Development of Interactive Data Broadcasting System Compliant with ATSC Standards

Jong Myeon Jeong, Yong Ju Lee, Min Sik Park, Ji Hoon Choi, Jin Soo Choi, and Jinwoong Kim

In this paper, we present an interactive data broadcasting system compliant with the Advanced Television Systems Committee (ATSC) standards. The proposed system provides users not only with various data broadcasting services but also remote interactive services. For various data broadcasting services, we have adopted a synchronized data injector that calculates the transmission time of synchronized data accurately and multiplexes synchronized data with the data of an MPEG-2 audio-visual program according to the calculated transmission time. To support remote interactive services, we designed and implemented a return channel server connected on a bi-directional interaction channel. Test results show that the proposed system provides both an asynchronous and synchronized data broadcasting service and remote interactive service appropriately.

Keywords: Data broadcasting, interactive broadcasting, synchronized data service, remote interactive service, data injector, return channel server.

Manuscript received Apr. 9, 2003; revised Nov. 17, 2003.

Jong Myeon Jeong (phone: +82 61 240 7314, email: jmjeong@mmu.ac.kr) was with ETRI, Daejeon, Korea and is currently with the Division of Marine Electronics and Communication Engineering, Mokpo National Maritime University, Mokpo, Korea.

Yong Ju Lee (email: draball@etri.re.kr), Min Sik Park (email: pms@etri.re.kr), Ji Hoon Choi (email: cjh@etri.re.kr), Jin Soo Choi (email: jschoi@etri.re.kr), and Jinwoong Kim (email: jwkim@etri.re.kr) are with Digital Broadcasting Research Division, ETRI, Daejeon, Korea.

I. Introduction

With the emergence of the digital era, interactive data broadcasting has come to play an important role as a mass media in providing users with various types of information. Interactive data broadcasting does not only mean going from analog to digital transmission, but also allowing users interactivity with multi-media data delivered through a broadcast channel.

Interactive data broadcasting can be classified into local interactive service and remote interactive service [1]. In local interactive service, data elements are stored into a local storage of receivers, and users can interact with the receiver using the data element stored in local storage. Also, interactive data broadcasting can provide both programs-related data, such as a drama synopsis; sports player profiles; and song lyrics; and unrelated data, such as weather forecasts; traffic information, and stock information. The concept of local interactive service is shown in Fig. 1(a).

Remote interactive services enable users to access a remote server to obtain enhanced information which is related or unrelated to the broadcasting program, as shown in Fig. 1(b). Furthermore, users can participate in the broadcasting program by interacting with the remote server. Examples of remote interactive service include a bi-directional quiz show, electronic commerce, e-mail, and so on.

Most existing interactive data broadcasting systems are concerned with local interactive service and, hence, there are little results for remote interactive service [2].

Data service can be classified into asynchronous, synchronous, and synchronized data according to their temporal constraints for decoding and presentation [3]. In order to provide users with various interactive data broadcasting services, it is desirable that the broadcasting system be able to transmit any type of data service mentioned above.

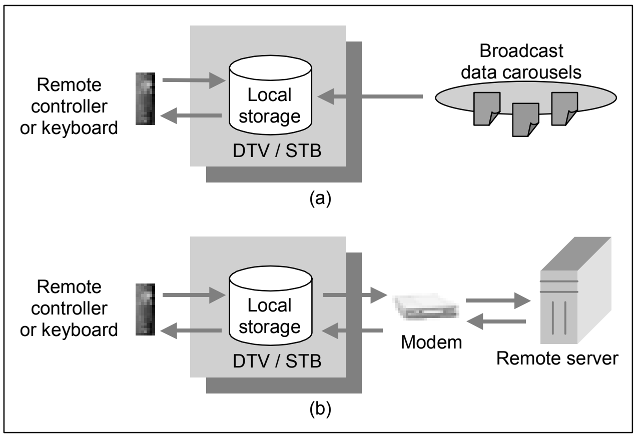


Fig. 1. Classification of interactive services: (a) local interactivity, (b) remote interactivity.

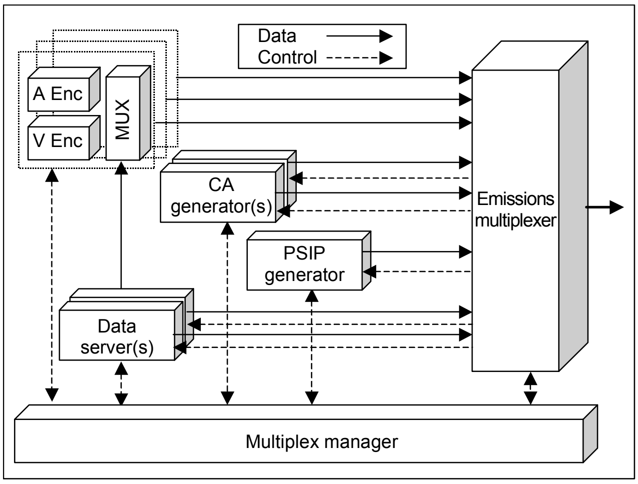


Fig. 2. DIWG data broadcasting system.

Through the National Association of Broadcasters 2003 exhibition [4] and the 2002 World Cup data-broadcasting trial service [5], a system model for local interactive service has been developed. However, most existing interactive data broadcasting systems have supported the mechanism to deliver only asynchronous data [2].

In this paper, we describe the structure and operation method of an interactive data broadcasting system supporting both local and remote interactive services compliant with the Advanced Television Systems Committee (ATSC) data standards [1], [3], [6]-[12]. To do this, we propose methods for the transmission of synchronized data and implement a synchronized data injector by using our proposed method. Also, we propose a method for constructing a return channel server to provide a user participation service. Some of the proposed techniques make use of extensions of the current ATSC standards in an interoperable manner.

This paper is organized as follows. In section II, we explain the overview of an interactive data broadcasting system. In section III, we describe the synchronized data injector responsible for synchronized data transmission. Then, the return channel server to provide a user participation service is described in section IV, followed by the experimental results of this work in section V. Finally, we summarize and conclude this paper in section VI.

II. Overview of Proposed Interactive Data Broadcasting System

The Data Interface Working Group (DIWG), an ad hoc group of the ATSC implementation subcommittee, provides implementation recommendations [6] on data broadcasting systems such as a system reference model and connection interface. Figure 2 illustrates DIWG's data broadcasting system including several function modules.

The functions of each module are explained as follows.

- Audio encoder (A Enc): The audio encoder reads a stream of input audio samples and generates a coded bit-stream as defined by ISO/IEC 13818-1 [7] and ATSC A/53 [9].
- Video encoder (V Enc): The video encoder reads a stream of input video samples and generates a coded bit-stream as defined by ISO/IEC 13818-2 [8] and ATSC A/53 [9].
- Program multiplexer (MUX): The program multiplexer combines a number of elementary streams into a single program and outputs an MPEG-2 transport stream (TS).
- Conditional access (CA) generator: The conditional access generator provides entitlement control messages and entitlement messages to the emission multiplexer in the form of MPEG-2 transport stream packets. This module is optional.
- Program and system information protocol (PSIP) generator: The PSIP generator provides the program and system information necessary for the multiplexer to generate an MPEG-2 multi-program transport stream compliant with ATSC.
- Data Server: The data server provides coded data using the protocols for any input data as defined in ATSC A/90 [3]. The output of the data server is a formatted MPEG-2 transport stream.
- Emission multiplexer: The emission multiplexer inserts MPEG-2 transport stream packets and extracts MPEG-2 transport stream packets from one or more MPEG-2 transport streams.
- Multiplex manager: The multiplex manager controls each module normally for the data broadcasting service.

DIWG recommends the data broadcasting system be built readily by using existing hardware and software. But conventional program multiplexers don't have the function to multiplex data elements into an AV program. Therefore the existing program multiplexer should be replaced by a new program multiplexer with an insertion function as defined by [6].

In order to revive the existing program multiplexer, we supplement a synchronized data injector to DIWG's data broadcasting system. The role of the synchronized data injector is to multiplex synchronized data with MPEG-2 AV TS transmitted from the program multiplexer. By using the synchronized data injector, we can install an interactive data broadcasting system without replacing any existing components.

On the other hand, DIWG's data broadcasting system can not provide remote interactive service because it does not consider the bi-directional interaction channel, therefore it needs the return channel server to be capable of communicating with the terminal for remote interactive service. An interactive data broadcasting system should include a return channel server which responds to a user's requests through a return channel.

We propose an interactive data broadcasting system including a synchronized data injector and return channel server, which are shown in Fig. 3. The synchronized data injector is responsible for inserting synchronized data into an AV program and the return channel server provides remote interactive service. The synchronized data injector is designed to be fault-tolerant in order to be used as one component in a real DTV station.

The proposed interactive data broadcasting system consists of an AV encoder (or TS generator), a contents file server, data server, PSIP generator, re-multiplexer, synchronized data injector, return channel server and multiplex manager. The contents file server provides the data element, and the MPEG-2

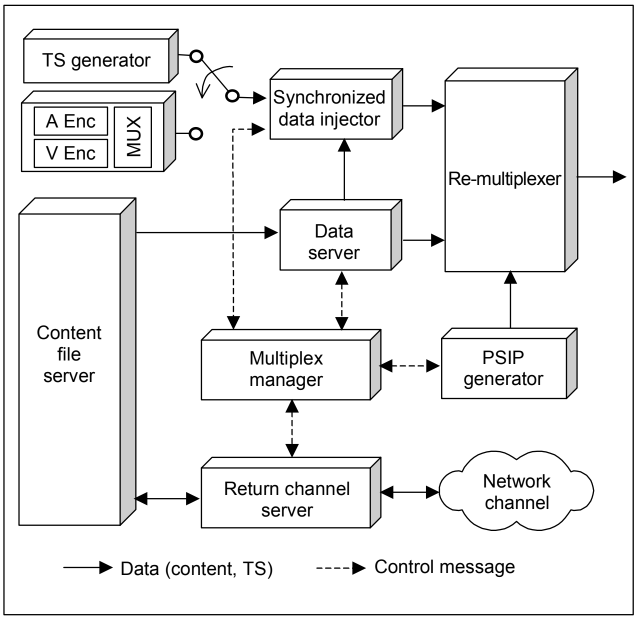


Fig. 3. Proposed interactive data broadcasting system.

AV transport stream is delivered to the synchronized data injector. The role of the synchronized data injector is for multiplexing AV with synchronized data instead of using a program multiplexer. By adding a synchronized data injector into a digital broadcasting system, there is no necessity for replacing the conventional program multiplexer.

The return channel server collects a user's data sent from the receivers through a bi-directional interaction channel, then creates contents reflecting the user's data and delivers those contents to the content file server so that the contents reflecting the user's data are transmitted into a receiver via a broadcast channel by the interactive data broadcasting system. As Fig. 4 illustrates, the operational flow of an interactive data broadcasting system is as follows.

Step 1. The PSIP generator generates a PSIP, PSI (program

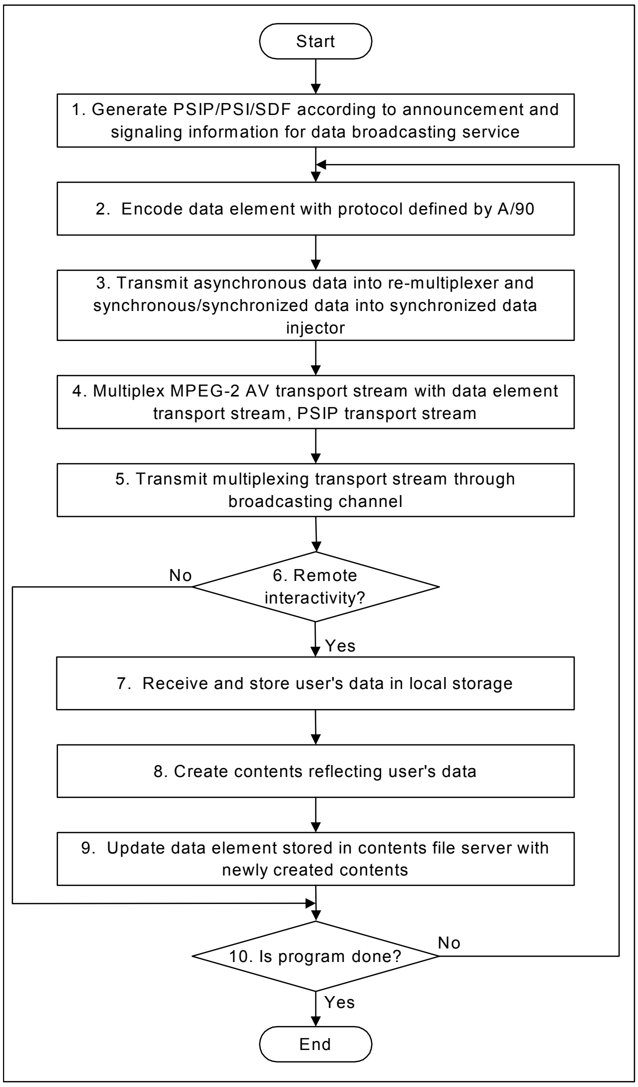


Fig. 4. Operation flow of a proposed interactive data broadcasting system.

specific information), and a SDF (service description framework) according to the announcement and signaling information [14] for data broadcasting services.

Step 2. The data server encodes the data element with a protocol defined by ATSC A/90 [3]. The contents file server feeds the data element into the data server.

Step 3. If the data element is made up of asynchronous data, The data server transmits the data element into the re-multiplexer. Otherwise, the data server transmits it into the synchronized data injector.

Step 4. The synchronized data injector multiplexes the MPEG-2 AV transport stream with a synchronized data transport stream from the data server. And, the re-multiplexer combines the MPEG-2 transport streams from the synchronized data injector, data server, and PSIP generator into an MPEG-2 program TS.

Step 5. The interactive broadcasting system transmits a multiplexing TS to receivers through a broadcasting channel.

Step 6. If the broadcasting program which is on the air is a remote interactive service, continue to step 7. Otherwise, skip to step 10.

Step 7. The return channel server receives and stores the user's data in local storage.

Step 8. The return channel server creates contents reflecting the user's data in real-time according to service specific requirements.

Step 9. The return channel server updates the data element stored in the contents file server with newly created contents.

Step 10. If the termination condition is satisfied, the broadcasting program is terminated. Otherwise, return to step 2.

III. Synchronized Data Injector

Data elements can be classified in three types: asynchronous data, synchronous data, and synchronized data. Asynchronous data is not associated with any transmitted clock reference. Synchronous and synchronized data use an MPEG-2 program clock reference (PCR) and an MPEG-2 presentation time stamp (PTS) with the objective of delivery or presentation of its own data unit [3].

The PTS, which is the time information for a presentation, makes it possible for synchronized data to be presented in the desired time for synchronization with a video or audio frame. It is important that synchronous and synchronized data are transmitted in the timeframe necessary for accurate transmission. An inaccurate delivery of synchronized data may bring a buffer overflow or delayed presentation of data in the system target decoder (STD). For this reason, it is important to calculate the accurate time for the transmission of the data element in order to provide synchronized data service. The synchronized data injector calculates the accurate time for the transmission of synchronized data, and then multiplexes the synchronized data with the MPEG-2 AV TS.

In this section, we propose a method to calculate the accurate transmission time of synchronized data. The structure and operation of a synchronized data injector will be also described.

1. Transmission Time of Synchronized Data

Synchronized data can be classified as streaming data and non-streaming data according to the continuity of presentation. Non-streaming data has just one PTS, and appears only one time. Streaming data has several data access units (DAUs), that have their own PTS or decoding time stamp (DTS), and each DAU is presented sequentially. For this reason, we should consider the transmission of synchronized data as that of streaming data and non-streaming data.

A. Transmission Time of Synchronized Non-streaming Data

To present synchronized non-streaming data in accurate time, a data broadcasting system should transmit the data element before the 90 kHz portion of the system time clock (STC) of the STD is equal to its PTS, since synchronized non-streaming data have only one PTS. But, if it is transmitted too early, it may enlarge the minimum channel maintenance time (MCMT) which is the minimum time that viewers have to hold on a channel for data to be presented in the receiver. The MCMT is the time between the moment when the first bit of data enters the receiver and the moment when the data are presented in the receiver. Figure 5 shows the relationship between the transmission start-time and the MCMT in a receiver.

In the figure, case 1 shows that synchronized data was delivered to the receiver from T1, and its receiving and decoding were ended at T2. In case 2, synchronized data, which was transmitted later than in case 1, was delivered to the receiver from T4, and its receiving and decoding ended at T5. The MCMT of case 1 is larger than in case 2, because T1 is earlier than T4 from the time the

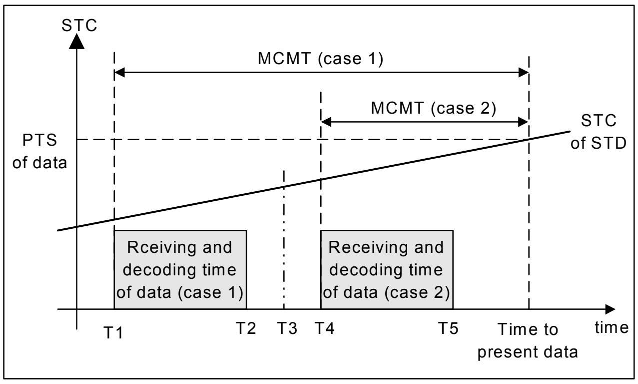


Fig. 5. The relationship between transmission start-time and MCMT.

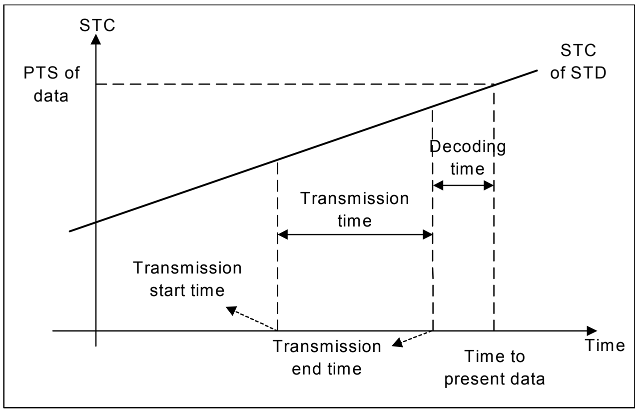


Fig. 6. Transmission time of synchronized non-streaming data.

data is presented. In case 1, viewers who were watching the channel from T3 could not see the data being presented because not all of the data was received. But in case 2, the above viewers could see the data because the receiver received all the data. In the case of channel changing, as the MCMT grows larger, the chance of data being presented in the receiver becomes less. For this reason, non-streaming data have to be delivered as closely as possible to the time when the 90 kHz portion of the STC becomes equal to the PTS in the receiver.

The time period between the transmission of data and their presentation is the sum of the transmission time and data decoding time [13]. Figure 6 indicates the ideal transmission time of non-streaming data. From this, we can calculate the transmission start-time using

$$T_{st} = PTS_d - D_t - T_t, \quad (1)$$

where T_{st} is the transmission start-time, T_t is the transmission time, PTS_d is the PTS of the data, D_t is the decoding time, and where T_{st}, T_t and D_t have the same dimension of the 90 kHz portion of the PCR.

In (1), the decoding time is a variable value according to the complexity of data [12]. So, we assume that the decoding time is zero in this paper. And, because the PTS of the data is a known value, the only unknown value in (1) is T_t.

The minimum transmission time of the data can be calculated with (2) using the size of the data and available bandwidth for data transmission.

$$T_t = \frac{S_{Data}}{BW_{Available}} \times 90000, \quad (2)$$

where S_{Data} is the size of the data, $BW_{Available}$ is the available bandwidth, and 90,000 corresponds to the frequency of the system time clock, 27 MHz divided by 300 (90 kHz is the resolution of the presentation and decoding time stamps).

In general, the data element can be transmitted in a residual bandwidth that is not used for the transmission of the MPEG-2 AV TS [13]. So, we can know the available bandwidth if the whole bandwidth and MPEG-2 AV bandwidth are known.

From (2), (1) can be written as follows.

$$T_{st} = PTS_d - \frac{S_{Data}}{BW_{Available}} \times 90000. \quad (3)$$

Consequently, we can calculate the transmission start-time of the synchronized non-streaming data according to (3).

B. Transmission Time of Synchronized Streaming Data

Streaming data has several DAUs. The nominal, or reference, frequency of the DAU used in ATSC A/90 [3] is 60 Hz × 1000/1001. And, the size of the DAU and Data Elementary Buffer (DEB) is defined according to the service level, as shown in Table 1.

Table 1. Size of DEB and nominal DAU.

Classification	Minimum size of DEB (byte)	Maximum size of nominal DAU (byte)
Service level 1	120,120	40,040
Service level 2	480,480	160,160
Service level 3	1,921,920	640,640
Service level 4	7,687,680	2,562,560

The DEB is a buffer used to re-assemble the data into the DAU. The size of the DEB corresponds to three times the maximum size of the nominal DAU. All DAUs transmitted into the receiver enter the DEB, and are then decoded and presented when the 90 kHz portion of the STC becomes equal to the PTS of the DAU.

When the transmission of the DAU is too early, the DEB may overflow, as the other DAUs not yet decoded are still in the DEB. When the transmission of the DAU is too late, when the 90 kHz portion of the STC is bigger than the PTS at the moment the last byte of the DAU is entering the DEB, the DAU may not be decoded. This is why the transmission start-time of the DAU is very important. In this section, we propose a calculating method for the transmission start-time of a nominal DAU to avoid these problems.

At first, in order to prevent a delayed delivery, the DAU has to enter the DEB when the value of the 90 kHz portion of the STC is smaller than the value of the PTS. This can be written as

$$\text{90 kHz portion of the } STC_{te} < PTS_d, \quad (4)$$

where STC_{te} is the 90 kHz portion of the STC value when reception of the synchronized DAU is complete. According to the MPEG-2 systems standards [7], the STC is set as the MPEG-2 PCR. Thus, (4) can be written as

$$PCR_{te} < PTS_d, \tag{5}$$

where PCR_{te} is the 90 kHz portion of the PCR value when the transmission of the synchronized DAU is complete.

On the other hand, the maximum staying time of a nominal DAU that has a nominal frequency in the DEB is 50 ms, because the DEB can contain three DAUs, so 50 ms is needed to decode these three nominal DAUs. If a DAU remains in the DEB for more than 50 ms, a buffer overflow may occur. In other words, if a DAU enters and is removed within 50 ms, a buffer overflow will not occur. Because the DAU is removed from the DEB when the 90 kHz portion of the STC is equal to the PTS, (6) can be derived as

$$PCR_{ts} > PTS_d - \frac{50}{1000} \times 90000, \tag{6}$$

where PCR_{ts} is the 90 kHz portion of the PCR value when the data transmission starts.

Because the data transmission start-time (PCR_{ts}) is always earlier than the data transmission end time (PCR_{te}), (5) and (6) can be re-written in (7).

$$PTS_d - \frac{50}{1000} \times 90000 < PCR_{ts} < PCR_{te} < PTS_d. \tag{7}$$

Consequently, (7) is the sufficient condition needed for transmitting the DAU without a buffer overflow or delayed delivery.

2. Functions and Structure of Synchronized Data Injector

In the previous section, we proposed a method for calculating the transmission start-time of synchronized data. In this section, we explain the functions and structure of the synchronized data injector, which can transmit synchronized data while satisfying (3) and (7). In order to transmit synchronized data while satisfying (3) and (7), the synchronized data injector should have the following functions:

- A function to acquire an MPEG-2 PCR from the MPEG-2 TS: to check the current PCR.
- A function to acquire the size and PTS of synchronized data: to calculate T_{st} of (3)
- A function to multiplex the data with MPEG-2 AV TS: to transmit the data element in an accurate amount of time.

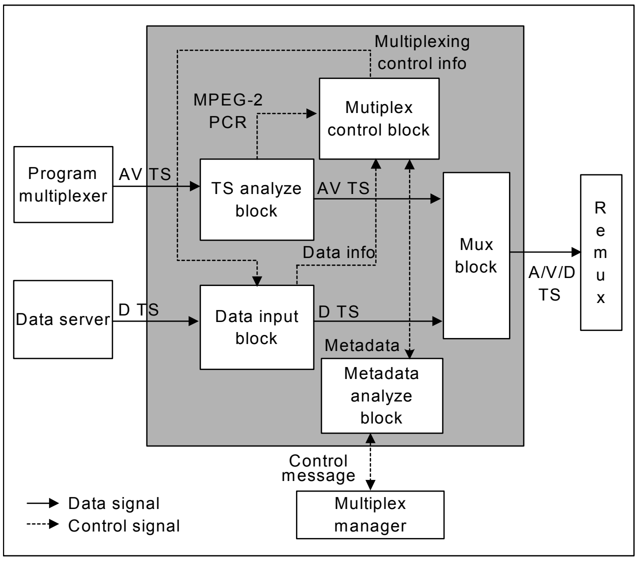


Fig. 7. Structure of synchronized data injector.

To perform the above functions, the synchronized data injector consists of five functional modules, as shown in Fig. 7.

The roles of the functional modules depicted in Fig. 7 are as follows:

- TS analyze block is used to acquire MEPG-2 PCR by analyzing the MPEG-2 TS that is entered in real-time.
- Data input block is used to save synchronized data from the Data Server into local storage and analyze it.
- Metadata analyze block is used to analyze metadata that contains information about the synchronized data.
- Multiplex control block is used to calculate the multiplexing time and control the multiplexing of data using the PCR and PTS of the data.
- Multiplexer (MUX) block is used to multiplex and transmit the MPEG-2 AV TS and synchronized data.

3. Operation Procedure of Synchronized Data Injector

The Synchronized data injector receives and transmits the MPEG-2 AV TS in real-time. If synchronized data to be transmitted exists, the synchronized data injector multiplexes the synchronized data with the MPEG-2 AV TS by using the following method.

A. Operation Procedure to Multiplex Synchronize Non-streaming Data

Figure 8 is the flow chart of the synchronized data injector for the transmission of synchronized non-streaming data. As illustrated in Fig. 8, the transmission of synchronized non-streaming data is performed as follows:

Step 1. Obtain the information about the size and PTS of the synchronized data.

Step 2. Calculate the transmission start-time according to (3).

Step 3. Obtain an MPEG-2 PCR by analyzing the MPEG-2 AV TS.

Step 4. Compare the MPEG-2 PCR with the transmission start-time. If the transmission start-time is equal to the MPEG-2 PCR, continue on to step 5. Otherwise, return to step 3.

Step 5. Multiplex the synchronized data with the MPEG-2 AV TS.

Step 6. Check whether all of the data are multiplexed. If all of the data are transmitted, the procedure is terminated, otherwise return to step 2.

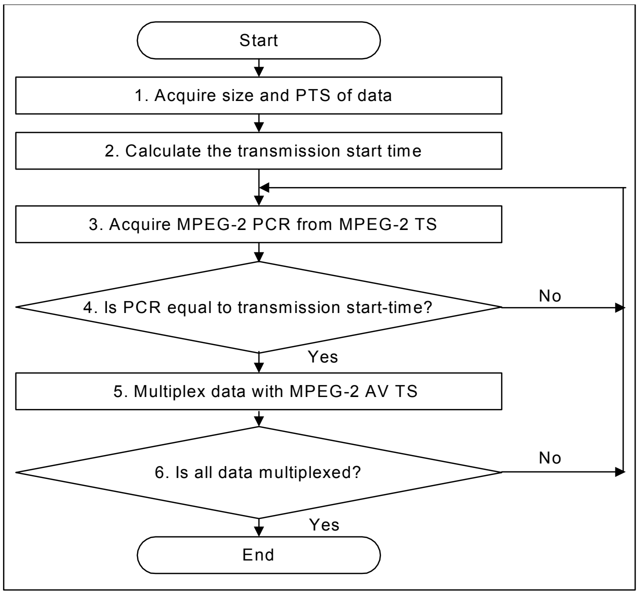


Fig. 8. Flow chart for transmission of synchronized non-streaming data.

B. Operation Procedure to Multiplex Synchronized Streaming Data

Figure 9 is the flow chart of the synchronized data injector for synchronized streaming data. As illustrated in Fig. 9, the transmission of synchronized streaming data is performed as follows:

Step 1. Obtain an MPEG-2 PCR by analyzing the MPEG-2 AV TS.

Step 2. Obtain the information about the size and PTS of the DAU.

Step 3. Calculate the transmission time according to (7).

Step 4. Compare the MPEG-2 PCR with the transmission start-time. If the transmission start-time is equal to the MPEG-2 PCR, continue on to step 5, otherwise return to step 1.

Step 5. Multiplex the DAU with the MPEG-2 AV TS.

Step 6. Check whether all of the DAUs are multiplexed. If all of the DAUs are multiplexed, terminate the process. Otherwise, return to step 1.

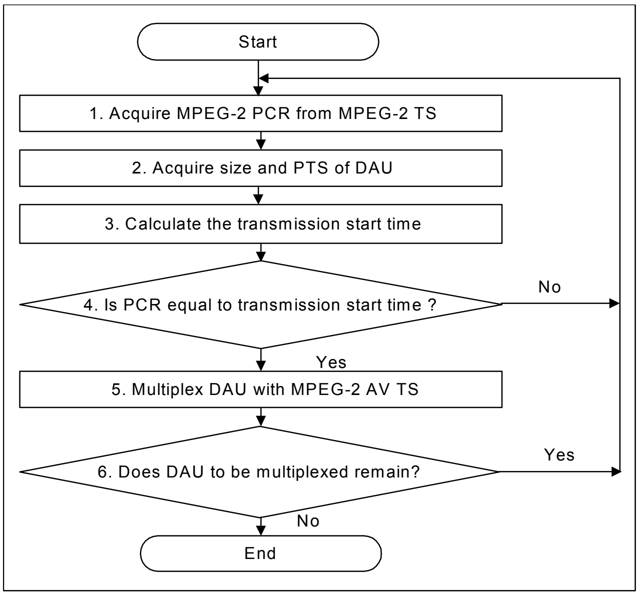


Fig. 9. Flow chart for the transmission of synchronized streaming data.

IV. Return Channel Server

1. Structure of the Return Channel Server

In this paper, we design and implement a return channel server to allow users to participate in a program by using a bi-directional interaction channel.

To provide user participation in a program and reflect the user's opinions or responses in real-time, the return channel server should include the following functions for receiving the user's data via an interaction channel: the storing and managing of a user's data via database management system (DBMS), producing contents which reflect the user's data stored in the DBMS, and providing content which reflects the user's opinion or response to the receivers by using a broadcast channel and/or IP network.

To support the above functions, the return channel server can be viewed as consisting of four major functional modules: a control module, network interface module, DBMS module, and real-time content authoring module. Figure 10 illustrates the structure of the return channel server.

2. Hierarchical Structure of a Return Channel Server

To construct a return channel server that has a service independent architecture, we designed a return channel server to include the hierarchical structure shown in Fig. 11. In our method, the return channel server consists of two parts: an RCSAE (return channel server application environment) and RCSA (return channel server application).

An RCSA is composed of an execution code and a parameter for executing the execution code. The execution code of an

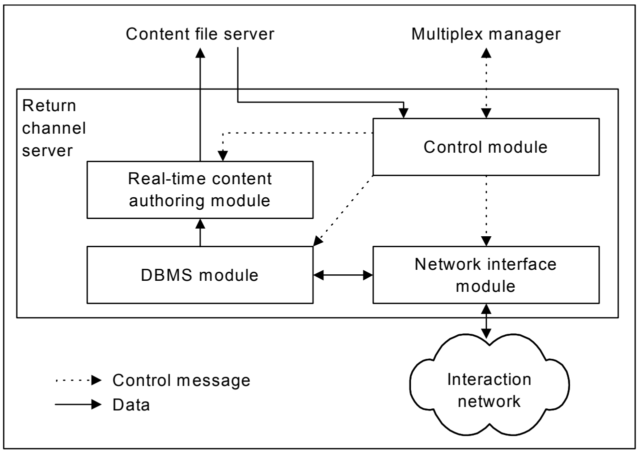


Fig. 10. Structure of a return channel server.

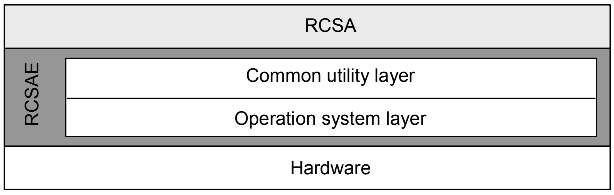


Fig. 11. Hierarchical structure of a return channel server.

RCSA defines the procedures for providing a specific broadcasting program using the return channel server and cooperates with the contents running on the receiver. An RCSA can be easily reconfigurable according to the broadcasting producer's intention. The broadcasting producer should produce both the RCSA and the contents for the receiver when he or she produces a new interactive broadcasting program.

As described earlier, an RCSA consists of a control module, network interface module, DBMS module, and a real-time content authoring module.

The RCSA for a control module is a parameter which indicates when the return channel server starts and terminates as it collects a user's data, how often the return channel server creates new content reflecting a user's data, and so on. The control module of an RCSAE controls the return channel server by using the information described in the RCSA to be used in the control module for a specific service.

The RCSA for the network interface module is a Web server program implemented by a common gateway interface (CGI) and cooperates with content running on the receiver. When bi-directional interactive broadcasting starts, the CGI permits the receivers to access the return channel server. At the same time, content running on the receiver informs users which interactive broadcastings are available and waits for a response from the users. If a user responds with a broadcasting program, the receiver transmits the user's data to the return channel server using the HTTP protocol. When the user's data arrives at the return channel server, the CGI provides proper services to the receiver according to the predefined service scenario. Furthermore, the CGI sends the user's data to the DMBS module.

The RCSA for the DBMS module receives a user's data from the network interface module and stores it. Also, the RCSA for the DBMS module serves inquiries from other modules, which may include user's information, user's data, log-in information, and so on.

The RCSA for the real-time content authoring module creates content reflecting the user's data and provides it to the receivers via a broadcasting channel in real-time. To do this, the real-time content authoring module asks the DBMS module to obtain the user's data, and the control module informs the real-time content authoring module how often it creates new content and where the created contents are stored.

To broadcast content reflecting a user's data, on the other hand, a multiplex manager must have information about the name of the content, size of the content, and so on. The control module sends information including the content's name, content's size, and so on, to the multiplex manager. Then, the multiplex manager can encode and broadcast new contents reflecting the user's data using the above information.

An RCSAE, on the other hand, provides the environment for the execution of RCSAs. It consists of an operating system layer and common utility layer. The common utility layer involves three functional modules: a control module, network interface module, and DBMS module.

When an interactive broadcasting program starts, the multiplex Manager sends a control message to the control module for an announcement of interactive service. The control message sent by the multiplex manager presents information about the RCSA such as its start-time, duration, location information, and the number of applications for the RCSA. Complying with the control message sent by the multiplex manager, the control module of the RCSAE takes and executes the RCSA.

In the meanwhile, the network interface module of the RCSAE provides the environment for an interaction channel by using a web server utility, and the DBMS module of the RCSAE provides functions for database management by using the DBMS.

Note that the RCSAE provides a service-independent platform for the RCSA, and the RCSA defines the service-dependent procedures for the specific broadcasting program.

V. Experiment

In order to verify the proposed broadcasting system, we constructed an interactive broadcasting system. As Fig. 12 depicts, the proposed interactive broadcasting system consists of a return channel server (label 6), synchronized data injector (label 7), contents file server (label 8), PSIP generator (label 9), multiplex manager (label 10), data server (label 11), and so on. A monitor (label 1), AV encoder (label 2), VCR (label 3), TS generator (label 4), set-top box (label 5), modulator (label 12), and re-multiplexer (label 13) were constructed using commercial products.

We conducted experiments in the interactive data broadcasting system in order to verify the synchronized data multiplexing function of the synchronized data injector, as well as the interaction communication of the return channel server, by demonstrating several services with local interactivity and remote interactivity.

As test services, we used a drama program designed for a local interactive service focusing on synchronized service, and a quiz show program designed for a remote interactive service emphasizing user participation in the broadcast program. The results of the test services were displayed on the TV monitor using a receiver complying with ATSC-DASE (DTV Application Software Environment) [11].

Fig. 12. Interactive data broadcasting system.

1. Local Interactive Service

In this section, we demonstrate a synchronized data service with local interaction to verify the functions of a synchronized data injector.

In order to test the functions for the synchronized data service, a drama program is used as the AV media. A data element such as a character profile for synchronized non-stream data and an advertisement for the synchronized streaming data are also used.

In the synchronized non-stream data service, the character profile data is presented at a certain time corresponding to a character's appearance in the drama program displayed on the TV monitor.

In the synchronized streaming data service, on the other hand, the car advertisement is executed at a certain time corresponding to a car's appearance in the drama program displayed on the TV monitor.

Table 2 shows the content specifications of the AV and data elements used in the test. The drama program is composed of MPEG-2 video and AC-3 audio. The character profile consists of XDML (eXtensible DTV Markup Language), MNG (multiple networks graphics) and JPEG. The car advertisement is also made up of MPEG-2.

Table 3 shows the transmission parameters of the AV and data elements used in the test. The drama AV program is transmitted at the rate of 17 Mbps and the data element at the rate of 2 Mbps. The character profile is encoded into the DSM-CC section (synchronized download protocol), and the car advertisement is packetized into an MPEG-2 packetized elementary stream (PES) defined in [3].

Table 2. Specifications of contents used in the test.

Service	AV		Data	
	Description	Specification	Description	Specification
synchronized non-stream	Drama	MPEG-2 video, Dolby AC-3 audio	Character profile	XDML, MNG, JPEG
synchronized stream			Car advertisement	MPEG-2 ES

Table 3. Transmission parameter.

Service	AV		Data	
	Bandwidth	Protocol	Bandwidth	Protocol
synchronized non-stream	17 Mbps	MPEG-2 system	2 Mbps	DSM-CC Section (Character profile)
synchronized stream				PES (Car advertisement)

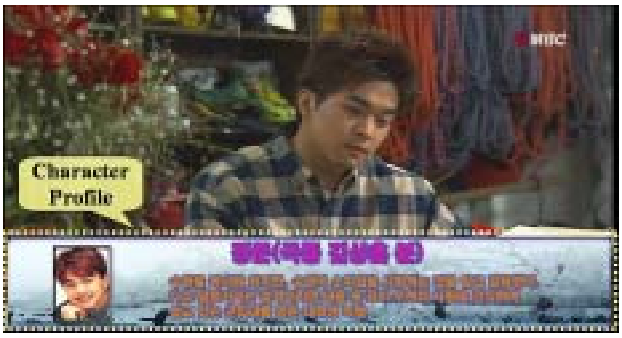
For the experiment, we encode an AV signal into an MPEG-2 AV TS in advance. Then, from the MPEG-2 AV TS, we obtain the PCR value of the time when the data have to be presented and can set the acquired PCR value to the PTS of the DAU.

Figures 13 and 14 show that the synchronized non-stream data and synchronized stream data were successfully presented by the receiver.

As shown in Figure 13(b), the character profile in the dotted rectangle is displayed on the screen of the TV monitor when the STC of the receiver is equal to the PTS of non-stream data delivered by the synchronized data injector.

As shown in Figure 14(b), the stream data synchronized with the main program; the car advertisement in the dotted rectangle is displayed in the screen at the moment when the car door closes.

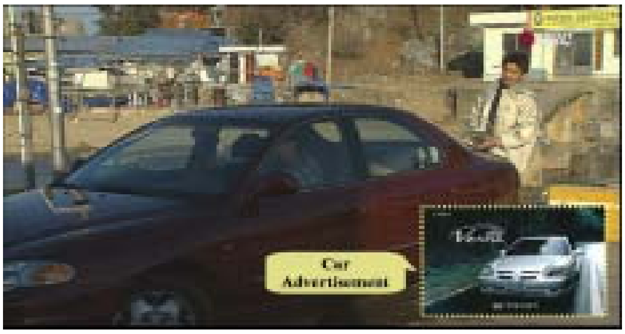
(a) A screen shot before the data is synchronized with the program.



(b) A screen shot soon after the data is synchronized with the program.

Fig. 13. The presentation of synchronized non-stream data.

(a) A screen shot before the data is synchronized with the program.



(b) A screen shot soon after the data is synchronized with the program.

Fig. 14. Presentation of synchronized stream data.

The car advertisement mentioned above is an encoded MPEG-2 elementary stream (ES), which consists of a single video elementary stream and is defined as content type 'video/mpv' of DASE-1 [11]. Then, it is put into the synchronized data-byte field of the synchronized data-packet structure defined by ATSC A/90 [3].

2. Remote Interactive Service

In this section, we demonstrate a remote interactive service, like a quiz show, to verify the interaction function of the return channel server. A quiz show enables users to participate in the broadcast program by allowing them to answer quiz questions through a bi-directional interaction channel. After the return channel server checks whether the users' responses are correct or not, it sends each user's data, such as their scores or answers, through the interaction channel and their ranking information through the broadcast channel. Table 4 shows the data path and flow that occurred in the quiz show.

The theoretical bandwidth limitation for newly created content is up to about 19.39 Mbps because of the bandwidth limitation of a broadcasting channel is limited to about 19.39 Mbps according to ATSC standard. In our test, however, we broadcast the content at a maximum of 2 Mbps, considering the quality of the AV. The bandwidth for an interaction channel depends on the various characteristics of the physical media.

Figures 15 through 18 illustrate a quiz show, one example of a remote interactive service using the return channel server.

Figure 15 shows a main view of a quiz show, which is transmitted through the broadcasting channel. A main menu inside the dotted lines depicts three sub-menus such as a help menu and event menu.

If a user wants to participate with a quiz show, the user inputs his or her ID number into the edit-box shown inside the dotted lines in Fig. 16. The user's ID is then transmitted to the return channel server because it is needed for certification of the user, a summation of the user's responses, and for determining the ranking information of the user.

If users give answers to the quiz show by clicking on the

Table 4. Data flow of a quiz show.

Description	Direction	Path
Question, menu	Return channel server → Receiver	Broadcast channel
User's response	Receiver → Return channel server	Interaction channel
Answer, score	Return channel server → Receiver	Interaction channel
Ranking	Return channel server → Receiver	Broadcast channel

Fig. 15. Main menu of a quiz show.

Fig. 16. Login window.

Fig. 17. View of an O-X quiz.

Fig. 18. View of a final result.

icon, the receiver sends each user's answer to the return channel server via the bi-directional interaction channel. Then, the return channel server makes a scoring for each user and sends a score and grade to each receiver. Figure 17 shows the screen shot of an O-X quiz. The part shown inside the dotted lines is a temporary score and grade. Note that this temporary score and grade are created in real-time, reflecting each user's responses.

When the quiz show is finished, the return channel server lists up the scores and rankings of all of the participants, as shown in Fig. 18. Then, the final results are transmitted through both the broadcasting channel and the return channel. The part shown inside the dotted lines shows a final result.

VI. Summary and Conclusion

We present an interactive data broadcasting system compliant with ATSC data standards. The proposed system can provide various data services by adopting a synchronized data injector and return channel server to supplement the data-broadcasting system model recommended by ATSC DIWG.

To provide various data services, we propose a method for calculating an accurate time for the transmission of synchronized data and multiplexing it in real-time, as well as a method for the implementation of a synchronized data injector.

Also, we propose a method for constructing a return channel server to provide user participation in programs using the bi-directional interaction network. The proposed method provides a generic system architecture of a return channel server, which has a service-independent platform by adopting hierarchical structures. Test results show that the proposed system is appropriate for synchronized data broadcasting and remote interactive services, complying with ATSC standards.

Further works will include a supplement of the proposed system to support intelligent multimedia services for use at anytime and anywhere using MPEG-4 and MPEG-7 technologies.

References

[1] ATSC T3/S16 Draft Standard 030, *Interaction Channel Protocols*, Sept. 2003.

[2] J.H. Choi, M.S. Park, Y.J. Lee, J.S. Choi, and J.W. Kim, "Design and Implementation of Data Broadcasting Emission System based on ATSC," *Proc. KSBE Conf. 2001*, 2001, pp. 15-19.

[3] ATSC Standard A/90, *ATSC Data Broadcast Standard*, 2000.

[4] http://www.nab.org/conventions/nab2003/exhibit.asp (current Nov. 17, 2003).

[5] News from ATSC Volume Three, Issue Seven, July 2002. http://www.atac.org/news_information/newsletter/ATSC_Newsletter-7.pdf (current Nov. 17, 2003).

[6] ATSC IS/151 Standard, *Implementation of Data Broadcasting in a DTV Station*, Nov. 1999.
[7] ISO/IEC 13818-1, *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 1: Systems*, International Standard (IS), 1995.
[8] ISO/IEC 13818-2, *Information Technology – Generic Coding of Moving Pictures and Associated Audio Information – Part 2: Video*, International Standard (IS), 1995.
[9] ATSC Standard A/53, *ATSC Digital Television Standard*, Sept. 1995.
[10] ATSC Standard A/52A, *Digital Audio Compression (AC-3), Rev. A*, Aug. 2001.
[11] ATSC Standard A/100, *DTV Application Software Environment Level 1 (DASE-1)*, Mar. 2003.
[12] ATSC Standard A/93, *Synchronized /Asynchronous Trigger*, 2002.
[13] Bill Nicholls, "Data Transmission Efficiency," *Proc. 2001 Broadcasting Engineering Conf.*, Apr. 2001.
[14] ATSC Standards A/65A, *Program and System Information Protocol*, 2000.

Jong Myeon Jeong received the BS, MS, and PhD degrees in computer science and engineering from Hanyang University, Korea in 1992, 1994, and 2001. After working as a Senior Member of Engineering Staff at Electronics and Telecommunication Research Institute (ETRI) located in Daejeon, Korea between March 2001 and February 2004, he became an Assistant Professor in the Division of Marine Electronics and Communication Engineering at Mokpo National Maritime University, starting in March 2004. His research interests are data broadcasting, computer vision, pattern recognition, multimedia application, and so on.

Yong Ju Lee received the BS and MS degrees in electronics from Kyungpook National University, Daegu, Korea, in 1999 and 2001. Since 2001, he has been a Member of Research Staff in ETRI, Daejeon, Korea, where he has been involved in developing an interactive data broadcasting system. His research interests include intelligent broadcasting and digital signal processing.

Min Sik Park received the BS degree in electrical engineering from Kwangwoon University of Seoul, Korea in 1997 and the MS degree in mechatronics from Kwangju Institute of Science and Technology, Gwangju, Korea in 1999. Since 1999, he has been a Member of Research Staff in ETRI, where he has been involved in developing an Interactive Digital Television (IDTV). His research interests include signal and image processing, robot vision, visual communications and data broadcasting systems.

Ji Hoon Choi received the BS and the MS degrees in electronic engineering from Kyunghee University, Suwon, Korea, in 1999 and in 2001. In 2001, he joined the Data Broadcasting Research Group in ETRI, Daejeon, Korea, where he has been working on the interactive data broadcasting system. His research interests are data broadcasting and network QoS.

Jin Soo Choi received the BE, ME, and PhD degrees in electronic engineering from Kyungpook National University, Korea, in 1990, 1992, and 1996, respectively. Since 1996, he has been a Senior Member of Engineering Staff in ETRI, Korea. He is currently the leader of the Data Broadcasting Research Team. He has been involved in developing an MPEG-4 codec system and data broadcasting system. His research interests are visual signal processing and interactive services in the field of digital broadcasting technology.

Jinwoong Kim received BS and MS degrees in electronics engineering from Seoul National University, Korea, in 1981 and 1983, and the PhD degree in electrical engineering from Texas A&M University, USA in 1993. Since 1983, he has been a Member of Research Staff in ETRI, Korea, working for the development of the TDX digital switching system, MPEG-2 video encoder, HDTV encoder system, MPEG-7 technology and data broadcasting system. He is currently Director of the Broadcast Media Research Group. His research interests include digital signal processing in the field of video communications, multimedia systems and interactive broadcast systems.