A Framework for MPEG-4 Contents Delivery over DMB

Bong-Ho Lee, Kyu-Tae Yang, Young Kwon Hahm, Soo In Lee, and Chieteuk Ahn

Digital Multimedia Broadcasting (DMB) is an upcoming standard in Korea used to provide mobile multimedia broadcasting service based on the Eureka-147 Digital Audio Broadcasting (DAB) system. The current dominant multimedia coding standard, MPEG-4, is foreseen to play an important role in forthcoming DMB services. However, the current approaches for transporting MPEG-4 content over DMB networks are not optimized. To address this issue we propose a novel MPEG-4 stream multiplexer, called M4SMux, which provides better stream multiplexing and delivery over DMB networks. M4SMux features an MPEG-4 elementary-stream interleaving mechanism that reduces the multiplexing overhead and a multiplex configuration mechanism that utilizes M4SLinkTable for easy content access. In addition, we propose an error correction method which enhances transport efficiency.

Keywords: MPEG-4, DAB, DMB, delivery framework.

Manuscript received Apr. 17, 2003; revised June 10, 2003.

This work was supported in part by the Ministry of Information and Communication of Korea under the title of "The Development of SmarTV Technology."

Bong-Ho Lee (phone: +82 42 860 1030, email: leebh@etri.re.kr), Kyu-Tae Yang (email: ktyang@etri.re.kr), Young Kwon Hahm (email: ykhahm@etri.re.kr), Soo In Lee (email: silee@etri.re.kr), and Chieteuk Ahn (email: ahnc@etri.re.kr) are with Digital Broadcasting Research Division, ETRI, Daejeon, Korea.

I. Introduction

The Eureka-147 Digital Audio Broadcasting (DAB) system is a novel audio broadcasting system that is intended to supersede existing analogue systems. It is a rugged, power-efficient sound-and-data broadcasting system with a high spectrum. It has been designed for terrestrial and satellite delivery, as well as for hybrid and mixed delivery [1]. Originating from the Eureka-147 DAB system, the Digital Multimedia Broadcasting (DMB) system is designed for mobile TV service, in addition to portable and fixed environments.

Recently, this DMB system has been considered as an valuable solution for mobile multimedia broadcasting service, especially for mobile TV service, in some countries that have adopted terrestrial Advanced Television System Committee (ATSC) systems which do not accommodate for mobile requirements. This trend is due to an orthogonal frequency division multiplexing (OFDM) transmission system with a bandwidth of 1.5MHz combined with a convolutional code for error protection.

Moreover, in conventional digital broadcasting systems, the MPEG-2 system takes care of the synchronization and multiplexing of audio and video data in the digital TV service. This is mainly because the MPEG-2 system has a clear and verified audiovisual synchronization and multiplexing tool called a transport stream (TS). Through the program clock reference mechanism of the MPEG-2 system [2], audio and video elementary streams can be synchronously decoded and presented.

However, when we consider the MPEG-2 system along with MPEG-4 [3]-[5] for inclusion in a DMB system in mobile multimedia broadcasting services, they will likely generate lots of unexpected data due to the duplication of synchronization and the multiplexing process, which we will describe in section II. Additionally, the MPEG-2 system tends to increase the system complexity especially at the end terminal. In that case, the terminal has to be equipped with a full MPEG-2 demultiplxer.

To overcome these disadvantages of the MPEG-2 system and enhance the framework efficiency, we considered the MPEG-4 system for use in the DMB system for the following benefits: a low bit-rate/error-resilient natural codec, scene-based presentation, and a network transparent integration framework. We propose an enhanced delivery framework based on the delivery of multimedia integration framework (DMIF) recommendation using the definition of the M4SMux structure and its linking mechanism.

In the following section, after a brief introduction of the DMB transmission system, we discuss the MPEG-2 system approach [6] and the recommended DMIF framework [7]. In section III, we cover an enhanced framework based on the DMIF recommendation with an introduction of M4SMux and its linking mechanism. We also introduce an error correction scheme applied to M4SMux. Finally, we conclude this paper by showing some evaluation results and planned future work.

II. DMB System Overview

1. Overview of the Eureka-147 DAB System

As shown in Fig. 1, the Eureka-147 DAB system defines two transport mechanisms: the fast information channel (FIC) and the main service channel (MSC). The primary function of the FIC, which is made up of fast information blocks (FIB), is to carry the control information necessary to interpret the configuration of the MSC. The essential part of this control information is the multiplex configuration information (MCI), which contains information on the multiplex structure and its reconfiguration, if necessary. Other types of information that can be included in the FIC are the service information (SI), the conditional access (CA) management information, and the fast information data channel (FIDC). In order to allow a rapid and safe response to the MCI, the FIC is transmitted without time interleaving, but with a high level of protection against transmission errors [1].

For a logical service channel, the MSC is introduced to constitute a multiplex of sub-channels grouped with several integral numbers of capacity units to constitute the basic transport unit. For service components in the MSC, two different transport modes are defined: the stream mode and the packet mode, as outlined in the following sections.

In that aspect, MPEG-4 content can be delivered in the packet mode or stream mode through an appropriate delivery framework, which is outlined in section III.

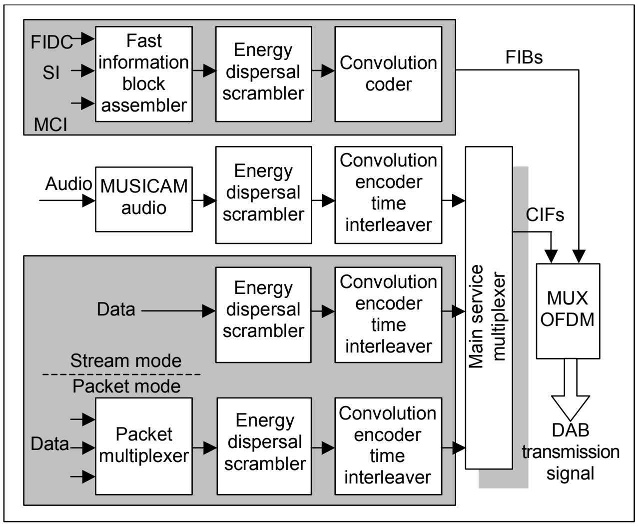

Fig. 1. Eureka-147 DAB system block diagram.

A. Stream Mode

The stream mode is used for applications which require a constant data rate of a multiple of 8 kbps (32 kbps for the B coding profiles). For example, at a sampling rate of 48 kbps, the MPEG layer II audio encoder generates a data frame every 24 ms, which meets this requirement, exactly. When transmitting general data, the data stream can be divided into “logical frames” containing the data corresponding to a time interval of 24 ms. Those logical frames can be transmitted sequentially in the same manner as MPEG audio frames [8].

B. Packet Mode

The packet mode, the most general transport mechanism, is defined for the purpose of conveying several data service components into a single sub-channel. For packet mode transmission, the data is organized into data groups which consist of a header, a data field of up to 8191 bytes and, optionally, a cyclic redundancy check (CRC) for error detection. The data group header allows for the identification of different data group types such that scrambled data and the parameters to access them can be carried in the same packet stream, for instance.

The packets themselves constitute the logical frames in the packet mode, similar to the audio frames in the stream mode. Several packet lengths of 24, 48, 72, and 96 bytes are defined in the Eureka-147 system so as to fit into the DAB structure requiring a multiple of 8 kbps/s. The packets consist of a 5-byte header, a data field, padding if necessary, and a CRC for error detection, as illustrated in Fig 2. This figure shows an example of a packet mode transmission multiplexing the data from two different applications [8].

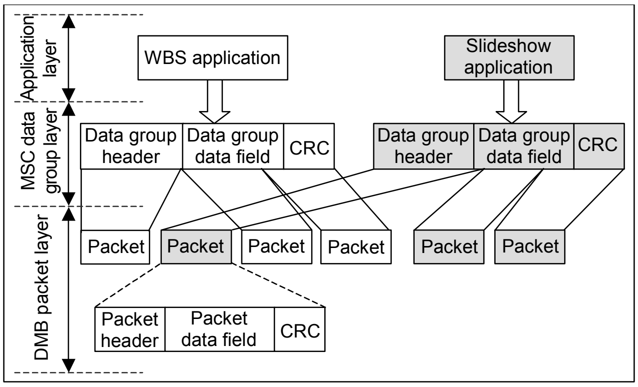

Fig. 2. An example of a packet mode transmission multiplexing two applications.

2. Previous DMB Approach Using an MPEG-2 System

The previous approach, known as the MPEG-4-based DMB system, offers multimedia information with an excellent audio and video quality to mobile environments [6]. In this system, the audio and the video elementary streams resulting from the MPEG-4 encoding stages are wrapped into an MPEG-2 TS at the multiplexing operation. Compared to the conventional Eureka-147 system, this approach improves the error protection with the additional blocks: scrambling, Reed Solomon encoder, and convolutional interleaver, as depicted in Fig. 3.

As can be seen in Fig. 3, this approach does not take into account the MPEG-4 system coping with the media synchronization and the scene-based presentation called binary format for scene (BIFS). Only audio and video elementary streams are considered for mobile TV application. In other words, only audio and video elementary streams are packetized into MPEG-2 TS packets after the program elementary stream (PES) packetization process. Those TS packets are framed into a DMB logical frame at 24 ms, which is to be delivered in stream mode.

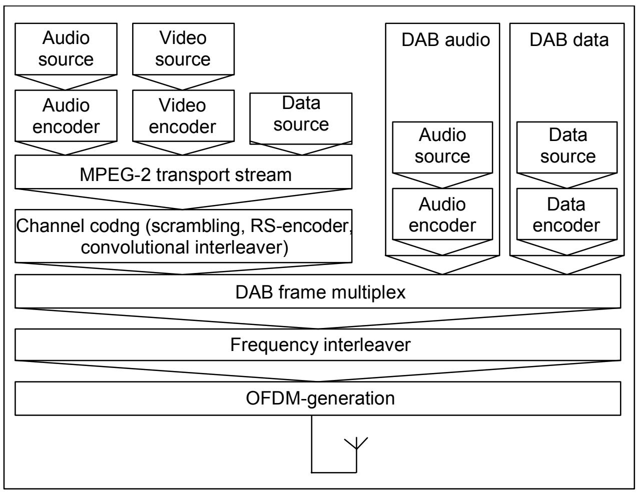

Fig. 3. A block diagram of the precedent DMB approach.

In the meantime, a new DMB approach using the MPEG-4 system is being researched and developed for mobile multimedia broadcasting service in Korea. In this system, audio-visual elementary streams and MPEG-4 system data, called crucial data, are encapsulated into an MPEG-2 TS after the packetization of MPEG-4 sync layer (SL) packets according to the ISO/IEC 13818-1 amendment framework [9]. This framework is highly defined and well organized for delivering MPEG-4 data over a conventional broadcasting network, such as Digital Video Broadcasting or ATSC. Derived from this framework, this new approach tries to deliver the MPEG-2 TS packets encapsulating MPEG-4 data over the DMB network.

As mentioned in section I, this approach reveals two inefficiencies. The first is the duplication of synchronization: MPEG-4 system provides a synchronization tool that incorporates the object clock reference, decoding time stamp, composition time stamp and object time base, which are similar to the program clock reference, decoding time stamp and presentation time stamp of the MPEG-2 system. The second is the duplication of multiplexing: the DMB system also provides a service aggregation scheme known as the ensemble. In order to overcome these inefficiencies, we propose an enhanced delivery framework in the following section.

3. Overview of the DMIF Recommended Framework

It is well known that the system layer and transport layer are transparently separated by the introduction of the DMIF in the MPEG-4 standard. The DMIF plays an important role in the transmission of MPEG-4 contents over underlying physical networks, such as MPEG-2 TS, IP, ATM or DMB. In the DMIF standard, there is an informative delivery framework which allows the transmission of MPEG-4 contents over the DMB network. The objective of this framework is to explore the delivery of MPEG-4 contents along with three main policies: synchronization, multiplexing and linking.

A. Synchronization Policy

Unlike the DMB approach, this framework adopts the MPEG-4 system instead of the MPEG-2 system. The synchronization of the elementary streams is accomplished on the sync layer, in which an elementary stream is mapped into a sequence of packets, called an SL-packetized stream. An SL packet consists of an SL packet header and an SL packet payload. The SL packet header provides the means for continuity checking in case of data loss and carries the coded representations of the time stamps and associated information.

These concepts are specified in [3] on the system decoder model, which models the buffer and timing behavior of an ideal MPEG-4 terminal.

B. Multiplexing Policy

The DMIF-based DMB approach includes a multiplexing scheme. This scheme is used depending on the following content characteristics: priority, transmission repetition, and streaming type. Therefore, the crucial data of the MPEG-4 system, such as the object descriptor (OD) and BIFS data, are wrapped into the DMB packet for the transmission of the packet mode, since these data have high priorities and repetition rates. Whereas, in the case of the audio-visual streams, it is recommended that those streams be delivered in the stream mode separately.

C. Linking Policy

Access to MPEG-4 content may occur in different ways, depending on the application context. A part of this process is standardized by MPEG in an abstract way. This way relies on the availability of an interface that can establish a session context and open and manage transport channels. Such an interface is embodied by the DMIF application interface [10].

In order to enable content access, this framework embodies a kind of linking mechanism between the DMB multiplex and MPEG-4 contents. The initial access information of the initial object descriptor (IOD) is signaled through the DMB MCI. Therefore, the MPEG-4 application instance can reach the IOD at the beginning of service to get the location of the BIFS and OD data. After acquiring the linking information of the BIFS and OD data, the application can access the media streams. Similar to the program specific information table specified in the MPEG-2 system [2], the MPEG-4 Stream Link Table (M4SLinkTable) is also defined in this framework to provide linking information with the DMB multiplexing. Through this table, the elementary stream identifier (ES_ID) is signaled along with the DMB logical channel identifier, in addition to its forward error correction (FEC) type.

This framework also includes and defines not only the linking mechanism but also the framing work of SL packets into logical channels. Here the logical frame is just a container to pack unspecified raw data, as specified in the Eureka-147 system. It also provides a frame synchronization mechanism using the packetStartPointer field at the first two bytes of its frame. The packetStartPointer indicates the address of the first complete SL packet. Besides, according to the service channel allocation scheme, each media stream must be packed into different sub-channels. However, if there are too many media streams to be delivered, managing and operating so many sub-channels can become unwieldy.

Thus, it is necessary to group synchronously related media streams together into one sub-channel for better utilization of the sub-channels in both the transmitter and receiver sides. For such requirements, the use of the multiplex tool FlexMux is recommended in the DMIF approach. However, FlexMux is not as useful in the DMB environment due to the limited capacity of the FlexMux payload, which is 255 bytes. In general, the logical frame constituting a stream mode sub-channel is apt to exceed 255 bytes. Furthermore, even though the FlexMux is available in a single stream case, it might not be applicable to a multiple stream case due to the precondition that only one stream can be mapped into one sub-channel.

III. Proposed Delivery Framework

The DMB approach tends to generate redundancy due to TS encapsulation. In addition, there is no adequate stream-mode framing mechanism or synchronization policy in the previous DMIF recommendation, so the reduction of overhead is the key issue of the proposed delivery framework. The other issue is to enhance the delivery framework based on the DMIF recommendation by the definition of the alternative multiplexer called M4SMux.

1. Enhanced Delivery Framework

The enhancement of the delivery over DMB is mainly focused on two frameworks. One is a stream mode framing that requires a constitution of the logical frame and synchronization. The other is the linking mechanism by which the MPEG-4 application instance can access the DMB logical channel constituted with the M4SMux. Figure 4 shows an enhanced delivery framework.

First, the MPEG-4 crucial data, IOD, OD, and BIFS-Command streams are delivered in the packet mode. Second, the synchronously related MPEG-4 media streams, such as the

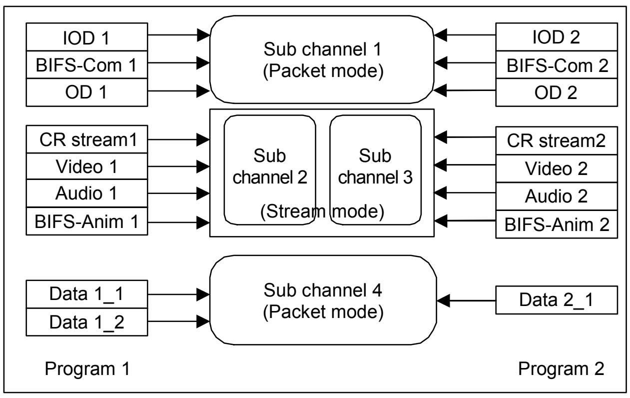

Fig. 4. Enhanced delivery framework concept.

clock reference stream, BIFS-Animation stream, and audio-visual streams are delivered in the stream mode. Third, other object data, which are not synchronously related to the audio-visual streams, are delivered in the packet mode into the crucial data sub-channel or another new sub-channel. The detailed scheme is outlined in the following subsections.

A. MPEG-4 Crucial Data Delivery

Figure 5 shows the MPEG-4 crucial data delivery protocol stack. In a DMB system, the crucial data needs to be transmitted periodically. In particular, the IOD data needs to be transmitted out of band to enable random access to the broadcasting session. In that sense, the IOD, OD, and BIFS command data are encapsulated into each data field as defined in the DMIF recommendation [7] using the syntax shown below. Furthermore, it is necessary for these data to be strongly protected. For the packet mode transmission, the encapsulated data field is packetized into an MSC data group compliant with the Eureka-147 system. In the case of BIFS data, only the encapsulation of the BIFS-Command stream is considered, since the BIFS-Animation stream is another elementary stream. In terms of the BIFS update handling, in the DAB packet mode the MSC data group packet has the update mechanism in the version field. So, the update can be automatically triggered by checking the version field.

- Syntax of the IOD-DataField

```
class InitialObjectDescriptorDataField {
        bit(5)              version_number;
        M4SLinkTable M4SLinkTbl;
        InitialObjectDescriptor    initialObjDescr;
}
```

- Syntax of the OD-DataField

```
class ObjectDescriptorStreamDataField {
      bit(5)                  version_number;
      bit(1)                  current_next_indicator;
      bit(8)                  data_field_number;
      bit(8)                  last_data_field_number;
      M4SlinkTable       M4SLinkTbl[[data_field_number]];
      bit(13)                 SL_packet_length;
      SL_ packet           objectDescriptorStream
                                  Packet[[data_field_number]];
}
```

- Syntax of the BIFSCommand-DataField

```
class BIFSCommandDataField {
      bit(5)               version_number;
      bit(1)               current_next_indicator;
      bit(8)               data_field_number;
      bit(8)               last_data_field_number;
      const bit          reserved = 0b1111.1;
      bit(13)             SL_packet_length;
      SL_packet        BIFSCommandFrame
                           Packet[[data_field_number]];
}
```

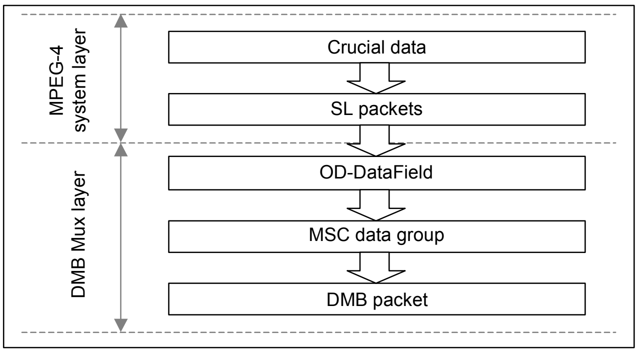

Fig. 5. Crucial data delivery protocol stack.

B. Delivery of MPEG-4 Media Streams

Figure 6 shows the delivery protocol stack of the MPEG-4 media streams. In this work, the clock reference, the BIFS-Animation, and the typical MPEG-4 audiovisual streams are packed into the logical frame, M4SMux, prior to transmission into the stream mode. Each of the SL packets is multiplexed into the M4SMux. Unlike the DMIF recommendation, in this framework the packet start pointer is not used. A more detailed framing scheme for the frame synchronization is described in section III. 2.

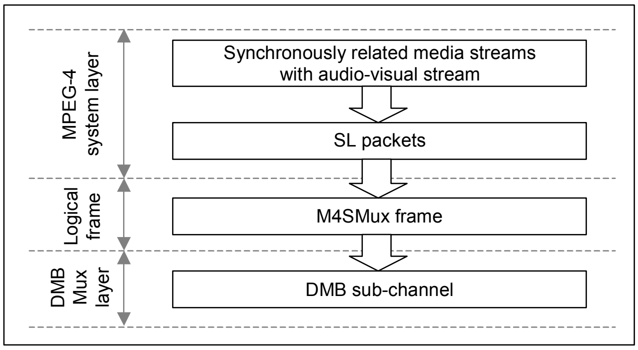

Fig. 6. Media streams delivery protocol stack.

C. Linking of the MPEG-4 Streams

In order to provide a linking mechanism between the ES_ID and DMB multiplex, it is necessary to define the M4SLinkTable structure as the following syntax.

```
class M4SLinkTable( ) {
      const bit(6)    reserved = 0b1111.11;
```

```
    bit(8)              stream_number;
    for(i=0; i<stream_number; i++) {
        bit(6)              ES_ID;
        bit(1)              M4SMuxFlag;
        bit(1)              FEC_flag;
        bit(1)              Inerleaver_Flag;
        if(M4SMuxFlag)
                SubChId;
        else
                SCId;
        if(FEC_Flag)
        bit(8)      FEC_type;
    }
}
```

This link table links the MPEG-4 streams and the DMB logical channel. It signals the number of streams, its FEC type along with the linking information of the ES_ID, and the DMB logical sub-channel identifier or service component identifier. Hence, if synchronously related media streams are encapsulated into M4SMux, each ES_ID of a different stream must have the same sub-channel identifier number. Furthermore, the grouped media streams must be defined and identified as a single service component type in the ExtDSCTy field of the DMB MCI.

The signaling of the MPEG-4 media streams transmitted in the different sub-channels is given by means of the MCI. According to the recommendation [7], all MPEG-4 service components are identified by the same general entry, for example “MPEG-4” in the ExtDSCTy field. In the meantime, the sub-channel of the service component that carries the IOD-DataField is signaled as a primary component in the FIG 0/2 service structure information of the MCI field.

If the MPEG-4 application is installed in the DMB receiver terminal initiated to render the scene, the MPEG-4 application context needs to get an IOD stream, first, through the DMB MCI. In other words, the initial access information of the logical sub-channel which carries the IOD stream is indicated by the DMB MCI. Figure 7 illustrates an example of the

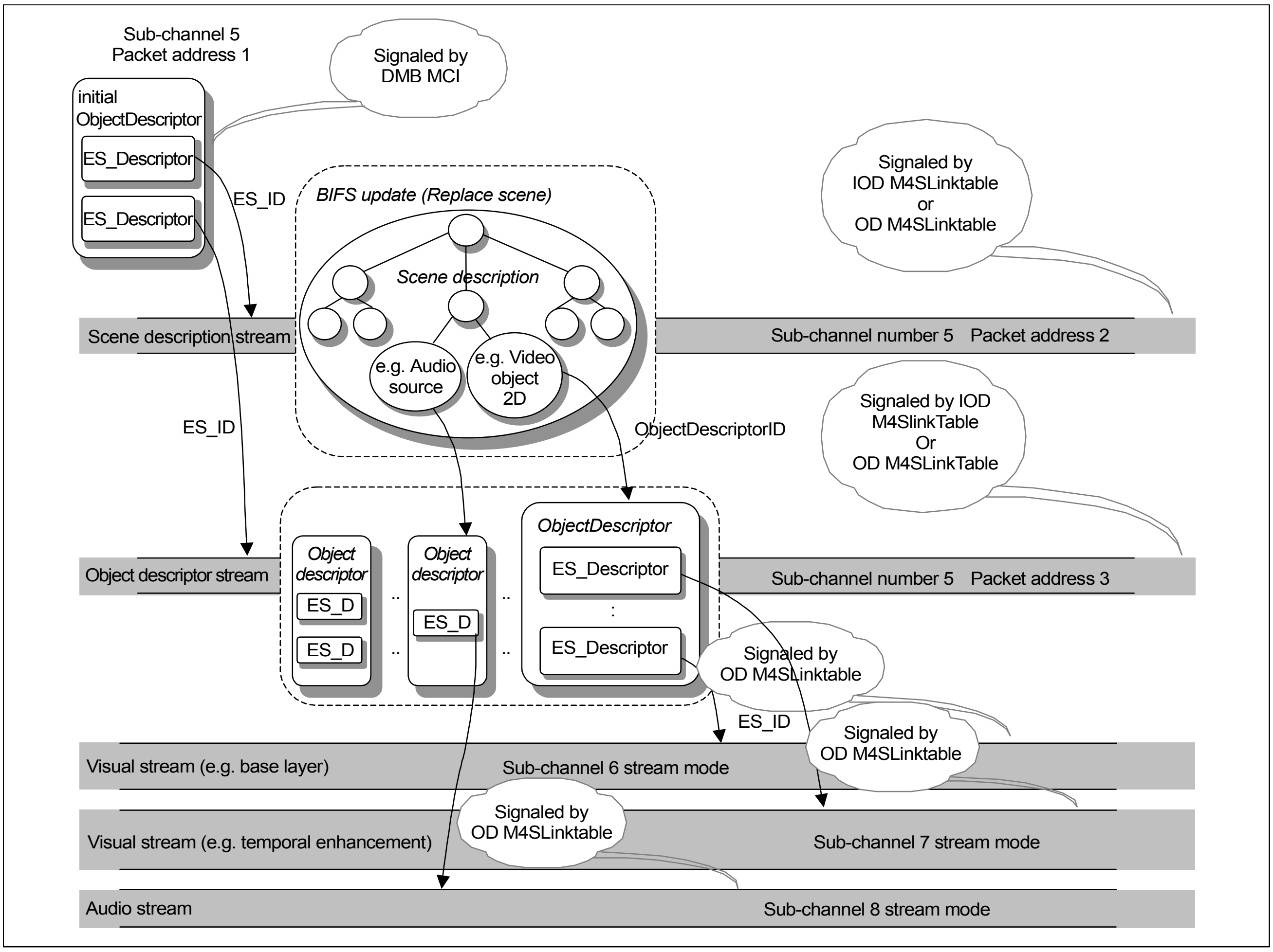

Fig. 7. An example of a linking mechanism.

proposed linking mechanism and details the following steps.

- The initial session, called IOD, between the broadcast server and the client is opened out of band through the DMB MCI.
- The ES descriptors in this IOD identify the primary OD stream and BIFS stream for the service.
- The client selects the ES_IDs of those OD and BIFS streams that are required.
- The client links between the ES_ID and DMB logical sub-channel and service component according to the M4SLinkTable provided in the IOD-Data field.
- The client receives the OD and BIFS data.
- The client parses the received OD and BIFS data.
- The client selects an ES_ID and links it to a logical DMB sub-channel and service component according to the M4SLinkTable provided in the OD-Data field.
- The client receives the elementary streams and decodes them.

2. M4SMux Framing

As shown in Fig. 8, the M4SMux is a soft multiplexer similar to the MPEG-4 FlexMux. It was designed for the stream mode transmission of MPEG-4 content streams mapped into a single sub-channel. Audio-visual SL packets are encapsulated into the M4SMux. This frame consists of the header and payload. There is information about random access, the number and type of the SL packets, and the length of the payload in the header. Actual SL packets are located in the payload.

The random-access flag signals whether the random access SL packet exists in the M4SMux. The configuration field gives information about the SL packet length and its type, in addition to the packet order, because the SL packet header does not provide size and order information. So the actual SL packets are multiplexed into the payload according to the configuration information. Similar to the MuxCode mode of the FlexMux tool, the MPEG-4 stream channel (M4SC) is defined to configure a variety of SL packets.

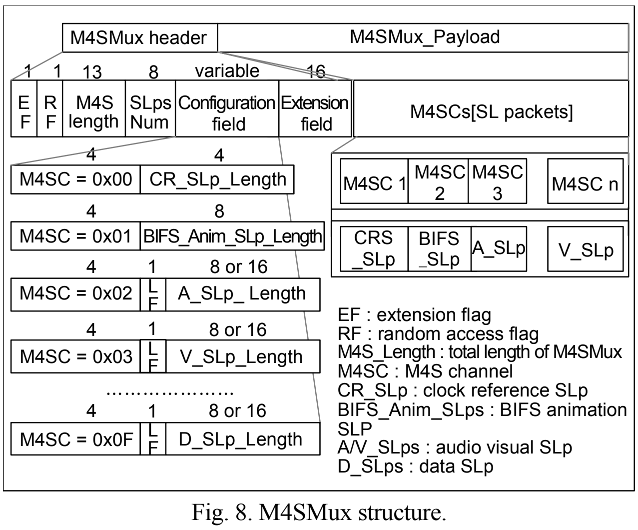

Fig. 8. M4SMux structure.

A. Syntax

```
M4SMux {
   bit(1)          extension_field_flag;
   bit(1)          random_access_flag;
   bit(13)         D4SMux_length;
   bit(8)          SLps_number;
  if(M4SC == "0x00")
         bit(4)   clock_reference_length;
  else if(M4SC =="0x0 1")
         bit(8)   BIFS_Anim_SLp_length;
  else if(M4SC =="0x0 2"){
         bit(1)   LF(length_flag);
         if(LF)
                  bit(16) audio_SLp_length;
         else
                  bit(8) audio_SLp_length;
  }
  else if(M4SC =="0x03"){
         bit(1)   length_flag;
         if(LF)
                  bit(16) video_SLp_length;
         else
                  bit(8) video_SLp_length;
  }
  .........
  else if(M4SC =="0x0F") {
         bit(1)   length_flag;
         if(LF)
                  bit(16) video_SLp_length;
         else
                  bit(8) video_SLp_length;
  }
         if(extension_field_flag)
                  bit(16)      extension_field;
         M4SMux_payload;
}
```

B. Semantics

Extension Flag indicates the presence of the extension field.

Random_Access_Flag indicates the presence of random access VOP.

M4SMux_length indicates the total length of the M4SMux

SLps_num indicates the total number of SL packets.

M4SC indicates a channel type identical to the SL packet.

M4SMux_payload is a container which carries SL packets.

LF is a 1-bit length flag which indicates that " 0 " means the length byte is 1, and that "1" means the length byte is 2.

Extension field contains the conditional access information.

C. Framing Scheme of SL packets

In order to de-multiplex the SL packets efficiently at the end terminal, it is necessary for us to frame and synchronize the SL packets to the M4SMux. When one or more SL packets do not fit into the size of the M4SMux payload, a stuffing scheme is required. If the last SL packet is too large to be loaded into the frame, the packet must be fragmented into smaller packets, as defined in [3].

In that case, the fragmented SL packet has a small header, which contains only the fragmentation information. Consequently, it is necessary for the first SL packet to always be byte-aligned for precise frame synchronization. Figure 9 illustrates a scheme of SL packet fragmentation and byte alignment.

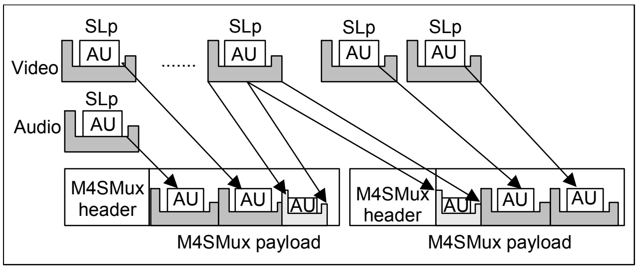

Fig. 9. The framing scheme of SL packets.

3. Error Correction Code

If we want to deliver MPEG-4 data over a DMB channel, we need to consider the outer coder, called the error correction code, against error-prone environments, because the DMB channel has various types of noise, distortion, and interference. Therefore, the demand for an error-correction code arises from the massive amount of data processed in the DMB channel. In that sense, in order to promote the performance of the receiver, we chose the Reed Solomon (RS) code and convolutional interleaver. In our proposed framework, the RS code and convolutional interleaver are applied to the logical frame for the stream mode transmission. As depicted in Fig. 10, the

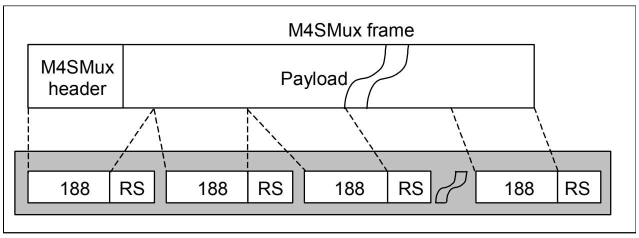

Fig. 10. Outer coder structure.

Table 1. Comparison of the overhead.

Framework	384kbps	512kbps
New DMB approach	7.9%	8.2%
Proposed	3.81%	3.9%

inputed M4SMux is sequentially divided into 188 bytes to the outer coder. After adding the 16-byte RS code, 204-byte packets are interleaved by a convolutional interleaver.

IV. Evaluations

We implemented the proposed framework using IM1 reference software along with a simple-profile video codec and an AAC audio codec targeting 384 kbps and 512 kbps bit rates. We evaluated two performances. One is the overhead comparison between our proposed framework and the new DMB approach, described in section II.2. The other is the performance of the error correction code between the applied framework of the outer code and the bare framework.

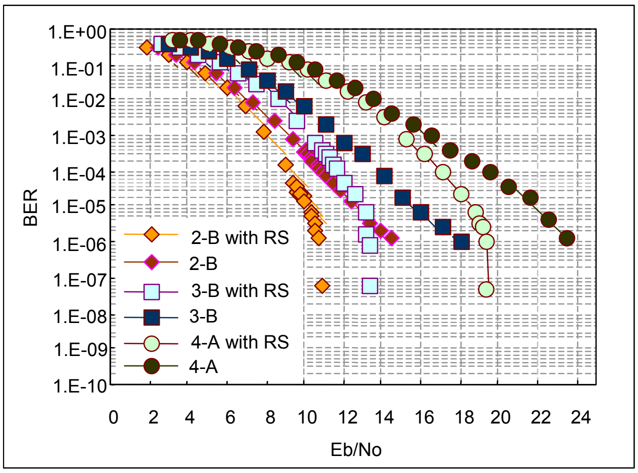

Fig. 11. Performance results of the error-correction code.

First, under the environments described in the pervious sections, we compared the overhead of the proposed framework against its predecessor, the new DMB approach using the MPEG-2 TS. In order to evaluate the latter case, we repeatedly transmitted only the Program Association Table and Program Map Table at approximately 500 ms. In addition, the TS packets were packed into logical frames per every 24 ms unit by padding some stuffing bytes. According to ISO/IEC 13818-1 amendment 7 [9], the crucial data of the MPEG-4 system were also encapsulated into the TS.

In the former case, the evaluation test was carried out for the proposed framework. System crucial data fields such as IOD-DataField, OD-DataField and BIFSCommand-DataField including the M4SLinkTable, were repeatedly transmitted in the packet mode within one sub-channel every 500ms.

Table 1 shows the evaluation results of the overhead. The percentage indicates the generated overhead bits against the target bit-rates. The overhead of the proposed framework is lower than that of the latter case by about two times because the redundancy of synchronization and the multiplex process are removed by the proposed framework.

Next, to evaluate the performance of the error-correction code, we used simulated bit error ratio (BER) performances of the DMB system, comparing RS coding and no RS coding to the proposed bare framework, M4SMux. Figure 11 shows the simulation results of the proposed framework.

The maximum Doppler shift is 13.3 Hz and the protection rates are 4/7(2-B), 2/3(3-B), and 3/4(4-A). Without an RS coder, there was a degradation of 2-B, 3-B, and 4-A in Eb/No of 15 dB, 18 dB, and 24 dB, respectively, at a BER of 10^{-6}. A degradation of 2-B, 3-B, and 4-A with an RS coder reached approximately 11 dB, 13 dB, and 19 dB, respectively, at the same BER value. Therefore, the gain of the RS coder was approximately 4 to 5 dB. These results show that to enhance performance with a reasonable Eb/No of 11 to 20 dB at a BER of 10^{-6}, RS coding can be an efficient channel scheme for the proposed delivery framework.

V. Conclusions

We proposed an enhanced delivery framework, which is based on the DMIF recommendation for the delivery of MPEG-4 contents over a DMB system. We made a detailed description of the delivery framework to deliver MPEG-4 content over DMB. We also outlined the evaluation results of the overhead comparison between the proposed framework and the previous technique. Additionally, we evaluated the performance of the error-correction code as applied to the M4SMux. Using the proposed framework, we reduced the overhead of the new DMB approach. Also, we found that the error-correction code showed a better performance. In conclusion, there is no doubt that the proposed framework is more efficient for delivering MPEG-4 content over DMB for mobile multimedia broadcasting services, such as a mobile TV service, than the previous approach. As a future work, we plan to implement the DMIF stack compliant to the DMIF standard using the proposed framework.

Acknowledgments

We would like to thank the anonymous reviewers whose comments improved this paper.

References

[1] ETSI EN 300 401, *Radio Broadcasting Systems; Digital Audio Broadcasting (DAB) to Mobile, Portable and Fixed Receivers*, May 2001.

[2] ISO/IEC 13818-1, *Information Technology–Generic Coding of Moving Pictures and Associated Audio Information: Systems*, Dec. 1997.

[3] ISO/IEC 14496-1, *Information Technology–Coding of Audio-Visual Objects-Part 1: Systems*, Dec. 1999.

[4] ISO/IEC 14496-2, *Information Technology–Coding of Audio-Visual Objects-Part 2: Visual*, Dec. 1999.

[5] ISO/IEC 14496-3, *Information Technology–Coding of Audio-Visual Objects-Part 3: Audio*, Dec. 1999.

[6] Marco Grube, Peter Siepen, Christian Mittendorf, Marco Boltz, and Mayur Srinivasan, "Application of MPEG-4: Digital Multimedia Broadcasting," *IEEE Trans. Consumer Electronics*, vol. 47, no. 3, Aug. 2001, pp. 474-484.

[7] ISO/IEC 14496-6, *Information Technology–Coding of Audio-Visual Objects-Part 6: Delivery of Multimedia Integration Framework*, Dec. 1999.

[8] Wolfgang Hoeg and Thomas Lauterbach, *Digital Audio Broadcasting Principles and Applications*, John Wiley & Sons, 2000, pp. 36-46.

[9] ISO/IEC 13818-1 FDAM 7, *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information: Systems – Amendment 7: Transport of 14496 Data over ISO/IEC 13818-1*, June 2000.

[10] Carsten Herpel, "Elementary Stream Management in MPEG-4," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 2, Mar. 1999, pp. 315-324.

Bong-Ho Lee received the BS and MS degrees in avionics engineering from Hankuk Aviation University, Goyang-city, Gyeonggi-do, Korea, in 1997 and 1999. In 1999, he joined Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, where he has been working on digital video broadcasting system technologies. Currently as a senior engineer, he is in the Mobile Multimedia Broadcasting Research Team of Broadcasting System Research Group. His main research interests include mobile multimedia broadcasting systems, digital video broadcasting systems, and multimedia coding and applications.

Kyu-Tae Yang received the BS and MS degrees in electronic engineering from Kyungpook National University, Daegu, Korea, in 1986 and 1991. In 1991, he joined ETRI where he is currently a senior engineer in Broadcasting System Research Group. His current research interests include voice driven data broadcasting services in digital multimedia broadcasting (DMB).

Young Kwon Hahm received the BS, MS, and PhD degrees in electronic engineering from Yonsei University, Seoul, Korea, in 1980, 1982, and 1996. In 1984, he joined ETRI. He is currently the team leader of the Mobile Multimedia Broadcasting Research Team. His current research interests include digital mobile communication and digital broadcasting.

Soo In Lee received the MS and PhD degrees, all in electronics engineering from Kyungpook National University, Daegu, Korea, in 1989 and 1996. In 1990, he joined Electronics and Telecommunications Research Institute (ETRI), Daejeon, Korea, where he has been working on broadcasting system technologies. Currently he serves as the Director for Broadcasting System Research Group. His research interests include terrestrial DTV and DMB systems, digital CATV systems, and 3DTV transmission systems.

Chieteuk Ahn is a vice-president and a principal member of technical staff in the Digital Broadcasting Research Division of ETRI, Korea. He received the Bs and MS from Seoul National University, Korea, in Feb., 1980 and 1982, and the PhD from the University of Florida, USA, in Aug. 1991. Since he joined ETRI in 1982, he has been involved in developing digital switching systems, MPEG standardization and broadcasting technology. His recent work has been focused on MPEG technology as well as on developing interactive multimedia technology. He has served as an HOD of MPEG-Korea and SC29-Korea since 1996. His main interests are in the areas of multimedia signal processing, broadcasting, and communications.