An Efficient Transmission Scheme of MPEG2-TS over RTP for a Hybrid DMB System

Hyung-Yoon Seo, Byungjun Bae, and Jong-Deok Kim

Hybrid digital multimedia broadcasting (DMB) is a next-generation mobile TV system that combines broadcasting and wireless communication networks and can provide various high-quality multimedia services. However, if a system adheres to the current standard of transmitting the DMB content in the form of MPEG2-TS through wireless networks, it results in a burden on the network due to low transmission efficiency. The reasons for the low transmission efficiency are as follows. First, due to its constant bitrate characteristic, DMB MPEG2-TS includes a considerable amount of needless information, such as NULL packets and stuffing bytes. Second, due to the inflexibility of the Real-time Transport Protocol (RTP) standard, one cannot fully utilize the maximum transmission unit of the network when converting MPEG2-TS to RTP stream for transmission. This paper proposes a new transmission scheme that resolves these problems. Experiment results show that the proposed scheme improves data bitrate transmission efficiency by 8% to 36%, compared to the standard scheme, in the streaming of various real-DMB contents.

Keywords: Hybrid DMB, MPEG2-TS over RTP.

Manuscript received Feb. 24, 2012; revised June 29, 2012; accepted Aug. 13, 2012.

This work was supported by the Korea Communications Commission (KCC) under the R&D program supervised by the Korea Communications Agency (KCA) (KCA-2012-11912-02001).

Hyung-Yoon Seo (phone: +82 51 510 3941, tanyak@pusan.ac.kr) and Jong-Deok Kim (corresponding author, kimjd@pusan.ac.kr) are with the Department of Computer Science and Engineering, Pusan National University, Busan, Rep. of Korea.

Byungjun Bae (1080i@etri.re.kr) is with the Broadcasting & Telecommunications Media Research Laboratory, ETRI, Daejeon, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.13.0112.0124

I. Introduction

Terrestrial digital multimedia broadcasting (T-DMB) [1], referred to as DMB hereafter, is a mobile TV standard leading the mobile TV market through successful commercialization in several countries [2]. With the ever-increasing popularity of mobile personal devices, such as smartphones and tablet PCs, mobile TV is attracting much interest because it is regarded as a key service for these devices. In fact, in the Republic of Korea, most android-based smartphones and tablets support DMB [2]. These devices have an exclusive chip that can receive and demodulate the DMB RF signal and decode and display DMB media encoded in H.264 [3] and bit sliced arithmetic coding (BSAC) [4].

However, DMB is facing difficulties due to its limited service coverage and insufficient broadcasting bandwidth. Generally, DMB transmits signals via the base stations of the terrestrial broadcasting system and its service coverage is similar to that of radio and TV. It can receive signals from vehicles moving at over 100 km/h. However, there are places, such as subways, tunnels, and the inside of buildings, where the DMB signal cannot be received easily. While a gap filler, which is a DMB signal repeater station, can be used to cover those areas [5], DMB broadcasters are reluctant to use gap fillers because of the cost, which results in many DMB shadow areas. DMB uses a 6-MHz channel in the VHF band, and the channel consists of three ensembles. The effective data rate of an ensemble is approximately 1.5 Mbps. In the case of the Republic of Korea, one broadcaster uses one ensemble and generally provides two TV channels, one radio channel, and one or two data service channels. While broadcasters want to expand or improve their services in response to users' requirements, the limited broadcasting bandwidth prevents

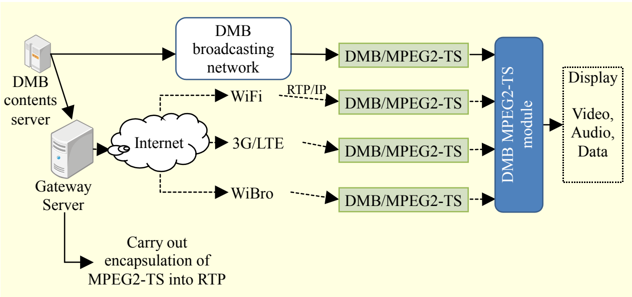

Fig. 1. DMB-MPEG2-TS-based transmission in hybrid DMB system.

them from doing so.

Hybrid DMB was introduced to alleviate the aforementioned limitations of DMB by making use of various mobile wireless communication networks, such as WiFi, 3G/LTE, and WiBro (the Korean version of Mobile WiMAX). It can transmit multimedia content using either or both a broadcasting network and a wireless network, as needed, to cover DMB shadow areas or to expand bandwidth. (Note that backward compatibility is not a problem because hybrid DMB does not change the existing DMB broadcasting technology.) We can realize a hybrid DMB system using the existing network infrastructures and user terminals, as most smartphones and tablets, except iPhones and iPads, are already equipped with a DMB-specific module as well as wireless communication modules. Although most devices currently suitable for hybrid DMB in the real world are based on the Android OS, hybrid DMB is not technically dependent on the Android OS and can be realized in other OSs.

However, the multimedia streaming format or protocol of DMB is different from those that are widely used in such general Internet multimedia services as YouTube. For example, DMB adopts a proprietary multiplexing scheme called MPEG4 sync layer on MPEG2-TS (M4onM2), uses BSAC to encode audio, and includes MPEG4 binary format for scene data for interactive multimedia services [6]. These are seldom used or supported in general Internet multimedia systems. As a result, processing a DMB stream in an Android system using its own multimedia features is not straightforward. This is why an exclusive DMB module is included in most DMB-enabled devices.

For easy integration and synchronization of multimedia streams from the broadcasting network and the wireless network, it is necessary to use the same multimedia streaming format and protocol in both networks. Two options are possible. One option utilizes the DMB MPEG2-TS format and its protocol, while the other utilizes an Internet streaming-friendly format such as those used in YouTube. As it is not possible to change the format and the protocol used in the DMB network due to backward compatibility, we adopt the first option. That is, we transmit multimedia content in the form of DMB MPEG2-TS through both the DMB network and the wireless network, as depicted in Fig. 1. We make use of the Real-time Transport Protocol (RTP) standard to transport MPEG2-TS through an Internet Protocol (IP)-based wireless network. It is easy to process MPEG2-TS in the wireless network, as we can also make use of the embedded DMB chip in the terminal to handle it.

However, the above-mentioned approach has transmission inefficiencies, and some network providers worry about the overhead that will be incurred when the hybrid DMB service is launched. We find two reasons for the low transmission efficiencies. The first reason is the inefficiency of DMB MPEG2-TS itself, which includes a considerable amount of needless information, such as NULL packets and stuffing bytes. The second reason is due to the inflexibility of the RTP standard [7], [8]: we cannot fully use the maximum transmission unit (MTU) of the network when converting MPEG2-TS to an RTP stream, which results in many small RTP packets that aggravate transmission efficiency. This paper proposes a new scheme to transport DMB MPEG2-TS over RTP that is more efficient than the standard one.

II. Background and Related Work

1. DMB MPEG2-TS Multiplexing Architecture

In this subsection, we explain the multiplexing architecture of DMB MPEG2-TS and look at its inherent inefficiency. MPEG2-TS is a standard [9] format for the transmission of multimedia data as well as the control data or metadata, including program specific information (PSI) and system information (SI). MPEG2-TS is used in such broadcast systems as DVB, ATSC, and DMB.

As illustrated in Fig. 2, an access unit of multimedia data is encapsulated in a packetized elementary stream (PES) packet that can vary in size. PSI/SI is carried in the form of tables; the program association table (PAT) and the program map table (PMT) are examples of those tables. PSI/SI tables also vary in size. Variable sized PES packets and PSI/SI tables are further encapsulated in MPEG2-TS packets. An MPEG2-TS packet is a basic unit in an MPEG2 transport stream and has a fixed length of 188 B, including a 4-B header. A PES packet or a PSI/SI table may span multiple MPEG2-TS packets. If the length of a PES packet or a PSI/SI table is not a multiple of 184 B, the last MPEG2-TS packet is filled with stuffing bytes (0xff). Of course, these stuffing bytes reduce the efficiency of MPEG2-TS.

A fixed rate DMB subchannel is allocated to transmit

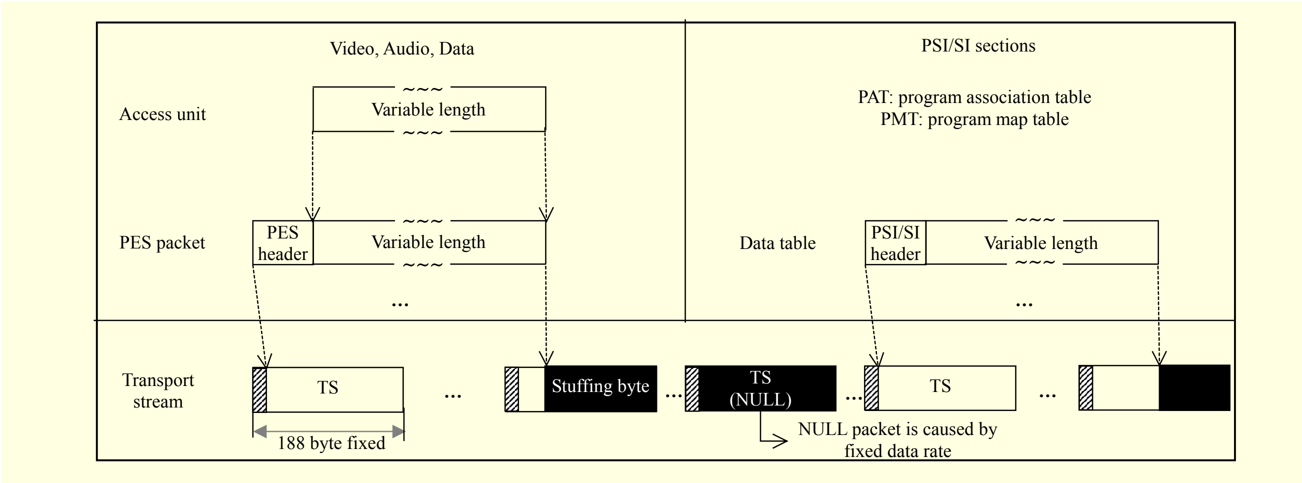

Fig. 2. MPEG2-TS multiplexing architecture.

MPEG2-TS through the DMB network. To fill the allocated rate of the subchannel, we must generate a constant bitrate MPEG2-TS that strictly fits to the rate.

To realize this, some additional MPEG2-TS packets referred to as NULL packets are inserted into the stream. The payload of the NULL packets does not contain any valid data, so the receiver can safely ignore them. As will be shown in subsection II.3, the number of NULL packets and stuffing bytes is far from negligible. According to our experiment on various real DMB streams, it comprises 4% to 29% of the source data rate. If we remove NULL packets and stuffing bytes from the transmission, we can significantly improve transmission efficiency by a corresponding percentage.

2. Encapsulation of MPEG2-TS in RTP

RTP defines a standardized packet format for delivering audio and video over IP networks. RTP is used extensively in various real-time Internet multimedia systems, including Voice over IP (VoIP). One of the key design principles of RTP is its support for various multimedia formats, such as H.264 video, MPEG-4 elementary stream, and MPEG2-TS, without changes to the RTP standard. To achieve this end, the concept known as application level framing is adopted. That is, the information required for a specific media format's needs is not included in the generic RTP header but is provided through media-specific RTP profiles and payload formats. The profile and payload format for a specific media format is usually defined in a separate specification [10]. For example, the profile and payload format for MPEG2-TS is defined in RFC-2250, and the generic RTP header is defined in RFC-3350.

RFC-2250 specifies that the RTP payload of MPEG2-TS has no specific payload header and contains an integral number of

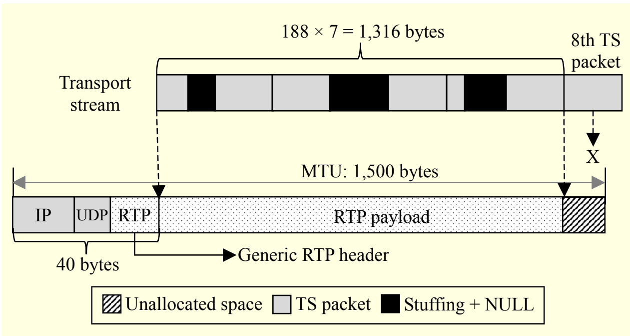

Fig. 3. Standard RTP encapsulation structure of MPEG2-TS.

MPEG2-TS packets. While it is simple and easy to implement, the payload format is not flexible in size, which may result in transmission inefficiency. We will show some examples as to why inflexibility in size may result in transmission inefficiency.

Figure 3 depicts the RTP encapsulation structure of MPEG2-TS for an MTU of 1,500 B. MTU is the largest packet size that can be sent in a network. In fact, 1,500 B is the MTU size for Ethernet, but it is called Internet MTU because most Internet hosts use this size to generate IP datagrams to avoid fragmentation. Generally, a larger packet results in greater efficiency [11], [12] because each packet carries more user data and the header overhead remains fixed. Thus, it would be better to make RTP packets as large as MTU. However, as shown in Fig. 3, an RTP packet can contain as much as seven MPEG2-TS packets for the MTU, as it must have an integral number of MPEG2-TS packets, which is 144 B smaller than the MTU. The header overhead of an RTP packet is 40 B, which comprises 20 B of IP header, 8 B of UDP header, and 12 B of generic RTP header. If we can enlarge the RTP payload to the limit of the MTU, we can reduce the header overhead to the user payload size ratio, from 3.04% to 2.74% in this case.

One may argue that the header overhead is not substantial,

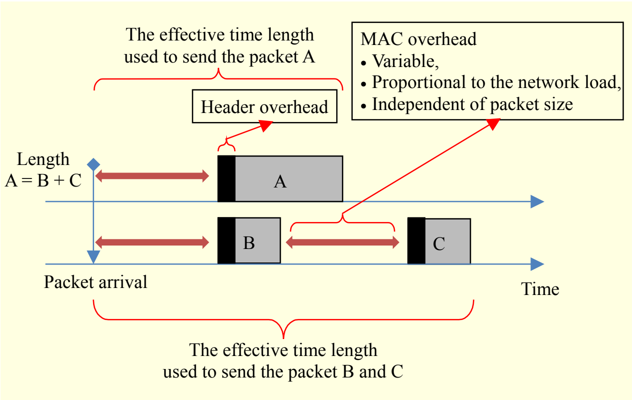

Fig. 4. Scenario in which packets have same length of data but different transmission times in 802.11 WLAN.

but, even so, there are better solutions (such as Robust Header Compression [ROHC], which will be addressed later in this subsection). Note that header overhead is not the only thing that constitutes protocol overheads. Consider a scenario in which a packet is being sent through an IEEE 802.11 WLAN, as shown in Fig. 4. In addition to header overhead, each packet suffers from variable MAC overhead that results from the collision avoidance algorithm of IEEE 802.11 [13]. The MAC overhead is independent of packet length and usually increases as the network load increases. The MAC overhead is far from negligible and previous studies have shown that we can achieve better performance by using a larger packet in IEEE 802.11 [11]. We can also find similar studies for other networks, such as 3G [12].

However, it is obvious that a larger packet is not always better than a smaller one. For example, a larger packet is generally more vulnerable to error in error-prone wireless access networks. Some studies investigated the optimal packet size for a wireless network considering both protocol overhead and error [14]. Here, we want to stress that the main problem of the current standard that we want to address in this paper is not size itself but rather inflexibility in size. We do not urge that an RTP packet should always be as large as the MTU. If a certain packet size is beneficial, we want to make RTP packets of that size, but it is hard to do in the current standard because of its inflexibility in size. In section III, we propose a new RTP payload format for MPEG2-TS that is more flexible in payload size without changing the generic RTP header.

ROHC is a standard used in the compression of the IP, UDP, and RTP headers of Internet packets to reduce their transmission overheads. ROHC is very useful for VoIP applications. The normal payload size of VoIP is only 20 B to 30 B. Therefore, a header overhead of 40 B to carry those small payloads is too expensive, especially in bandwidth-scarce wireless networks. ROHC can compress the header overhead to only 1 B or 3 B. Though the gain is not as great as for VoIP, it would also be beneficial for a hybrid DMB system to adopt ROHC if the underlying networks support it. That is, in the case of Fig. 3, we can reduce the header overhead from 3% to nearly 0% with ROHC. Note that we can still take advantage of ROHC with our proposal, as it does not change the generic RTP header.

Some Internet-based streaming systems make use of UDP directly to transmit MPEG2-TS packets instead of using RTP. IPTV over ADSL is representative of this approach. As it can eliminate RTP header overhead by 12 B, it may be favorable in terms of transmission efficiency. That is, in the case of Fig. 3, we can reduce the header overhead from 3% to 2.1% with the direct UDP approach. However, this approach still suffers from a similar inflexibility problem to the one we pointed out for RTP. This is because it is also restricted to including only an integral number of MPEG2-TS packets. Moreover, it only makes sense in situations in which the information included in the RTP header is of little use. The main components of a generic RTP header are sequence number and timestamp. The RTP sequence number is typically used to manage packet loss or reordering. While they seldom or never occur in wired networks like ADSL, we cannot rule out those problems in wireless networks. Moreover, it has obvious advantages in the implementation aspect, as it can use the existing RTP library and the FEC relevant feature for error-handling instead of implementing its own program. The timestamp is a little tricky, as there is a similar field in MPEG2-TS itself. However, we are currently developing a scheme that makes use of the RTP timestamp for the synchronization of different media streams in hybrid DMB. Therefore, though it is not essential to use RTP in a hybrid DMB system, we think we can still take advantage of RTP even though we must improve it by resolving its inflexibility problem.

3. Data Rate Analysis of Real DMB MPEG2-TS Streams

In this subsection, we examine the results of data rate analysis for real DMB streams. Six live DMB MPEG2-TS streams are captured from the DMB TV channels of three different Korean DMB broadcasters: KBS, KNN, and MBC. For each broadcaster, we capture two streams comprising different program categories: news and drama. Each stream is captured for about one hour.

We first measure the bitrate of the MPEG2-TS source and the average bitrate of the void data, including the NULL packets and the stuffing bytes in the source. We then encapsulate the MPEG2-TS source into RTP following the standard and measure the rate of the RTP stream. The results are shown in Table 1.

Table 1. Average bitrate of MPEG2-TS, void, and standard RTP.

Broadcasting station		MPEG2-TS (kbps)	Void (stuffing + NULL) (kbps)	(%)	Standard RTP (kbps)
KBS	Drama	400	45	11.25	425
	News	400	70	17.5	425
KNN	Drama	475	21	4.42	500
	News	475	21	4.42	500
MBC	Drama	460	136	29.5	485
	News	460	52	11.3	485

The bitrates of the source streams are different for different broadcasters. However, all the source streams from the same broadcaster have the same bitrate because a fixed rate is allocated to its channel. For example, both the news and drama streams for KBS have the same bitrate of 400 kbps.

The average bitrates of the void data are very diverse for different streams. They range from 4% to 29% of the source bitrates. Thus, it is clear that the void data consumes considerable amounts of bandwidth for no practical purpose.

The bitrate of an RTP stream is about 5% larger than that of its MPEG2-TS source. Because of the fixed encapsulation structure of the standard, RTP streams from different MPEG2-TS source streams have the same bitrate only if the bitrates of the source streams are the same.

4. Related Work

For the past two decades, there has been a considerable effort to integrate broadcasting and communications based on Internet technologies. For example, Multiprotocol Encapsulation (MPE) [15] and Ultra Lightweight Encapsulation (ULE) [16] were developed for the transport of IP datagrams over MPEG2-TS. MPE and ULE are used in several systems, including DVB-H, a major mobile TV standard used in Europe.

Hybrid DMB uses “MPEG2-TS over IP” to transmit its multimedia content through wireless networks, whereas DVB-H uses “IP over MPEG2-TS” to carry its multimedia content over broadcasting networks. IP over MPEG2-TS also has similar efficiency problems in encapsulation to that of MPEG2-TS over IP (as explained earlier). For example, as MPEG2-TS over IP needs stuffing bytes in the encapsulation of a PES packet into DMB MPEG2-TS, MPE and ULE require “padding” bytes in the encapsulation of an IP datagram, as illustrated in Fig. 5. To improve efficiency, MPE and ULE adopt a scheme called “packing,” in which an MPEG2-TS packet may have fragments of different IP datagrams to eliminate padding space.

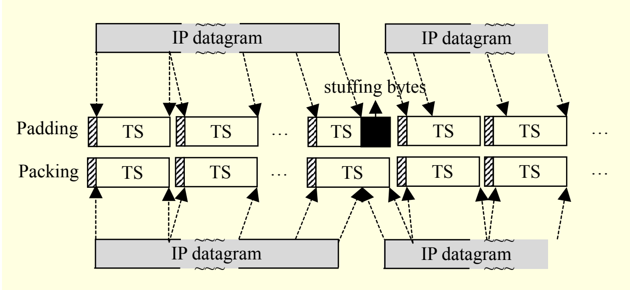

Fig. 5. Encapsulation of IP datagram in MPEG2-TS packets.

In addition to the MPE and ULE standards, there have been several studies [17]-[20] dedicated to improving the transmission efficiency of IP over MPEG2-TS. However, serious research papers on the transmission efficiency of MPEG2-TS over IP are rare [21], [22], although a standard has been established for MPEG2-TS transmission over RTP.

We conjecture that the reason for this is the absence of need. While IP over MPEG2-TS is used in several systems, including DVB-H, MPEG2-TS over IP was used almost exclusively in IPTV until recently. However, IPTV uses wired links that are less sensitive to bandwidth limitations than wireless links. Moreover, the significant RTP transmission overhead in IPTV is overlooked because there are more urgent issues, such as video encoding. In fact, we find that studies on IPTV efficiency usually focus on improving encoding efficiency by utilizing the latest encoding technologies, such as H.264, instead of the MPEG2 used in the terrestrial digital video broadcasting systems. After reviewing many previous studies, we have come to believe that this paper is one of the first studies to deal with the transmission efficiency of MPEG2-TS over IP.

III. Proposed Transmission Scheme of MPEG2-TS

Figure 6 shows the multiplexing structure of the RTP payload proposed in this paper. There are two options for the proposed multiplexing structure: Remove needless information (RNI) removes the stuffing bytes and NULL packet before transmitting the RTP packets; Remove needless information + packing (RNI+) applies the packing to the RNI option and transmits the RTP packets.

To remove the stuffing bytes and NULL packets, knowledge of the packet structure of MPEG2-TS is essential. An MPEG2-TS packet is composed of two classes: one is the PSI/SI section that contains program information, and the other is the data packet containing such data as video and audio.

Figure 7 shows the field structure that can get the length of

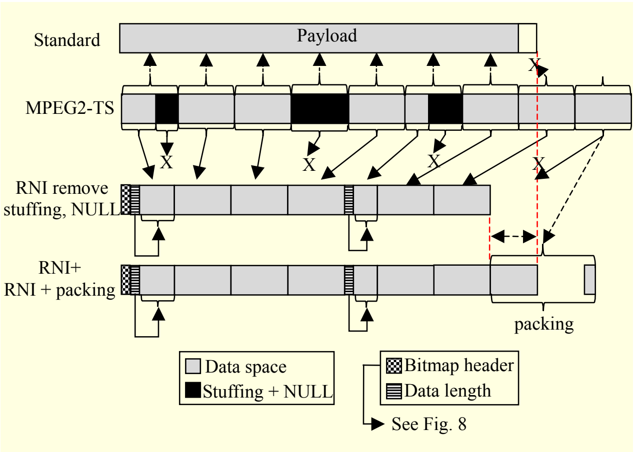

Fig. 6. RTP payload multiplexing structure.

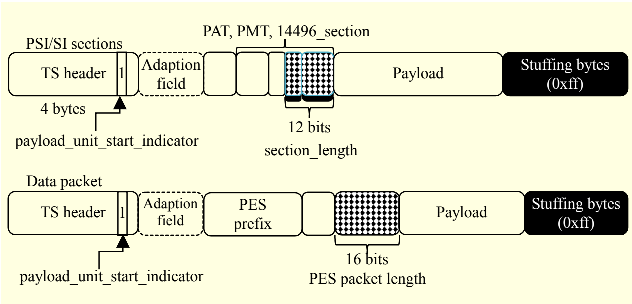

Fig. 7. Field structure for obtaining information on length of TS data from TS packet.

the data from an MPEG2-TS packet. If the payload_unit_start_indicator of the TS header is set to 1, this indicates the start of the PSI/SI section and data packet, and the length information given by section_length has the length information of the PSI/SI section, while PES_packet_length has the length information of data packets [9]. With this length information, we can see the length of the last MPEG2-TS packet and can remove stuffing bytes. NULL packets can be removed easily because the packet ID (PID) of each NULL packet is reserved.

If an RTP packet is transmitted by the proposed multiplexing method, the receiver should recognize and process the multiplexing structure. Figure 8 depicts the proposed multiplexing structure and the relationship between the bitmap header and the MPEG2-TS packets.

The bitmap header contains information about the structure of an RTP packet. The structure information contained in the bitmap header comprises such information as from where the NULL packet has been removed, whether the MPEG2-TS packet is 188 B or not, and the location where packing has been applied to the MPEG2-TS packet. As a result, the proposed transmission scheme is more flexible than the standard scheme. The receiver can restore the original

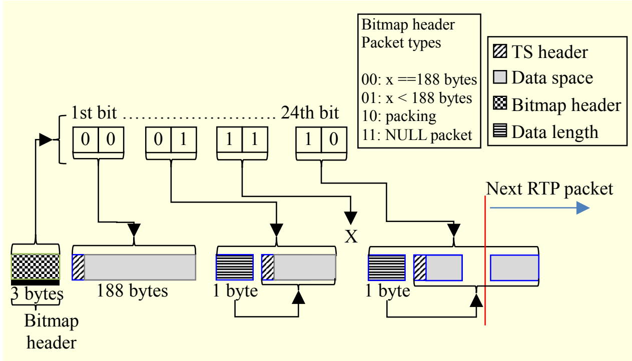

Fig. 8. Proposed multiplexing structure and relation between bitmap header and MPEG2-TS packets.

MPEG2-TS packet by using information from the bitmap header. The bitmap header is at the beginning of the RTP packet payload and consists of 24 bits.

The bit sequence of the bitmap header corresponds to that of the MPEG2-TS packet sequence. The first two bits having value "00" in the bitmap header shown in Fig. 8 means that the corresponding first MPEG2-TS packet is 188 B. A value of "01" in the bitmap header means that the corresponding MPEG2-TS packet is smaller than 188 B. If an MPEG2-TS packet is smaller than 188 B, information on the actual length of the MPEG2-TS packet should be given to the receiver. Thus, there is a 1-B length field that signifies the length of the MPEG2-TS packet before the sync byte of the corresponding MPEG2-TS packet. A value of "10" means that the corresponding MPEG2-TS packet is fragmented by packing. Packing occurs at the last MPEG2-TS of an RTP packet, and it is combined with the first part of the MPEG2-TS packet of the next RTP packet. Therefore, there is a length field for the corresponding MPEG2-TS packet that informs the receiver of the length of the packed MPEG2-TS packet to facilitate restoration of the overall MEPG2-TS packet. Thus, the receiver can restore the MPEG2-TS packets by combining the bitmap header "packing information" with the length information. A value of "11" means that the corresponding MPEG2-TS packet has been removed because that packet was a NULL packet.

Figure 9 shows the flow of the initializing program after getting the DMB MPEG2-TS and making a compressed MPEG2-TS/RTP packet, and the process of decompression and restoration of the original MPEG2-TS packet from a received MPEG2-TS/RTP packet.

The process of "initialization of program information" to receive an MPEG2-TS packet distinguishes the NULL packet, PSI/SI, data, and so on by PID. Each of the programs listed in the PAT has an associated value of the PID for its PMT. Thus, all the received packets are dropped until the first PAT is received. The PMT PID that distinguishes each program's

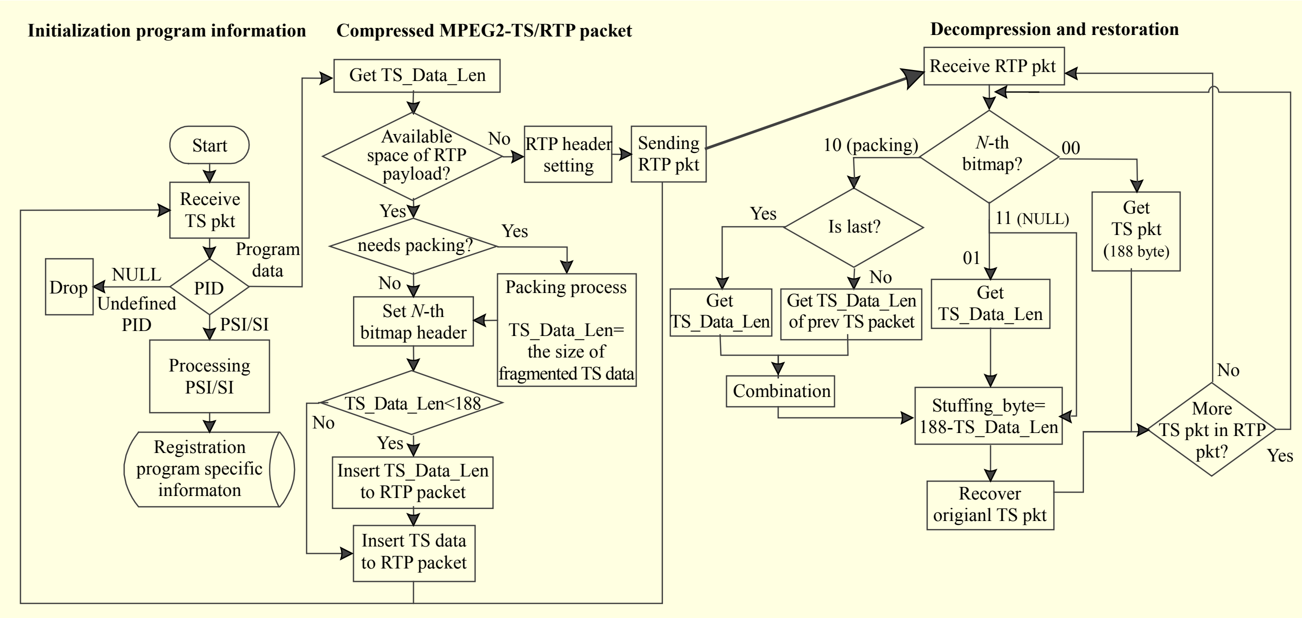

Fig. 9. Flow of initializing program after getting DMB MPEG2-TS and making compressed MPEG2-TS/RTP packet and process of decompression and restoration of original MPEG2-TS packet received from compressed MPEG2-TS /RTP packet.

information is saved when the PAT is received, and each program PID in the PMT is saved when the PMT is received. This process is called the "registration program specific information" process. Next, as each piece of data is received, the data processes the flow of the compressed MPEG2-TS/RTP packet. The following is the process by which a compressed MPEG2-TS/RTP packet is made. The process receives an MPEG2-TS packet and gets the length of the TS data (TS_Data_Len) without the stuffing bytes. After that, the process checks the space available for the TS data to be inserted into the RTP payload. If there is available space and the TS data needs packing, the TS data is fragmented and is buffered by the rest of the TS data after fragmentation (packing process: RNI+).

Because the rest of the TS data after fragmentation is inserted at the first part of the payload of the next RTP packet, the value of the bitmap header that corresponds with the TS data is set to "10," and TS_Data_Len (the size of the fragmented TS data) and the fragmented TS data are successively inserted into the payload of the RTP packet. The buffered TS data is inserted into the first part of the payload of the next RTP packet (packing process: RNI+). If there is available space and the TS data does not need packing, the process differs depending on TS_Data_Len. If the value of TS_Data_Len is 188 B, the value of the bitmap header that corresponds with the TS data is set to "00," and then inserted into the RTP packet payload. If TS_Data_Len is smaller than 188 B, the corresponding bitmap header is set to "01." TS_Data_Len is also inserted into the RTP packet so that the receiver can recognize the length of the TS data, and the TS data is then inserted into the RTP packet. The process is repeated until the RTP payload maximum is reached. Finally, the RTP header is set, and the RTP packet is transmitted.

The following is the process of decompression and restoration that restores the original MPEG2-TS packet. The receiver checks the bitmap header when it receives an RTP packet. If the value of the *N*-th bitmap header is "00," the receiver can find the length of the TS data, which is 188 B. If the value of the *N*-th bitmap header is "01," there is TS data that is smaller than 188 B. In this case, the length of the TS data can be obtained from TS_Data_Len (1-B field), which has information on the length of the TS data. If the value of the *N*-th bitmap header is "10," the corresponding TS data is packed. If the value of the *N*-th bitmap header is "10" and is the value for the last bits, the receiver finds out the length information from TS_Data_Len and keeps that information. Finally, if the value of the *N*-th bitmap header is "10" and is the value for the first bits, the receiver gets TS data from the length information of the previous RTP packet and then restores the MEPG2-TS packet. Based on the length information, the TS data is obtained and the original MPEG2-TS packet is restored by adding stuffing bytes to it. The receiver recovers the original MPEG2-TS packet by repeating the above process until it reaches the maximum length of the received RTP packet payload.

The proposed RTP transmission scheme in this paper is more complex than the standard RTP transmission scheme due to reformatting. Thus, it is necessary to compare the

```
Algorithm commented pseudocode of sender
(comments begin with %)

INPUT: M2TS          % receive a MPEG2-TS packet
S1: M2TS_len = Get M2TS len
             %The length information of a MPEG2-TS packet can get by Fig 7.
        …
U1: if (Max_len > RTP_len+M2TS_len) {
             %A RTP packet cannot exceed the length of MTU(1500bytes).
S2:   if(needs packing){
         …
S3:      B_M=10            % BitMap Header Set to '10'
S4:      M2TS = F_M2TS     % fragmentation M2TS and buffering the
                             rest of M2TS
         or
         M2TS = buffered F_M2TS    % buffered M2TS are inserted
                                     into RTP
S5:      M2TS_len = len    % the size of fragmented M2TS
         } else {
S6:       if(M2TS_len < 188) {    % if a MPEG2-TS packet length smaller
                                    than 188 bytes.
S7:         B_M=01       % BitMap Header Set to '01'
S8:         RTP=M2TS_len    % insert MPEG2-TS length to RTP packet
          }
        }
U2:   RTP=M2TS      % Removes its stuffing byte from MPEG2-TS and
                      insert to RTP packet
      }
OUTPUT: RTP=[BitMap, M2TS1, M2TS2, … M2TSn ]
```

Fig. 10. Pseudocode for making standard RTP packet and proposed RTP packet after receiving MPEG2-TS packet.

```
Algorithm commented pseudocode of receiver
(comments begin with %)

INPUT: RTP    % receive a RTP packet
…
R1: if(BitMap==01} {                  % The data smaller than 188bytes.
R2:   M2TS_len=Get M2TS len           % Get the length information of a
                                        MPEG2-TS packet
R3: } else if(BitMap == 10){          % packing
R4:   prev_M2TS_len = Get M2TS len    % Get the length information of
                                        current MPEG2-TS packet
R5:   cur_M2TS_len = Get M2TS len     % Get the length information of
                                        previous MPEG2-TS packet
R6:   M2TS_len = prev_M2TS_len + cur_M2TS_len
                                % Get the total length information from R5 and R6.
    }
R7: STUFF_len=188-M2TS_len
                                % Get the length information of stuffing byte.
U1: Get M2TS                    % Recover a original MPEG2-TS packet
…
OUTPUT:   Original MPEG2-TS packets
```

Fig. 11. Pseudocode for restoring received standard RTP packet and proposed RTP packet to original MPEG2-TS packet format.

computational complexity and algorithms of the proposed transmission scheme with the standard transmission scheme.

Figure 10 shows the pseudocode for converting the received MPEG2-TS packets to standard RTP packets and the proposed RTP packets. In the making of the RTP packets, "U" is a process that is common to both the standard RTP packets and the proposed RTP packets. "S" is the process of making the proposed RTP packets. Thus, the proposed transmission scheme should process S_1 through S_8, which are not needed in the standard transmission scheme.

S_1 is the part that gets the length information of an MPEG2-TS packet, which can be obtained as shown in Fig. 7. S_2 through S_5 are packing processes when TS data needs packing. S_6 checks whether the length of an MPEG2-TS packet is smaller than 188 B. S_7 sets the corresponding bit value of the bitmap header. S_8 inserts information about the length of MPEG2-TS packets that are smaller than 188 B into an RTP packet.

Figure 11 shows the pseudocode that restores the received standard RTP packets and proposed RTP packets to the original MPEG2-TS packet format. "R" is the process of restoring the proposed RTP packets in this paper to the original MPEG2-TS packet format. Thus, the proposed transmission scheme should process R_1 through R_7, which are not needed in the standard transmission scheme.

R_1 and R_3 are the processes that check the bit value of the bitmap header. R_1 checks if the MPEG2-TS packet is smaller than 188 B, while process R_2 gets the length of the corresponding MPEG2-TS packet. R_3 means that the MPEG2-TS packet is packed. R_4 through R_6 seek to find out the length of the packed MPEG2-TS packet. R_7 is the process that finds out the length of the stuffing byte needed to restore original MPEG2-TS packets that are less than or equal to 188 B.

The pseudocode does not include the bitmap headers "00" and "11." Because they are very simple processes that insert an RTP packet or restore the original MPEG2-TS packet from a received RTP packet. Thus, they are not represented in the pseudocode.

The S_1 through S_8 and R_1 through R_7 processes are such simple processes as parse, save, conditional statement, fragmentation, and copy. Thus, the computation complexity of the proposed transmission scheme in this paper is more than that of the standard transmission scheme, but it has a fixed value, which can be represented as $O(1)$. So, the delay or complexity of the RNI and RNI+ is negligible and does not affect the system performance.

IV. Performance Evaluation

In this section, we present the results of experiments conducted to compare the performance of the RTP standard scheme with that of the proposed RTP scheme. Figure 12

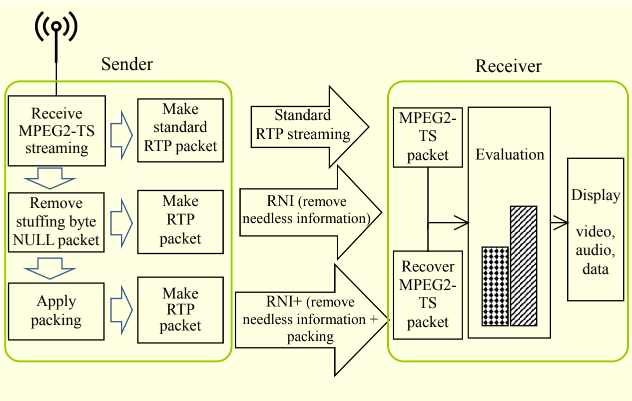

Fig. 12. Logical structure of experimental system.

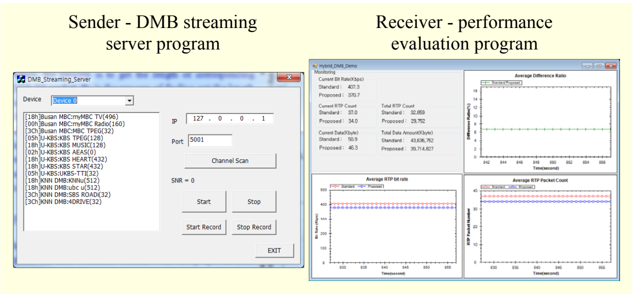

Fig. 13. Screenshots of programs used in our experiment.

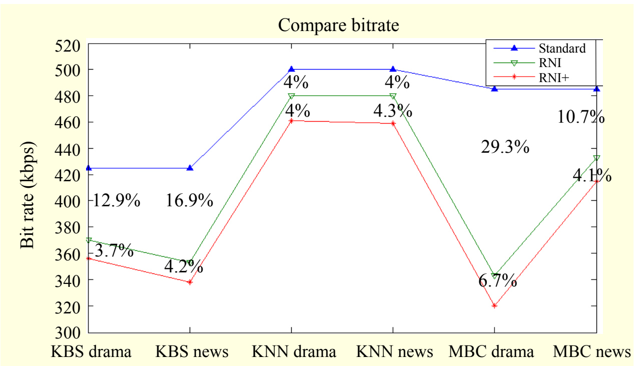

Fig. 14. Standard RTP bitrate vs. proposed RTP bitrate.

shows the logical structure of the experimental system used, while Fig. 13 shows actual screenshots of the programs used in our experiments. The experimental system receives the MPEG2-TS stream in real time and converts it to standard RTP streaming and the RNI and RNI+ RTP streaming, respectively, and they are then transmitted to the receiver.

The payload of a standard RTP packet consists of seven MPEG2-TS packets. The RNI RTP packet payload is made by parsing the MPEG2-TS packet and removing the stuffing bytes and NULL packets. The RNI+ RTP packet payload is made by applying packing to the RNI RTP packet payload. The receiver receives the standard RTP streaming and the RNI and RNI+

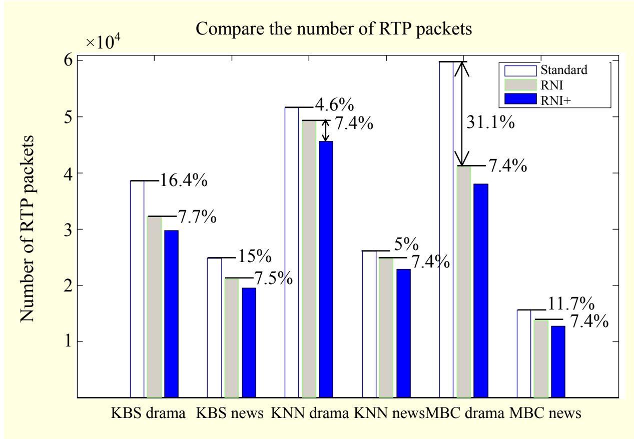

Fig. 15. Number of standard RTP packets vs. number of proposed RTP packets.

Table 2. Average packet length of standard RTP, RNI RTP, and RNI+ RTP.

Broadcasting station		Standard	RNI	RNI+ (RNI + packing)
KBS	Drama	1328	1327.7	1437.8
	News	1328	1331.0	1437.8
KNN	Drama	1328	1331.1	1437.6
	News	1328	1331.0	1437.6
MBC	Drama	1328	1330.1	1437.9
	News	1328	1330.2	1437.8

RTP streaming from the sender and compares the standard RTP streaming with the RNI and RNI+ RTP streaming.

The RTP bitrate and the number of RTP packets of the standard RTP scheme that affects the transmission performance are also compared with that of the RNI and RNI+ RTP schemes.

The RNI transmission scheme described in this paper transmits the RTP stream after removing the stuffing bytes and NULL packets included in MPEG2-TS. This means that the RNI transmission scheme reduces both the amount of data transmitted and the transmission overhead.

The result varies a little according to the component and the character of DMB contents. As shown in Fig. 14, the RNI scheme has a 4% to 29% higher bitrate efficiency than the standard scheme. Further, as shown in Fig. 15, the RNI scheme has 4% to 31% higher efficiency in the number of RTP packets than the standard scheme. This result is almost the same as that of the bitrate efficiency shown in Fig. 14 because the average length of a packet is similar to the length of the standard RTP, as shown in Table 2.

As a result, the RNI scheme is more efficient than the standard scheme in that it reduces the amount of data transmitted; however, the RNI scheme is still inflexible in terms of size. Therefore, we propose an RTP packet payload scheme, RNI+, which applies packing to the RNI scheme. The RNI+ scheme of the RTP packet payload works to reduce the transmission overhead by reducing the number of RTP packets. The reason for reducing the number of packets by using the flexible MTU is that if the same number of packets are transmitted, the longer the length of the packets, the better the throughput is in 802.11[11] and 3G [12].

As shown in Fig. 14, the efficiency of the RNI+ scheme is about 3% to 6% higher than that of the RNI scheme in terms of bitrate, but, as shown in Fig. 15, the efficiency of the RNI+ scheme is about 7% better than that of the RNI scheme in terms of the number of RTP packets. The efficiency of the bitrate of the RNI+ scheme is lower than that of the number of RTP packets. The reason the bitrate of the RNI+ scheme has a lower efficiency than the number of RTP packets is because packing causes the average packet length to get longer. However, the RNI+ scheme is more efficient than the standard scheme because it reduces the number of RTP packets transmitted per second by packing. As a result, the efficiency of the RNI+ scheme is about 8% to 36% higher than that of the standard scheme in terms of RTP bitrate, as shown in Fig. 14, and is about 11% to 38% higher than the standard scheme in terms of the number of RTP packets, as shown in Fig. 15.

V. Conclusion and Future Work

In this study, we examined the problems associated with DMB, which are limited service coverage and insufficient broadcasting bandwidth, and we posited solving these problems through the use of hybrid DMB. In addition, we investigated transmission inefficiency and proposed a transmission scheme to reduce the inefficiency. The transmission efficiency of real services that provide real-time DMB contents streaming was also analyzed.

We plan to study transmission and networks in further detail in the future. We will investigate transmission methods that can improve the efficiency of periodically transmitted data, such as PAT and PMT. In addition, we will examine the relationship between the packet size and the transmission error rate, according to the characteristics of various wireless networks, such as 3G, LTE, and WiBro in DMB receivers.

Reference

[1] ETSI, "Radio Broadcasting Systems: Digital Audio Broadcasting (DAB) to Mobile, Portable and Fixed Receivers," EN 300 401, ver. 1.4.1, June 2006.

[2] J.P. Shim, "Korea's Lead in Mobile Cellular and DMB Phone and Services," *Commun. AIS*, vol. 15, Mar. 2005, pp. 555-566.

[3] ITU-T, "Information Technology — Coding of Audio-Visual Objects — Part 10: Advanced Video Coding," Rec. H.264 | ISO/IEC 14496-10, Mar. 2003.

[4] ISO/IEC, "Information Technology — Coding of Audio-Visual Objects, Part 3: Audio," ISO/IEC 14496-3, Dec. 1999.

[5] W. Joseph et al., "Characterization of Coverage and Indoor Penetration Loss of DVB-H Signal of Indoor Gap Filler in UHF Band," *IEEE Trans. Broadcast.*, vol. 55, no. 3, Sept. 2009, pp. 589-597.

[6] ISO/IEC 14496-1, "Information Technology: Coding of Audio — Visual Object, Part 1: Systems," Nov. 2002.

[7] IETF, "RTP Payload Format for MPEG1/MPEG2 Video," RFC 2250, Jan. 1998.

[8] IETF, "RTP Profile for Audio and Video Conferences with Minimal Control," RFC 3551, July 2003.

[9] ISO/IEC 13818-1:2000, "Information Technology — Generic Coding of Moving Pictures and Associated Audio Information: Systems," Mar. 2002.

[10] L.L. Peterson, *Computer Networks: A Systems Approach*, Morgan Kaufmann, 2007.

[11] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 3, Mar. 2000, pp. 535-547.

[12] M. Jurvansuu et al., "HSDPA Performance in Live Networks," *IEEE Int. Conf. Commun.*, June 2007, pp. 467-471.

[13] IEEE, "IEEE Standard for Information Technology — Telecommunications and Information Exchange between Systems — Specific Requirements — Part 11: Wireless LAN MAC and PHY Specifications," 12 June 2007.

[14] C.K. Kodikara, "Optimal Settings of Maximum Transfer Unit (MTU) for Efficient Wireless Video Communications," *IEE Proc. Commun.*, vol. 152, no. 5, Oct. 2005, pp. 648-654.

[15] ETSI, "Digital Video Broadcasting (DVB); DVB Specification for Data Broadcasting," ETSI EN 301 192, ver. 1.4.2, Apr. 2008.

[16] IETF, "Unidirectional Lightweight Encapsulation (ULE) for Transmission of IP Datagrams over an MPEG-2 Transport Stream (TS)," RFC 4326, Dec. 2005.

[17] T.C. Hong, W.T. Chee, and R. Budiarto, "A Comparison of IP Datagrams Transmission using MPE and ULE over Mpeg-2/DVB Networks," *5th Int. Conf. Inf., Commun., Signal Process.*, Bangkok, Thailand, 2006, pp. 1173-1177.

[18] G. Xilouris et al., "Unidirectional Lightweight Encapsulation: Performance Evaluation and Application Perspectives," *IEEE Trans. Broadcast.*, vol. 52, no. 3, Sept. 2006, pp. 374-380.

[19] B. Collini-Nocker and G. Fairhurst, "ULE versus MPE as an IP over DVB Encapsulation," *Proc. HET-NETs*, 2004, pp. 70-78.

[20] M. Sooriyabandara et al., "A Lightweight Encapsulation Protocol

for IP over MPEG-2 Networks: Design, Implementation and Analysis," *Comput. Netw.*, vol. 48, issue 1, May 2005, pp. 5-19.
[21] J.D. Kim et al., "Design and Implementation of DMBonIP Gateway to Provide Internet Based DMB Service," *IEEE Trans. Consum. Electron.*, vol. 53, no. 1, Feb. 2007, pp. 120-123.
[22] B. Bae et al., "An Efficient and Flexible Hybrid Conditional Access System for Advanced T-DMB," *ETRI J.*, vol. 33, no. 4, Aug. 2011, pp. 629-632.

Hyung-Yoon Seo received his BS in computer information engineering from Hanseo University, Seosan, Rep. of Korea, in 2002 and received his MS in computer science and engineering from Pusan National University, Busan, Rep. of Korea, in 2011. He is now a PhD candidate of computer science and engineering at Pusan National University. His current research interests include mobile multimedia communication systems, MAC and routing protocols from mobile wireless networks, and all-IP-based network convergence.

Byungjun Bae received his BS, MS, and PhD in electronics engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1995, 1997, and 2006, respectively. From 1997 to 2000, he was a researcher at LG Electronics Inc., where he worked on digital signal processing in digital television. Since 2000, he has been with the Broadcasting System Research Department at ETRI, Daejeon, Rep. of Korea. His current research interests include digital signal processing, digital broadcasting protocols, digital multimedia broadcasting system, and interactive broadcasting services.

Jong-Deok Kim received his BS, MS, and PhD in computer engineering from Seoul National University, Seoul, Rep. of Korea, in 1994, 1996, and 2003, respectively. He was with Telcoware Ltd. as a senior researcher and engineer. Since 2004, he has been with the Computer Science and Engineering Department of Pusan National University, Busan, Rep. of Korea, as an assistant professor. His current research interests include mobile multimedia communication systems, MAC and routing protocols for mobile wireless networks, and all-IP-based network convergence.