Recovering Data from Pdu Stream with HCS Error in WIMAX

Hongyuan Chen
School of Electronic and Optical Engineering
Nanjing University of Science and Technology (NJUST)
Nanjing, China, 210094
chy@mail.njust.edu.cn

***Abstract*—In IEEE 802.16e (WiMAX), each data burst consists of multiple MPDUs. The MPDUs of one data burst are distinguished from each other according to the generic MAC header (GMH). The GMH contains the length information of MPDU. If the GMH is damaged, it doesn't know what's end of the MPDU. As a result, the damaged MPDU and successive MPDUs in the data burst shall be discarded even if the successive MPDUs are correct received. In this paper, we point out the cursory discarding and propose a recovering scheme to improve the network bandwidth.**

Keywords- IEEE 802.16e, WiMAX, HCS, CRC

I. INTRODUCTION

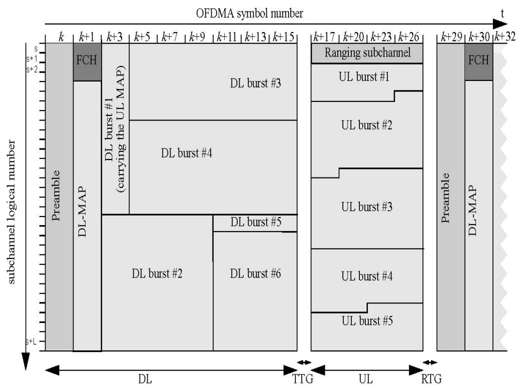

Figure 1. an OFDMA frame (with only mandatory zone) in TDD mode [1]

The IEEE 802.16e (WiMAX) adopts the OFDMA as the physical modulation scheme. TDD is the main operating mode of 802.16e. One example of a TDD frame structure is illustrated in Fig.1. A TDD frame consists of DownLink (DL) and UpLink (UL). The fields of Preamble, FCH, DL-MAP and UL-MAP in DL are used in the purpose of management while the ranging subchannel in UL is also for management. The data transmission between base station (BS) and mobile station (MS) are allocated in the data burst (DL Burst in DL or UL Burst in UL). The basic data transmission unit is the MAC packet data unit (MPDU). One data burst consists of multiple MPDUs.

The data transported over air is prone to be damaged due to a variety of noise sources. In the MAC layer, it is common method to append a Cyclic-Redundancy-Check (CRC) code into MPDU. The CRC code aims to check if the received MPDU bits are same as the legacy bits sent by the original transmitter. Thus, the CRC is used to judge if the MPDU is successfully received. If the MPDU is failed to transmission, the failed MPDU shall be sent over again. As a result, the CRC in MAC layer can be the means to overcome the channel bit error.

Two CRC codes are defined in the IEEE 802.16e [1]. One is a 32-bit code (CRC32) which is inserted into the end of one MPDU in order to protect the entire MPDU including the MAC header. Another is called as Header Check Sum (HCS), an 8-bit code, which is a part filed of the MAC header and used to protect the MAC header. The MAC header is called as Generic MAC Header (GMH). The GMH contains the management information of the MPDU, e.g. the MPDU length and the MPDU type (Management MPDU or data MPDU), etc. If the HCS detects that the GMH is damaged, the MPDU shall be deemed as a damaged MPDU because the length field of GMH is uncorrected. Since the length information of the damaged MPDU can be only obtained from the GMH of the MPDU, as a result, it doesn't know what's the end of the damaged MPDU. If some MPDUs follows the damaged MPDU in a data burst, they shall be discarded even if they are correctly received because the damaged MPDU has erased the following GMH information.

In fact, the MPDUs following the damaged MPDU may be successful transmitted. Based on the standard, those MPDUs shall be misled and discarded by the receiver because the receiver can't correctly decode the data bits which begin from the first bit of the damaged MPDUs. Thus, the bandwidth shall be waste if the receiver completely discards those MPDUs and the transmitter retransmits them. If the receiver is capable to recover those MPDUs, the bandwidth can be saved. Our motivation is based on the simple ideal: how to recover those MPDUs so that the bandwidth shall be correctly saved. In this paper, we propose one method to find which MPDUs (following the damaged GMH) may be correct and how to completely recover them in order to save the bandwidth. By using our method, the throughput/bandwidth shall be enhanced.

II. PROBLEM STATEMENT

It is necessary to explain what's problem in detail. We have to state that the following description is based on the receiver but not be involved in the transmitter. We describe what happens in receiver after it receives a data bit stream of one data burst.

The physical layer receives the signals from air and encodes them into a data burst. The data burst consists of some successive bits (0 or 1). The data bit stream of data burst is nonsense of physical layer. The physical layer needs to send the data bit stream into MAC layer. From the viewpoint of MAC layer, the data bit stream consists of multiple MPDUs. The MAC layer has the responsibility to decode each MPDU from the data bit stream of data burst. When the MAC layer decodes the data bit stream, it needs the GMH information to separate the MPDU from each other.

The GMH structure is shown in the Fig.2. Each MPDU begins with a fixed-length GMH. The header is followed by the Payload of the MPDU and a CRC32 code ends the MPDU. The GMH is a 6-bytes tuple which contains the management information of MPDUs. Two fields of GMH have to be stated. The LEN field (11-bit) saves the length of the MPDU while the HCS filed is to protect the GMH. These two fields are vital to separate each MPDU from the data bit stream.

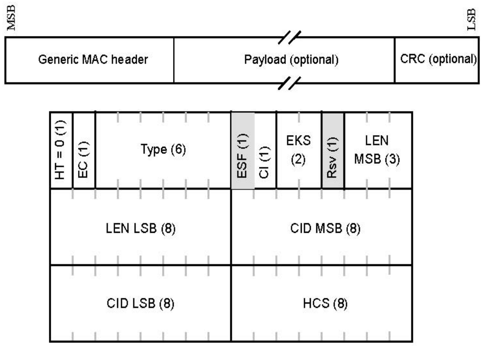

Figure 2. The Generic MAC Header Format [1]

When the MAC layer decodes the MPDU from the data bit stream, it needs two steps. The first step is to check if the GMH information is correctly received and can be utilized to separate the bit stream. The correctness of GMH is indicated by the HCS. If the GMH can be trusted, the second step is to utilize the LEN field of GMH to determine the MPDU length. Then, it can pick a MPDU from the data bit stream based on the length of the MPDU.

As stated above, the HCS is used to check whether the GMH is correct or not. If GMH is correctly received, it is no problem for the MAC layer to recognize the MPDU. But, if the HCS indicates that the GMH is damaged, the corresponding MPDU shall be damaged because all related information of the MPDU has been lost. Since the length of the MPDU has be lost, there is no way to know where is the termination of the MPDU. Thus, beginning from the damaged MPDU, all following MPDU can't be picked out from the data bit stream. For those data bits which are from the first bit of the damaged MPDU to the last bit of the data bit stream, all of them shall be discarded since it can't recognize the end of the damaged MPDU.

An example is illustrated in Fig.3. Multiple MPDUs are transmitted in one data burst. MPDU-n is the n-th MPDU transmitted in the current data burst. If the GMH of MPDU-n is detected error by the HCS, MPDU-n, MPDU-(n+1) and successive MPDUs of the data burst shall be discarded because it doesn't determine the MPDU delimiters. It is OK for the MAC layer to discard MPDU-n since its information has been damaged. But, it is cursory to discard MPDU-(n+1) and successive MPDUs.

Problem: In fact, the MPDU-(n+1) is correctly received. But, it has to discard the MPDU-(n+1) based on the above explanation even though all bits of MPDU-(n+1) has been correctly received. As a result, the throughput/bandwidth shall be wasted due to the cursory discarding action. In the worst case, if the GMH of the first MPDU of the data burst has only one bit error, the entire data burst has to be discarded even though the other bits of the data burst are correct. The bandwidth consumed by the data burst is waste. If the data burst is very long to contain a lot of bits, the case becomes more badly. Obviously, it has to overcome the cursory actions.

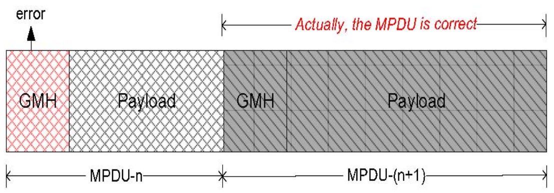

Figure 3. One example for problem explanation

III. Recovering Scheme

We propose a simple but effective scheme to recover those bits which should not be cursorily discarded. The basic ideal is to one-byte by one-byte check whether a six successive bytes tuple is a GMH or not.

Generally, a MPDU can be completely distinguished from the data burst via the GMH information. If it can completely dig out the GMH tuple from the data burst, those correctly received MPDUs shall not be discarded. Since the GMH is a fixed length tuple with 6-bytes, it can search a 6-bytes tuple to check if the 6-bytes tuple is GMH. Through computing the HCS code of a 6-bytes tuple, we can determine if the 6-bytes tuple is GMH. If the HCS code is equal to the last byte of the 6-bytes tuple, then the 6-bytes tuple is a GMH. Once the GMH is determined, MPDUs can be easily separated from the data burst. If the 6-bytes tuple is not GMH, a new 6-bytes tuple shall be used to check it. The new 6-bytes tuple starts with the second bytes of the old 6-bytes tuple. Through the one-byte by one-byte searching, it can dig out all correctly received GMH tuples.

After a data burst is received, the MAC layer shall begin decode the MPDUs from the first bit. When decoding, once there is an error reported by HCS, the recovering scheme is to work for checking where the next MPDU begins. The start point is the first byte which follows the damaged GMH. A 6-bytes tuple, whose first byte is the start point, is the first tuple to be checked. For example, in Fig.3, the start point is the 7-th byte of MPDU-n. The first 6-bytes tuple begins from the 7-th byte of MPDU-n and ends with the 12-th byte of MPDU-n. If the first 6-bytes tuple is not GMH, then the second 6-bytes tuple is comprised of the 8-th byte of MPDU-n and successive five bytes.

The basic flow chart of the recovering scheme is illustrated in Fig.4. The recovering scheme adopts the following steps:

One window is set. The start byte of the window is the first byte which follows the damaged GMH. The window size is constant to 6-bytes.

Suppose the GMH of next MPDU consists of the 6 bytes which are included in the current window. It uses the general HCS function to check if the 6-byte is a valid GMH, i.e., after computing the HCS code over the first 5 bytes, it should check if the HCS code is same as the 6th byte of the window.

If the HCS check sum is OK, the 6-byte should be deemed as a new GMH. Then, it can utilize the LEN field of the new GMH to separate the MPDU from the remaining bits of the data burst. Now, Quit the recovering scheme since the MPDU can correctly decoded.

If the HCS check sum is failure, it means the 6-byte tuple doesn't contain the GMH information. Move the start byte of the window to the next byte which follows the current start byte of the window. Go to step 2) to process the new 6-bytes tuple contained in the window.

Following the start byte of window, if there is no any more 6-bytes tuple to compose one new window, Quit the recovering scheme.

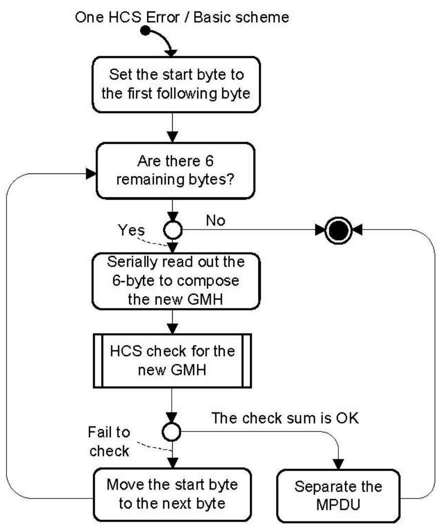
One HCS Error / Basic scheme
Set the start byte to the first following byte
Are there 6 remaining bytes?
Yes
No
Serially read out the 6-byte to compose the new GMH
HCS check for the new GMH
The check sum is OK
Fail to check
Move the start byte to the next byte
Separate the MPDU

Figure 4. Flow Chart of Recovering Scheme

As long as a damaged GMH occurs when decoding one data burst, we can utilize the recovering scheme to find the next correctly received MPDUs. It can use the above steps to check all GMH tuples until the last bytes of the data burst. Through the scheme, all correctly received MPDUs can be successfully separated from the data burst.

IV. Performance Evaluation

We implement the recovering scheme in the NS2 simulation tool. All parameters are compliant with the IEEE 802.16e. In the network, there is one MS station. In order to show the performance, only the DL subframe is used for data transmission. DL consists of multiple DL data bursts and each DL data burst is comprised of multiple MPDUs. Fig.5 illustrates the performance. The y-axis is the ratio of recovering scheme to the original scheme. The number of MPDUs in a data burst has the significant influence on the performance. Obviously, as the increasing of the number, the throughput is improved more greatly. At the same time, for very good channel condition, there is a slight improvement because the GMH is correct almost of time. But, for high channel bit error, the performance is significant improved. When the number is 40 and the channel bit error is 10-4, the throughput can be enhanced up to 10%. Through our scheme, the performance can be greatly improved under the high channel bit error and the big number of MPDUs in each data burst. At least, our scheme can be superior to the original scheme at all network scenarios.

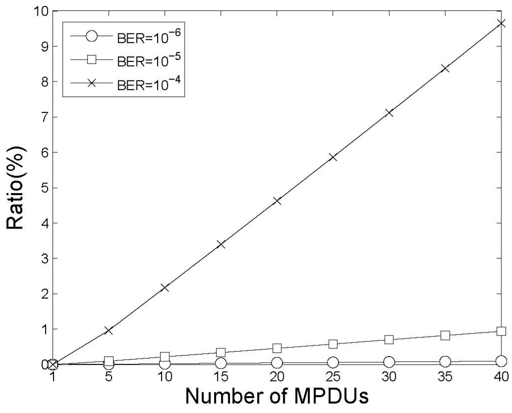

Figure 5. Performance enhancement: the improvement ratio by recovering scheme

V. Summaries

In the TDD mode of IEEE 802.16e, the data burst is the basic unit of data transmission in a TDD frame. Each data burst consists of multiple MPDUs. The GMH contains the length information of one MPDU. If the GMH is damaged due to the channel error, the MPDU will be discarded. As a result, all MPDUs which follow the damaged MPDU with the same data burst shall be discarded. However, those MPDUs may be correctly received. In this paper, we propose one scheme to recover those MPDUs. The scheme checks a 6-byte tuple one-byte by one-bye if it is a GMH. Through NS-2 simulation tool, we validate the efficiency of our scheme. For high channel bit error and big number of MPDUs in a data burst, the throughput can be greatly improved. Due to one-byte by one-byte checking, the complexity of the recovering scheme may be high. In the future, how to reduce the complexity of the recovering scheme and implement it in real product are our targets.

Acknowledge

This work was supported by ZiJing Scholar Plan of NJUST (No.AB41375).

References

[1] IEEE, Part 16: Air interface for fixed and mobile broadband wireless access systems Amendment 2: Physical and medium access control layers for combined fixed and mobile operation in licensed bands. IEEE 802.16e, Sept, 2005.

[2] Bianchi, G., Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Sel. Areas Commun., 18(3), Mar., pp. 535–547, 2000

[3] Ni, Q., Li, T., Turletti, T., and Xiao, Y., Saturation throughput analysis of error-prone 802.11 wireless networks. Wiley J. Wireless Commun. and Mobile Comput., 5(8), Dec., pp. 945–956, 2005