DOI: http://dx.doi.org/10.26483/ijarcs.v9i1.5277

Volume 9, No. 1, January-February 2018

ISSN No. 0976-5697

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

ENCAPSULATE AND TRANSFER MESSAGE ON DMB

Peng Xu
College of Electronic Engineering
Chongqing University of Posts and Telecommunications (CQUPT)
Chongqing, P.R. China

Abstract: In MySQL, there is a table simulating a message queue with some messages, which will be broadcast by digital multimedia broadcast(DMB). To facilitate the receiver decoding, every message need to be encapsulated. Besides, before being transferred, each encapsulated message will be split into multiple packages. After connection is established, packages will be sent to DMB server one by one.

Keywords: digital multimedia broadcast; encapsulated; split; transferred

I. INTRODUCTION

DAB (Digital Audio Broadcasting) is a new digital broadcasting standard [1][2] developed by the European Telecommunications Union and has been adopted by a large number of countries. In 2006, China set forth the digital audio broadcasting standard [3]. DAB has many advantages in excellent sound quality, multimedia, anti-attenuation of wave propagation, suitability for high-speed mobile reception [4] and so on. DAB is not just an audio broadcasting system, excellent data transmission and mobile reception make it evolve into digital multimedia broadcasting (DMB).

Via DMB, many kinds of data including audio, video and picture can be transferred and shown in different way. What is more, DMB server provides some application programming interfaces(APIs) to extend its data source, message in database is one of which.

This paper describes how a DMB message is encapsulated and transferred. The rest of the paper is organized as follows. Section II describes message structure and encapsulating format. Section III describes how to split a message into segments and package a segment. Section IV reports the tests.

II. ENCAPSULATE

A. *Message structure*

In MySQL, there is a table simulating a message queue, in which, there are some records including message targets, message type, content type and content.

The field, message targets, means who will receive the message. Target ID, as a field, is the only valid identify to distinguish different targets. Each message target has two IDs. The two IDs are respectively target ID and target group ID. Users can not only precisely control each target, but also directly control one group of targets. Target ID and target group ID are unique. Targets with the same target group ID belong to the same group.

There are 3 message types including single message, group message and all message. Single message is sent to some individual message targets, while group message is sent to some individual message target groups, and all message is sent to all message targets. Single message according to the mode is divided into discrete single message and continuous single message. Discrete single message refers to message targets with discrete IDs, and continuous single message refers to message targets with continuous IDs.

Content type must be one of audio, picture and video. It is to guide message targets to correctly parse content in appropriate way.

B. ***Encapsulate message***

After fetching a message from message queue, we need to encapsulate it in specified format as Fig. 1.

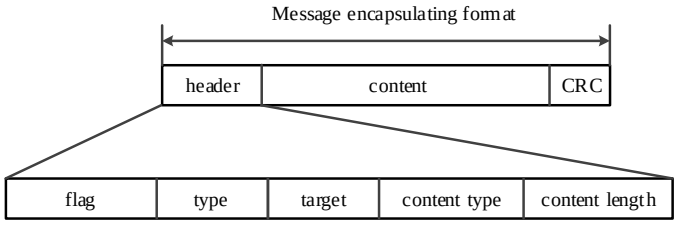

Fig. 1. Message encapsulating format

To enable the message target to correctly parse out the message broadcast by DMB, a header needs to be added to the message to define the format of the message. Meanwhile, to ensure the correctness of the message, messages need to be verified. Therefore, a tailor need to be added.

Flag is set at the beginning of the message to indicate the message format, we define flag as string “MSG”, which uses up 3 bytes.

Message type is determined according to Table 1.

Table 1. Message type

value	0	1	2	3
type	continuous single message	discrete single message	group message	all message

Target is influenced by type. When type is continuous single message, target has a start target ID and an end target ID, which respectively uses 2 bytes up. When type is discrete single message or group message, target has number of IDs and each independent ID. When type is all message, target does not exist.

Content type is determined according to Table 2.

Table 2. Content type

value	0	1	2	3
type	text	picture	audio	video

The next is content length and content data. Content length uses 4 bytes up, which is enough to contain almost all multimedia content.

CRC is at the ending of the message to ensure accuracy of the entire message.

III. TRANSFER

A. Build connection with DMB server

Communication between client and DMB server is via windows socket. Client can specify server IP address, port number and channel ID to shake hands with server. After that, client can get some information about server including channel type and bitrate.

B. Message split and package

Usually an encapsulated message is too large to be sent at once. We must split it into segments and transfer them one by one. But exceptions may occur including segment loss, repetition and disorder. Certain measures must be taken to eliminate such anomaly and ensure message targets receive complete and correct message. Packaging each segment, that is, adding header and trailer to each segment can solve the above problem. Each segment is packaged in such format as shown in Fig. 2.

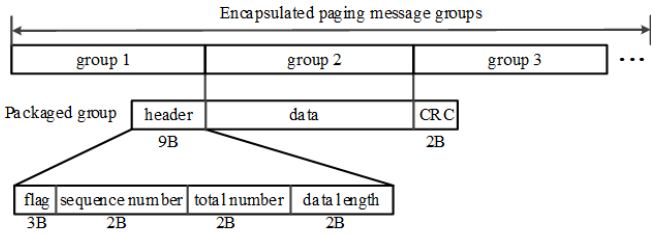

Fig. 2. segment packaging format

The first 3 bytes in the header are fixed string "MSG" as a message flag indicating that the current package is a DMB message package. The package sequence number facilitates message targets to adjust out-of-order packages and the total number of packages to facilitate message targets to determine whether the message is received completely. Package sequence number, total number and data length are calculated in this way.

In DAB working mode 1, one transmission frame contains 29184 bytes of data, and 2-bytes length is enough. Each DAB data service channel bitrate is an integral multiple of 8 Kbps, we suppose it is the minimum 8Kbps. Because one DAB frame takes up time of 24ms, then in the message channel each data frame length is 8Kbps * 24ms / 8bit, that is 24 bytes. Exclude the header and the tailor, which use 11 bytes, pure data is 13 bytes. Then 2-bytes length (65535) of packages can carry a total of 851,955 bytes, it is almost enough to carry an encapsulated message. In general, the capacity for message service channel can be flexibly set from 8Kbps to 1.5Mbps within the allowable range of the DAB standard, depending on user needs. Therefore, the channel can carry more data in use.

IV. SYSTEM TEST

The purpose of the test is to verify whether the message is encapsulated and transferred correctly. If specified message targets can receive, decode and decapsulate message, it means the design is correct.

Test steps are as follows.

a) Start DAB transmitting system, and specify one channel of 128 Kbps bitrate as message service channel from DAB transmitting managing server.
b) Start encapsulating and transferring software.
c) Prepare 10 DMB message targets, number them, divide them into two groups, 5 terminals every group, name them as group1 and group2; ID of 10 message targets are unique.
d) Store many kinds of messages in message queue. They must contain different message types. They must have different content types and content length.

Test results show that both the specified message target and message target groups can receive messages correctly, which means the design of encapsulating and transferring software is right.

REFERENCES

[1] European Broadcasting Union. ETSI EN 300 401, Radio broadcasting system; Digital audio broadcasting (DAB) to mobile, portable and
fixed receivers. 2006.

[2] European Broadcasting Union, ETSI TS 102 428, Digital audio broadcasting (DAB); DMB video service; User application specification. 2006.

[3] Guoyu Wang, Hongsheng Zhang, Mingying Lu. “Developmental trend of digital broadcasting in China (in Chinese)”. Chin Sci Bull (Chin Ver), 2014, 59: 2320-2327.

[4] Chao Pan, “Research on video coding optimization”. Master, Chongqing University of Posts and Telecommunications, Chongqing, China, 2016.