Global Journal of Pure and Applied Mathematics.
ISSN 0973-1768 Volume 11, Number 4 (2015), pp. 2755–2772

http://www.ripublication.com/gjpam.htm

Byte Oriented Error Locating Code

Ambika Tyagi
Research Scholar,
Department of Mathematics,
University of Delhi, Delhi 110027, India
E-mail: ambikajnu@gmail.com

Vinod Tyagi
Department of Mathematics,
Shyam Lal College (Eve.),
University of Delhi,
Shahdara, Delhi 110032, India.
E-mail: vinodtyagi@hotmail.com

Abstract

In this paper, we study e-error and burst error locating codes organized in bytes which can detect and locate a single byte among m bytes of length $t_1, t_2, \cdots t_m$; $t_1 + \cdots + t_n = n$, containing errors. In an (n, k) linear code, if there are m bytes of length β, then $n = m\beta$. By an e-error locating codes, we mean a code that can locate upto e-errors in a single byte. Similarly by a b-burst locating code we mean a code that can locate up to a burst of length b in one byte.

Keywords: Byte errors, error locating codes, parity check matrix, syndromes.

1. Introduction

Coding theory is the study of error detecting and error correcting codes and most of the studies are concentrated on these two aspects. Error location is a relatively later thought that lies midway between error detection and error correction. Codes possessing error locating abilities are referred to as error locating codes (ELC). The concept of ELC was introduced by Wolf and Elspas [10] in 1962 for codes where the block of length n is regarded as sub-divided into mutually exclusive sub-blocks. Error occurring within a particular sub-block is detected at the receiver and in addition, the receiver is able to determine which specific sub-block contains error.

In this paper, we investigate the property of error-location for byte-oriented codes. For such codes, when we say that the code is a e-byte locating code meaning that it detects a single error of length at most e within a byte of size β. So, byte locating codes are developed to detect and locate errors within a byte.

For a code of length n and dimension k, the redundancy r is $n - k$. A code is called a byte locating code if it can locate errors within a byte.

Tuvi Etzion [8] has defined five types of byte-correcting codes according to different sizes of the bytes viz.

Type 1: All bytes have the same size;

Type 2: One byte of size n_1 and other bytes of size n_2;

Type 3: Each byte is of either size n_1 or size n_2;

Type 4: The size of each byte is a power of 2;

Type 5: All the other cases;

Tyagi and Sethi [11] modified this classification by introducing size of the errors in a byte and gave eleven types of byte correcting codes.

Type 1: All bytes have the same size and the same burst length

- 1.1: All bytes have the same size and different burst lengths $b_i, i = 1, 2, \cdots, m$
- 1.2: All bytes have the same size but some have one burst length b_i for some i and others have other burst length b_j for some j; $b_i \neq b_j$.

2: One byte of size n_1 and other byte of size n_2 with same burst length

- 2.1: One byte of size n_1 and all other byte of size n_2 with different burst lengths
- 2.2: One byte of size n_1 with burst length b_1 and other bytes of size n_2 with burst length b_2

3: Each byte is of either size n_1 or size n_2 with same burst of length

- 3.1: Each byte is of either size n_1 or size $n_2$with different burst lengths
- 3.2: Each byte is of either size n_1 with burst length b_1 or size n_2 with burst length b_2

4: The size of each byte is a power of 2

5: All the other cases.

The paper is organized into two sections. In section 1, we discuss the possibilities of e-byte locating codes, whereas in section 2, we discuss b-byte locating codes

2. e-Byte Locating Codes

An e error-locating code (e-EL code) capable of detecting and locating a single byte among m bytes of lengths $t_1, t_2, \cdots t_m$ containing upto e-errors must satisfy the following two conditions:

(i) The syndrome resulting from the occurrence of e or fewer errors within any byte must be distinct from the zero syndrome.

(ii) The syndromes resulting from the occurrence of e or fewer errors within a byte of length t_i must be different from the syndrome resulting from any combination of e or fewer errors within any other byte of length t_j.

The first condition ensures detection of the error whereas the second condition assures the location of the byte containing errors.

We first give a necessary condition for the existence of e-byte locating codes.

We know that the largest number of distinct syndromes available using r check digits is q^r. First counting the number of syndromes that are required to be distinct by condition (i) and (ii) and then setting this number less than or equal to q^r.

For detection, the syndromes produced by e or fewer errors in a given byte must be distinct from any such syndromes likewise resulting from another set of upto e errors in the same byte or else there would exist a combination of atmost e errors in that byte resulting in the zero syndrome.

For location, any syndromes produced by a combination of upto e errors in any byte must be distinct from any other combination of upto e errors in any other byte; hence it will certainly be true for syndromes of upto e error whether in the same or in different byte.

Since there are $\binom{t_i}{j}$ combination of j errors out of t_i components in the i-th byte, therefore in all there must be

$$1 + \sum_{i=1}^{m} \sum_{j=1}^{e} \binom{t_i}{j} (q-1)^i \tag{2.1}$$

distinct syndromes including the zero syndrome.

Thus the necessary condition may now be given as follows:

Theorem 2.1. The number of parity-check digits r required for an e-byte locating code over GF(q) locating a single corrupted byte out of the m bytes of length $t_1, t_2, \cdots, t_m$; $t_1 + t_2 + \cdots + t_m = n$, is bounded from below by

$$r \geq \log_q \left[1 + \sum_{i=1}^{m} \sum_{j=1}^{e} \binom{t_i}{j} (q-1)^i \right] \tag{2.2}$$

Now we proceed to derive an upper bound that gives a sufficient condition for the existence of codes discussed in the previous theorem. The sufficient condition will be proved by constructing a parity check matrix H for the desired code.

Suppose we have added $t_1 + t_2 + \cdots + t_{m-1}$ columns corresponding to the first $m-1$ bytes and $i-1$ columns $h_1, h_2, \cdots, h_{i-1}$; corresponding to the m-th byte of H. In view of conditions (i) and (ii), the i-th column h_i to be added to the m-th byte of H must satisfy the following two conditions:

(i) h_i should not be a linear combination of any $(e-1)$ or fewer columns from amongst $h_1, h_2, \cdots, h_{i-1}$; and

(ii) h_i should not be a linear combination of any $(e-1)$ or fewer columns from amongst $h_1, h_2, \cdots, h_{i-1}$; together with a linear combination of any e or fewer columns corresponding to any one of the $(m-1)$ bytes.

Condition (i) gives rise to

$$1 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j \tag{2.3}$$

Possible linear combinations whereas the condition (ii) give rise to

$$\sum_{i=1}^{m-1} \sum_{j=1}^{e} \binom{t_i}{j} (q-1)^j \left(1 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j\right) \tag{2.4}$$

Possible linear combinations.

Thus the total number of linear combinations, to which h_i can not be equal, is

$$\left[1+\sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j + \sum_{i=1}^{m-1} \sum_{j=1}^{e} \binom{t_i}{j} (q-1)^j \left(1+\sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j\right)\right] \tag{2.5}$$

At worst, all these linear combinations might yield a distinct sum. Therefore, the column can be added to H provided that

$$q^r \geq 1 + \left[1 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j + \sum_{i=1}^{m-1} \sum_{j=1}^{e} \binom{t_i}{j} \cdot (q-1)^j \left(1 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j\right)\right] \tag{2.6}$$

$$r \geq \log_q \left[2 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j + \sum_{i=1}^{m-1} \sum_{j=1}^{e} \binom{t_i}{j} \cdot (q-1)^j \left(1 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j\right)\right] \tag{2.7}$$

So the sufficient condition may now be given as follows:

Theorem 2.2. An (n, k) linear code over GF(q) capable of detecting e or fewer errors occurring within a single byte and locating that byte from amongst a byte of lengths $t_1, t_2, \cdots, t_m$; $t_1 + t_2 + \cdots + t_m = n$; $e \leq t_i$ for all i can always be constructed using check-digits where r is the smallest integer satisfying the inequality

$$r \geq \log_q \Bigg[2 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j + \sum_{i=1}^{m-1} \sum_{j=1}^{e} \binom{t_i}{j} \cdot (q-1)^j \left(1 + \sum_{j=1}^{e-1} \binom{t_m - 1}{j} (q-1)^j \right) \Bigg] \tag{2.8}$$

3. Discussion

We discuss below different possibilities depending upon the size of bytes as mentioned by Tuvi [8]:

Type 1: All bytes have the same size.
Taking $q = 2$, $m = 3$, $e = 2$ $(t_1 = t_2 = t_3 = 5)$, $n = 15$, $k = 9$, the expression in (2.2) comes out to be

$$2^6 \geq 1 + \sum_{i=1}^{3} \sum_{j=1}^{2} \binom{t_i}{j} \tag{3.1}$$

$$\Rightarrow \quad 2^6 \geq 46\,. \tag{3.2}$$

Now, we try to construct (15, 9) byte locating code of type 1 in the following example.

Example 3.1. Consider a parity-check matrix for a (15, 9) binary code with

$$H = \begin{bmatrix} 00000 & 01111 & 00111 \\ 00000 & 10101 & 11001 \\ 00001 & 11100 & 01100 \\ 00110 & 00011 & 11000 \\ 01010 & 00010 & 10101 \\ 11110 & 00000 & 11110 \end{bmatrix}$$

This matrix has been constructed by the synthesis procedure given above by taking $t_1 = t_2 = t_3 = 5$; $n = 15$, $(t_1 + t_2 + t_3 = n)$ and $e = 2$. It can be verified from Table 1 that the syndromes of all the error patterns of weight 2 or less belonging to all the byte of length 5 are nonzero, and the syndromes of the error patterns belonging to the first byte are altogether different from the syndromes of the error patterns belonging to the next byte. This shows that this code is a 2-byte locating code.

Table 1:

Error Pattern	**Syndrome**
10000 00000 00000	000001
01000 00000 00000	000011
00100 00000 00000	000101
00010 00000 00000	000111
00001 00000 00000	001000
11000 00000 00000	000010
10100 00000 00000	000100
10010 00000 00000	000110
10001 00000 00000	001001
01100 00000 00000	000110
01010 00000 00000	000100
01001 00000 00000	001011
00110 00000 00000	000010
00101 00000 00000	001101
00011 00000 00000	001111
00000 10000 00000	011000
00000 01000 00000	101000
00000 00100 00000	111000
00000 00010 00000	100110
00000 00001 00000	110100
00000 11000 00000	110000
00000 10100 00000	100000
00000 10010 00000	111110
00000 10001 00000	101100
00000 01100 00000	010000
00000 01010 00000	001110
00000 01001 00000	011110
00000 00110 00000	011100
00000 00101 00000	001100
00000 00011 00000	010010
00000 00000 10000	010111
00000 00000 01000	011101
00000 00000 00100	101011
00000 00000 00010	100001
00000 00000 00001	110010
00000 00000 11000	001010
00000 00000 10100	111100
00000 00000 10010	110110
00000 00000 10001	100101
00000 00000 01100	110110
00000 00000 01010	111100
00000 00000 01001	101111
00000 00000 00110	001010
00000 00000 00101	011001
00000 00000 00011	010011

This shows the existence of type 1 codes mentioned by Tuvi [8].
Now we discuss type 2 codes.

Type 2: One byte of size t_1 and all other bytes are of the same size.
Taking $q = 2$, $m = 3$, $e = 2$, ($t_1 = 4$, $t_2 = t_3 = 5$), $n = 14$, $k = 8$, the expression in (2.2) comes out to be

$$2^6 \geq 1 + \sum_{i=1}^{3} \sum_{j=1}^{2} \binom{t_i}{j} \tag{3.3}$$

$$\Rightarrow \quad 2^6 \geq 41. \tag{3.4}$$

Now, we try to construct (14, 8) byte locating code of type 2 in the following example.

Example 3.2. Consider a (14, 8) binary codes with parity-check matrix

$$H = \begin{bmatrix} 0000 & 00111 & 11010 \\ 0000 & 01010 & 01001 \\ 0000 & 11110 & 00111 \\ 0011 & 00001 & 10001 \\ 0101 & 00001 & 01011 \\ 1111 & 00000 & 00111 \end{bmatrix}$$

This matrix has been constructed by the synthesis procedure in the proof of Theorem 2 by taking $t_1 = 4$, $t_2 = 5$, $t_3 = 5$; $n = 14$, ($t_1 + t_2 + t_3 = n$) and $e = 2$. It can be seen from Table 2 that the syndromes of all the error patterns of weight 2 or less belonging either to the first byte of length 3 or to the other two bytes of length 4 are nonzero, and the syndromes of the error patterns belonging to the first byte are altogether different from the syndromes of the error patterns belonging to the next byte. This shows that this code is a 2-byte locating code.

Table 2:

Error Pattern	**Syndrome**
1000 00000 00000	000001
0100 00000 00000	000011
0010 00000 00000	000101
0001 00000 00000	000111
1100 00000 00000	000010
1010 00000 00000	000100
1001 00000 00000	000110
0110 00000 00000	000110
0101 00000 00000	000100
0011 00000 00000	000010

Table 2: Continued.

Error Pattern	**Syndrome**
0000 10000 00000	001000
0000 01000 00000	011000
0000 00100 00000	101000
0000 00010 00000	111000
0000 00001 00000	100110
0000 11000 00000	010000
0000 10100 00000	100000
0000 10010 00000	110000
0000 10001 00000	101110
0000 01100 00000	110000
0000 01010 00000	100000
0000 01001 00000	111110
0000 00110 00000	010000
0000 00101 00000	001110
0000 00011 00000	011110
0000 00000 10000	100100
0000 00000 01000	110010
0000 00000 00100	001001
0000 00000 00010	101011
0000 00000 00001	011111
0000 00000 11000	010110
0000 00000 10100	101101
0000 00000 10010	001111
0000 00000 10001	111011
0000 00000 01100	111011
0000 00000 01010	011001
0000 00000 01001	101101
0000 00000 00110	100010
0000 00000 00101	010110
0000 00000 00011	110100

This shows the existence of type 2 codes considered by Tuvi.
Now we discuss type 3 codes.

Type 3: Each byte is of either size $t_1, t_2 \ldots$;
Taking $q = 2$, $m = 3$, $e = 2$ ($t_1 = 4$, $t_2 = 5$, $t_3 = 6$), $n = 15$, $k = 9$, the expression in (2.2) comes out to be

$$2^6 \geq 1 + \sum_{i=1}^{3} \sum_{j=1}^{2} \binom{t_i}{j} \tag{3.5}$$

$$\Rightarrow \quad 2^6 \geq 47\,. \tag{3.6}$$

Now, we try to construct (15, 9) byte locating code of type 3 in the following example.

Example 3.3. Consider a (15, 9) binary codes with parity-check matrix

$$H = \begin{bmatrix} 0000 & 00111 & 110100 \\ 0000 & 01010 & 010011 \\ 0000 & 11110 & 001110 \\ 0011 & 00001 & 100011 \\ 0101 & 00001 & 010111 \\ 1111 & 00000 & 001110 \end{bmatrix}$$

This matrix has been constructed by the synthesis procedure in the proof of Theorem 2 by taking $t_1 = 4$, $t_2 = 5$, $t_3 = 6$; $n = 15$, $(t_1 + t_2 + t_3 = n)$ and $e = 2$. It can be seen from Table 3 that the syndromes of all the error patterns of weight 2 or less belonging to the first byte of length 4, to the second byte of length 5 and to the third byte of length 6 are nonzero, and the syndromes of the error patterns belonging to the first byte are altogether different from the syndromes of the error patterns belonging to the others byte. This shows that this code is a 2-byte locating code.

Table 3:

Error Pattern	**Syndrome**
1000 00000 000000	000001
0100 00000 000000	000011
0010 00000 000000	000101
0001 00000 000000	000111
1100 00000 000000	000010
1010 00000 000000	000100
1001 00000 000000	000110
0110 00000 000000	000110
0101 00000 000000	000100
0011 00000 000000	000010
0000 10000 000000	001000
0000 01000 000000	011000
0000 00100 000000	101000
0000 00010 000000	111000
0000 00001 000000	100110
0000 11000 000000	010000
0000 10100 000000	100000
0000 10010 000000	110000
0000 10001 000000	101110
0000 01100 000000	110000
0000 01010 000000	100000
0000 01001 000000	111110
0000 00110 000000	010000
0000 00101 000000	001110
0000 00011 000000	011110

Table 3: Continued.

Error Pattern	Syndrome
0000 00000 100000	100100
0000 00000 010000	110010
0000 00000 001000	001001
0000 00000 000100	101110
0000 00000 000010	011111
0000 00000 000001	010101
0000 00000 110000	010110
0000 00000 101000	101101
0000 00000 100100	011111
0000 00000 100010	111011
0000 00000 100001	110010
0000 00000 011000	111011
0000 00000 010100	011001
0000 00000 010010	101101
0000 00000 010001	100100
0000 00000 001100	100010
0000 00000 001010	010110
0000 00000 001001	011111
0000 00000 000110	110100
0000 00000 000101	111101
0000 00000 000011	001001

4. b-Byte Locating Codes

A b-byte locating code (b-BEL code) capable of detecting and locating a burst of length b or less within a single byte among m bytes of lengths $t_1, t_2, \cdots, t_m$ containing burst errors must satisfy the following two conditions:

(i) The syndrome resulting from the occurrence of a burst of length b or less within any byte must be distinct from the zero syndrome.

(ii) The syndromes resulting from the occurrence of a burst of length b or less within any byte of length t_i must be different from the syndrome resulting from any combination of other burst of length b or less occurring within any other byte of length t_j.

We first give a necessary condition for the existence of b-byte locating codes.

Let us fix up the first b position in every byte.

For detection, no two vectors that are zero except in their first b symbols of the same byte could be in the same coset because if they were, their difference, which would be a burst of length b or less shall become a code vector and this violates condition (i).

For location, the syndromes of any burst of length b or less within a byte must be different from the syndrome of another burst of length b or less within any other byte; in particular, this should be true for those burst s which are confined to the first b positions of any byte. Thus bursts of length b or less confined to the first b positions whether belonging to the same byte or in different bytes should lie in different cosets.

The total number of such bursts, including the pattern of all zeros, is

$$1 + \left[b \sum_{i=1}^{m} (t_i)(q-1)^i \right] - m \tag{4.1}$$

Since this number should atmost be equal to the total number of cosets, i.e. q^r.

Thus the necessary condition may now be given as follows:

Theorem 4.1. The number of parity-check digits r required for a b-byte error locating code over GF(q) locating a single corrupted byte out of the m byte of length $t_1, t_2, \cdots, t_m$; $t_1 + t_2 + \cdots + t_m = n$, is bounded from below by

$$r \geq \log_q \left[1 + b \sum_{i=1}^{m} (t_i)(q-1)^i \right] - m \,. \tag{4.2}$$

Now we proceed to derive an upper bound that gives a sufficient condition for the existence of codes discussed in the previous theorem. The sufficient condition will be proved by constructing a parity check matrix H for the desired code.

Suppose we have added $t_1 + t_2 + \cdots + t_{m-1}$ columns corresponding to the first $m-1$ bytes and $i-1$ columns $h_1, h_2, \cdots, h_{i-1}$; corresponding to the m-th byte of H. In view of conditions (i) and (ii), the i-th column h_i to be added to the m-byte of H must satisfy the following two conditions:

(i) h_i should not be a linear combination of any $(b-1)$ or less columns of m-th byte,

(ii) h_i should not be a linear combination of any $(b-1)$ or less columns of the m-th byte together with a linear combination of any b or less columns corresponding to any one of the remaining $(m-1)$ bytes.

Condition (i) gives rise to

$$q^{(b-1)} \tag{4.3}$$

Possible linear combinations whereas the condition (ii) give rise to

$$\sum_{i=1}^{m-1} q^{(b-1)} [q^{(b-1)}((t_i - b + 1)(q-1) + 1) - 1] \tag{4.4}$$

Possible linear combinations.

Thus the total number of linear combinations, to which h_i can not be equal, is

$$q^{(b-1)} + \sum_{i=1}^{m-1} q^{(b-1)} [q^{(b-1)}((t_i - b + 1)(q-1) + 1) - 1] \tag{4.5}$$

At worst, all these linear combinations might yield a distinct sum. Therefore, the column can be added to H provided that

$$q^r \geq \left[1 + \sum_{i=1}^{m-1} [q^{(b-1)}((t_i - b + 1)(q-1) + 1) - 1]\right] q^{(b-1)} \tag{4.6}$$

$$r \geq \log_q \left[1 + \sum_{i=1}^{m-1} [q^{(b-1)}((t_i - b + 1)(q-1) + 1) - 1]\right] q^{(b-1)} \tag{4.7}$$

So the sufficient condition may now be given as follows:

Theorem 4.2. An (n, k) linear code over GF(q) capable of detecting a burst of length b or less, $b \leq t_i$ $(i = 1, 2, \cdots, m)$ within a single byte amongst m bytes of lengths $t_1, t_2, \cdots, t_m$; $t_1 + t_2 + \cdots + t_m = n$; for all i and locating that byte can always be constructed using check-digits where r is the smallest integer satisfying the inequality

$$r \geq \log_q \left[1 + \sum_{i=1}^{m-1} [q^{(b-1)}((t_i - b + 1)(q-1) + 1) - 1]\right] q^{(b-1)} \tag{4.8}$$

5. Discussion

We now discuss different possibilities for BEL codes depending upon the size of bytes as mentioned by Tuvi [8]:

Type 1: All bytes have the same size and the same burst length.
Taking $q = 2, m = 3, b = 2$ $(t_1 = t_2 = t_3 = 5), n = 15, k = 10$, the expression in (4.2) comes out to be

$$2^5 \geq \left[1 + 2\sum_{i=1}^{3} (t_i)(2-1)^i\right] - 3 \tag{5.1}$$

$$\Rightarrow \quad 2^5 \geq 28. \tag{5.2}$$

Now, we try to construct (15, 10) byte locating code of type 1 in the following example.

Example 5.1. Consider a parity-check matrix for a (15, 10) binary code with

$$H = \begin{bmatrix} 00000 & 00111 & 10111 \\ 00001 & 01010 & 01101 \\ 00110 & 11110 & 10001 \\ 01010 & 00001 & 01011 \\ 11110 & 00000 & 10110 \end{bmatrix}$$

This matrix has been constructed by the synthesis procedure given above by taking $t_1 = t_2 = t_3 = 5$; $n = 15$, $(t_1 + t_2 + t_3 = n)$ and $b = 2$. It can be seen from Table 4 that the syndromes of all the error patterns of burst of length 2 or less belonging to all the bytes of length 5 are nonzero, and the syndromes of the error patterns belonging to the first byte are altogether different from the syndromes of the error patterns belonging to the next byte. This shows that this code is a 2-burst error locating code.

Table 4:

Error Pattern	Syndrome
10000 00000 00000	00001
01000 00000 00000	00011
00100 00000 00000	00101
00010 00000 00000	00111
00001 00000 00000	01000
11000 00000 00000	00010
01100 00000 00000	00110
00110 00000 00000	00010
00011 00000 00000	01111
00000 10000 00000	00100
00000 01000 00000	01100
00000 00100 00000	10100
00000 00010 00000	11100
00000 00001 00000	10000
00000 11000 00000	01000
00000 01100 00000	11000
00000 00110 00000	01000
00000 00011 00000	01100
00000 00000 10000	10101
00000 00000 01000	01010
00000 00000 00100	11001
00000 00000 00010	10011
00000 00000 00001	11110
00000 00000 11000	11111
00000 00000 01100	10011
00000 00000 00110	01010
00000 00000 00011	01101

This shows the existence of type 1 codes mentioned by Tuvi [8].

Now we discuss type 2 codes.

Type 2: One byte is of size t_1 and all other bytes are of the same size with same burst length.

Taking $q = 2$, $m = 3$, $b = 2$ ($t_1 = 3$, $t_2 = t_3 = 4$), $n = 11$, $k = 6$, the expression in (4.2) comes out to be

$$2^5 \geq \left[1 + 2 \sum_{i=1}^{3} (t_i)(2-1)^i \right] - 2 \tag{5.3}$$

$$\Rightarrow \quad 2^5 \geq 20. \tag{5.4}$$

Now, we try to construct (11, 6) byte locating code of type 2 in the following example.

Example 5.2. Consider a (11, 6) binary codes with parity-check matrix

$$H = \begin{bmatrix} 011 & 0000 & 0011 \\ 101 & 0000 & 0101 \\ 011 & 0011 & 1111 \\ 101 & 0101 & 0000 \\ 011 & 1111 & 0000 \end{bmatrix}$$

This matrix has been constructed by the synthesis procedure in the proof of Theorem 4.1 by taking $t_1 = 3$, $t_2 = 4$, $t_3 = 4$; $n = 11$, $(t_1 + t_2 + t_3 = n)$ and $b = 2$. It can be seen from Table 5 that the syndromes of all the error patterns of burst 2 or less belonging either to the first byte of length 3 or to the other two bytes of length 4 are nonzero, and the syndromes of the error patterns belonging to the first byte are altogether different from the syndromes of the error patterns belonging to the next byte. This shows that this code is a 2-burst error locating code.

Table 5:

Error Pattern	**Syndrome**
100 0000 0000	01010
010 0000 0000	10101
001 0000 0000	11111
110 0000 0000	11111
011 0000 0000	01010
000 1000 0000	00001
000 0100 0000	00011
000 0010 0000	00101
000 0001 0000	00111
000 1100 0000	00010
000 0110 0000	00110
000 0011 0000	00010
000 0000 1000	00100
000 0000 0100	01100
000 0000 0010	10100
000 0000 0001	11100
000 0000 1100	01000
000 0000 0110	11000
000 0000 0011	01000

This shows the existence of type 2 codes considered by Tuvi.

Now we discuss type 3 codes.

Type 3: Each byte is of either size t_1, t_2, . . . ; with same burst length.

Taking $q = 2, m = 3, b = 2$ ($t_1 = 4, t_2 = 5, t_3 = 6$), $n = 15, k = 10$, the expression in (4.2) comes out to be

$$2^5 \geq \left[1 + 2\sum_{i=1}^{3}(t_i)(2-1)^i\right] - 3 \tag{5.5}$$

$$\Rightarrow \quad 2^5 \geq 28. \tag{5.6}$$

Now, we try to construct (15, 10) byte locating code of type 3 in the following example.

Example 5.3. Consider a (15, 10) binary codes with parity-check matrix

$$H = \begin{bmatrix} 1101 & 00000 & 011101 \\ 0111 & 00000 & 101011 \\ 0011 & 00110 & 011000 \\ 0000 & 01111 & 101111 \\ 0000 & 11010 & 011010 \end{bmatrix}$$

This matrix has been constructed by the synthesis procedure in the proof of Theorem 4.1 by taking $t_1 = 4$, $t_2 = 5$, $t_3 = 6$; $n = 15$, ($t_1 + t_2 + t_3 = n$) and $b = 2$. It can be seen from Table **??** that the syndromes of all the error patterns of burst 2 or less belonging to the first byte of length 4, to the second byte of length 5 and to the third byte of length 6 are nonzero, and the syndromes of the error patterns belonging to the first byte are altogether different from the syndromes of the error patterns belonging to the others byte. This shows that this code is a 2-burst error locating code.

Table 6:

Error Pattern	**Syndrome**
1000 00000 000000	10000
0100 00000 000000	11000
0010 00000 000000	01100
0001 00000 000000	11100
1100 00000 000000	01000
0110 00000 000000	10100
0011 00000 000000	10000
0000 10000 000000	00001
0000 01000 000000	00011
0000 00100 000000	00110
0000 00010 000000	00111
0000 00001 000000	00010
0000 11000 000000	00010
0000 01100 000000	00101
0000 00110 000000	00001
0000 00011 000000	00101

Table 6: Continued.

Error Pattern	Syndrome
0000 00000 100000	01010
0000 00000 010000	10101
0000 00000 001000	11111
0000 00000 000100	10010
0000 00000 000010	01011
0000 00000 000001	11010
0000 00000 110000	11111
0000 00000 011000	01010
0000 00000 001100	01101
0000 00000 000110	11001
0000 00000 000011	10001

This shows the existence of type 3 codes considered by Tuvi.

Now we discuss type 4 code.

Type 4: The size of each byte is a power of 2 with same burst length
Taking $q = 2$, $m = 3$, $b = 2$ ($t_1 = t_2 = t_3 = 8 = 2^3$), $n = 24$, $k = 18$, the expression in (4.2) comes out to be

$$2^6 \geq \left[1 + 2\sum_{i=1}^{3}(t_i)(2-1)^i \right] - 3 \tag{5.7}$$

$$\Rightarrow \quad 2^6 \geq 46. \tag{5.8}$$

Now, we try to construct (24, 18) byte locating code of type 4 in the following example.

Example 5.4. Consider a (24, 18) binary codes with parity-check matrix

$$H = \begin{bmatrix} 00011111 & 00000000 & 10101011 \\ 01101010 & 00000000 & 11000101 \\ 10100110 & 00000011 & 00111101 \\ 00000011 & 00011110 & 00000101 \\ 00000000 & 01101010 & 01011101 \\ 00000000 & 10100111 & 10110011 \end{bmatrix}$$

This matrix has been constructed by the synthesis procedure in the proof of Theorem 4.1 by taking $t_1 = t_2 = t_3 = 8$; $n = 24$, ($t_1 + t_2 + t_3 = n$) and $b = 2$. It can be seen from Table 7 that the syndromes of all the error patterns of burst 2 or less belonging to all bytes of the length 2^3 are nonzero, and the syndromes of the error patterns belonging to the first byte are altogether different from the syndromes of the error patterns belonging to the others byte. This shows that this code is a 2-burst error locating code.

Table 7:

Error Pattern	**Syndrome**
10000000 00000000 00000000	001000
01000000 00000000 00000000	010000
00100000 00000000 00000000	011000
00010000 00000000 00000000	100000
00001000 00000000 00000000	110000
00000100 00000000 00000000	101000
00000010 00000000 00000000	111100
00000001 00000000 00000000	100100
11000000 00000000 00000000	011000
01100000 00000000 00000000	001000
00110000 00000000 00000000	111000
00011000 00000000 00000000	010000
00001100 00000000 00000000	011000
00000110 00000000 00000000	010100
00000011 00000000 00000000	011000
00000000 10000000 00000000	000001
00000000 01000000 00000000	000010
00000000 00100000 00000000	000011
00000000 00010000 00000000	000100
00000000 00001000 00000000	000110
00000000 00000100 00000000	000101
00000000 00000010 00000000	001111
00000000 00000001 00000000	001001
00000000 11000000 00000000	000011
00000000 01100000 00000000	000001
00000000 00110000 00000000	000111
00000000 00011000 00000000	000010
00000000 00001100 00000000	000011
00000000 00000110 00000000	001010
00000000 00000011 00000000	000110
00000000 00000000 10000000	110001
00000000 00000000 01000000	010010
00000000 00000000 00100000	101001
00000000 00000000 00010000	001011
00000000 00000000 00001000	101010
00000000 00000000 00000100	011110
00000000 00000000 00000010	100001
00000000 00000000 00000001	111111
00000000 00000000 11000000	100011
00000000 00000000 01100000	111011
00000000 00000000 00110000	100010
00000000 00000000 00011000	100001
00000000 00000000 00001100	110100
00000000 00000000 00000110	111111
00000000 00000000 00000011	011110

This shows the existence of type 4 codes considered by Tuvi.

Acknowledgement

The authors are thankful to Prof. B.K. Dass, Department of Mathematics, University of Delhi for his helpful suggestions during the preparation of this paper.

References

[1] I. M. Boyarinov, On error locating codes, *Probl. Control Info. Theo.* **1** (1972) 277–286.

[2] S. H. Chang and L. T. Wang, Error locating codes, *IEEE Int. Conven. Rec.* **Part 7** (1965) 252–258.

[3] C. L. Chen, Error correcting codes with byte error detection capabilities, *IEEE Trans. Comput.* **C-32** (1983) 615–621.

[4] C. L. Chen, Byte oriented error correcting codes with semiconductor memory system, *IEEE Trans. Comput.* **C-35** (1986) 646–648.

[5] B. K. Dass, On a burst-error-correcting code, *J. Inf. Opt. Sciences* **1** (1980) 291–295.

[6] B. K. Dass, Burst error locating linear codes, *J. Inf. Opt. Sciences* **3** (1) (1982) 77–80.

[7] B. K. Dass and S. K. Muttoo, On the correction of random and burst errors, *IEE Part E*, CDT **127** (6) (1980) 249–252.

[8] Tuvi Etzion, Perfect byte correcting codes, *IEEE Trans. Inform. Theory* **44** (1998) 3140–3146.

[9] W. W. Peterson and E. J. Weldon, *Error Correcting Codes*, 2nd, MIT Press (1972).

[10] J. K. Wolf and B. Elspas, Error-locating-codes – a new concept in error control, *IEEE Trans.* **IT-9** (1963) 113–117.

[11] V.K. Tyagi and Amita Sethi, b_i-Byte Correcting Perfect Codes, *Asian Europeian Journal of Mathematics* **7** (1) (2014) 1–8.