Project	**IEEE 802.16 Broadband Wireless Access Working Group <http://ieee802.org/16>**
Title	**Block Turbo Code section edits in response to comment #524**
Date Submitted	**2000-11-09**

Source(s)	Dave Williams AHA 2365 NE Hopkins Ct. Pullman, WA 99163	Voice: 509 334 1000 Fax: 509 334 9000 mailto:davew@aha.com

Re:	IEEE 802.16.1-00/01/r4

Abstract	Clarifies shortening for both downstream Mode B and upstream. Replaces the 392-byte mode B down stream code with a 231-byte code due to TC layer limitations identified. Adds a method to shorten the last code word in a message.

Purpose	In response to comment #524 calling for a clarification of the shortening method for BTCs. This change also addresses last code word shortening. The change also fixes a TC layer related problem with maximum code word sizes.

In the downstream Mode B section replace text in the paragraph entitled Shortened BTCs with the following text. (~page 287 ln 15)

Shortened BTCs

To match packet sizes, removing symbols from the array shortens a product code. In general, rows or columns are removed until the appropriate size is reached. Codes selected shall have an integral number of information bytes. Shortening for the downstream Mode B, 128-byte code in this specification. The 253-byte code follows the same method starting with 64,57 constituent codes and shortening by thirteen rows and eleven columns. The exact shortening method used for codes other than the two codes in this specification is described in a following section entitled, "Method for determining codes for payload sizes different than the listed examples".

In the downstream Mode B, page 288 starting with line 7 make the following changes:

The Table figure is labeled twice as Table 73 and again as Table 3-1. Delete the Table 3-1 reference. It is a typo.

In the table, replace the 63,56 x 63,56 code with a 53,46 x 51,44 code. This is a 253-byte code. Also change the code rate to 0.749. The reason is that the original code exceeds the TC layer maximum block size, which is 255 bytes. The new code provides approximately the same code rate with a code that provides an integral number of bytes.

In the upstream section, replace text in the paragraph entitled Shortened BTCs with the following text (~page 303 ln 51)

Shortened BTCs

To match packet sizes, removing symbols from the array shortens a product code. In general, rows or columns are removed until the appropriate size is reached. Codes selected shall have an integral number of information bytes. Shortening for the upstream 57-byte code in this specification. The exact shortening method used for codes other than the 57-byte upstream code is described in a following section entitled, "Method for determining codes for payload sizes different than the listed examples"

In the upstream section (~page303 ln 62), replace text and figures with the following:

Shortened Two -Dimensional BTC example

For example, the 57 byte BTC code in the upstream is composed of (32,26) constituent codes in the two dimensional array which has been shortened by seven rows and two columns to form a (30,24) x (25,19) array. This code block has 24 x 19 = 456 data bits, which is exactly 57 bytes. Fig. 5 shows the structure of the resultant block.

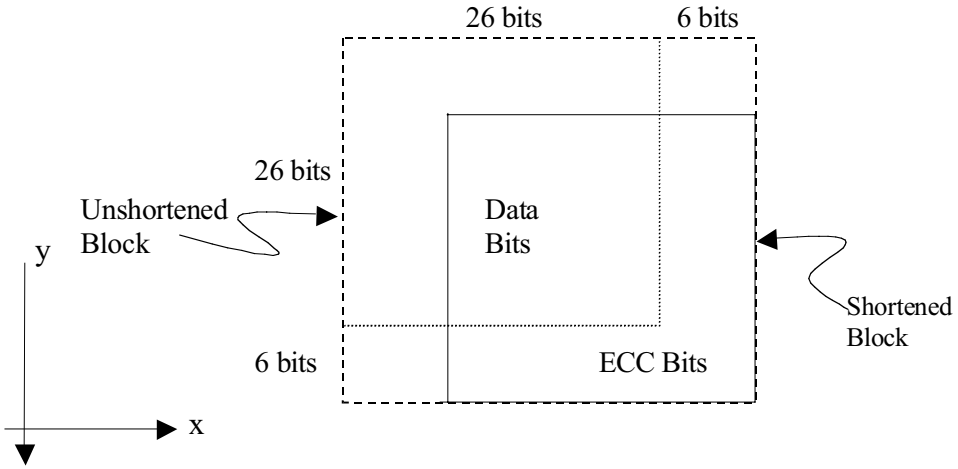

Fig. 5 Structure of Shortened 2 D Block

Modifications to the encoder to support shortening are minimal. Since shortened bits are always zero, and zeros input to the encoder LFSR result in a zero state, the shortened bits can simply be ignored for the purposes of encoding. The encoder simply needs to know how many bits per row to input to the row LFSR before shifting out the result. This also applies to the first row that has additional shortened bits. Similarly, it must know the number of columns to input to the column encoders.

Transmission of the resultant code block must start with the first data bit in the first row, proceed left to right and then row by row from top to bottom.

Add a section to address last code word shortening for the case when messages are not an integral number of codewords. Text as follows:

Shorten Last Codeword Mode

In the shortened last codeword mode, the last codeword is shortened to more closely match the required message length. The algorithm for this shortening is as follows:

1. Define a new code word that has the minimum number of rows that will carry the required number of information bits. The number of columns shall remain the same. In no case shall the last code word be shortened to less than 8 information rows.
2. If the number of the resultant codeword information bits (k') is greater than the number of bits needed to be transported (m). Add (k'-m) stuff bits (1's) to the end of the message.
3. Information bits and stuff bits (k') are randomized.

Comments on the above:

1. *This requires code changes on the fly for both the transmit and receive sides of the link.*
2. *The shortening process will result in the last code word having a lower code rate, which also results in better error protection for the last codeword.*