and it was concluded that both technologies can achieve acceptable channel-zapping times, even in the broadband-network environment. But, their bandwidth costs differ due to the differences between the signaling information that is required for the service acquisition.

REFERENCES

[1] *Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Part 1: Systems*, document ISI/IEC 13818-1, 2013.

[2] *Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 1: MPEG Media Transport (MMT)*, document ISO/IEC 23008-1:2014, 2014.

[3] *Information Technology—Dynamic Adaptive Streaming Over HTTP (DASH)—Part 1: Media Presentation Description and Segment Formats*, document ISO/IEC 23009-1:2012, 2012

[4] T. Stockhammer and M. G. Luby, "DASH in mobile networks and services," in *Proc. IEEE Visual Commun. Image Process. (VCIP)*, Nov. 2012, pp. 1–6.

[5] Y. Xu, S. Xie, H. Chen, L. Yang, and J. Sun, "DASH and MMT and Their Applications in ATSC 3.0," *ZTE Commun.*, vol. 14, no. 1, pp. 39–49, 2016.

[6] B. Li, C. Wang, Y. Xu, and Z. Ma, "An MMT based heterogeneous multimedia system using QUIC," in *Proc. 2nd Int. Conf. Cloud Comput. Internet Things (CCIOT)*, Oct. 2016, pp. 129–133.

[7] H. Jang, J. Lee, H. Choi, and S. Cho, "Study on the field test result of mobile MMT trial service over LTE network at open dense area, subway and high speed train," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2016, pp. 11–16.

[8] M. Park and D. Oh, "MMT based hybrid delivery system for a highly reliable UHD broadcasting service," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2016, pp. 19–21.

[9] S. Aoki, "Development of media transport protocol for 8K super hi–vision satellite broadcasting system using MMT," *Broadcast Technol.*, no. 61, pp. 12–21, 2015. [Online]. Available: http://scinet.dost.gov.ph/union/ShowSearchResult.php?s=2&f=&p=&x=&page=&sid=1&id=Development+of+media+transport+protocol+for+8K+super+Hi%E2%88%92Vision+Satellite+Broadcasting+System+using+MMT&Mtype=ANALYTICS

[10] S. Aoki, Y. Kawamura, K. Otsuki, N. Nakamura, and T. Kimura, "Development of MMT-based broadcasting system for hybrid delivery," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2015, pp. 1–4.

[11] Y. Heo *et al.*, "True realtime multimedia streaming system based on MMT," in *Proc. IEEE Int. Conf. Multimedia Expo Workshop (ICMEW)*, Jul. 2015, pp. 1–3.

[12] *YouTube*. Accessed on Jun. 2017. [Online]. Available: http://www.youtube.com

[13] P. Siebert, T. N. M. Van Caenegem, and M. Wagner, "Analysis and improvements of zapping times in IPTV systems," *IEEE Trans. Broadcast.*, vol. 55, no. 2, pp. 407–418, Jun. 2009.

[14] R. Kooij, K. Ahmed, and K. Brunnstrom, "Perceived quality of channel zapping," in *Proc. 5th IASTED Int. Conf. Commun. Syst. Netw. (CSN)*, Aug. 2006, pp. 155–158.

[15] B. Lee, "Considerations for reducing channel zapping time for protected services in IP datacast over DVB-H," *IEICE Trans. Commun.*, vol. E94B, no. 6, pp. 1699–1706, 2011.

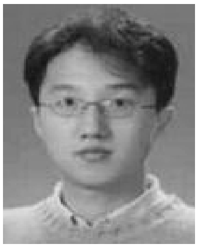
[16] Y. Lim, S. Aoki, I. Bouazizi, and J. Song, "New MPEG transport standard for next generation hybrid broadcasting system with IP," *IEEE Trans. Broadcast.*, vol. 60, no. 2, pp. 160–169, Jun. 2014.

[17] T. Paila, R. Walsh, M. Luby, V. Roca, and R. Lehtonen, "FLUTE—File delivery over unidirectional transport," document IETF RFC 3926, 2012.

[18] J. Xu, C. Chen, and K. Park, "A method for fast channel switching based on MPEG media transport for broadcast TV," in *Proc. IEEE Int. Conf. Multimedia Expo Workshop (ICMEW)*, Jul. 2015, pp. 1–4.

[19] *SCS and MPD Sizing for ROUTE/DASH Proposal*, document ATSC Working Draft S33-1-144r3, 2014.

KYUNGMO PARK (M'05) received the B.S., M.S., and Ph.D. degrees in electronic engineering from Kyung Hee University, South Korea, in 2003, 2005, and 2016. After he joined Samsung Electronics Co., Ltd., Seoul, South Korea, in 2005, he actively participated in the standardization and development of multimedia technologies, such as MPEG, 3GPP, ATSC, DLNA, and CEA. He has been involved in multimedia delivery system and its application in broadcasting and mobile network services. He is currently a Senior Standards Engineer with Samsung Electronics Headquarter and serving as a Chairman of *Ad Hoc* Group, an editor of various MPEG Systems standards, and a rapporteur of 3GPP SA4 specification. His research interests stretch over all aspects of multimedia communications and multimedia, with a strong focus on standardization activities. He also holds several patents and patent applications in the above-mentioned areas and applications.



NAMGI KIM received the B.S. degree in computer science from Sogang University, South Korea, in 1997, and the M.S. and Ph.D. degrees in computer science from KAIST in 2000 and 2005, respectively. From 2005 to 2007, he was a Research Staff of Samsung Electronics. Since 2007, he has been a Faculty Member of the Kyonggi University, South Korea, where he is currently an Associate Professor with the Department of Computer Science. His research interests include sensor system, wireless system, and mobile communication.

BYOUNG-DAI LEE received the B.S. and M.S. degrees in computer science from Yonsei University, South Korea, in 1996 and 1998 respectively, and the Ph.D. degree in computer science and engineering from the University of Minnesota, Minneapolis, MN, USA, in 2003. He was with Samsung Electronics Co., Ltd., as a Senior Engineer from 2003 to 2010. During the period, he has participated in many commercialization projects related to mobile broadcast systems. He is currently an Associate Professor with the Department of Computer Science, Kyonggi University, South Korea. His research interests include cloud computing, mobile multimedia platform, and mobile multimedia broadcasting.