Chapter 2
Asymmetric CT-Burst Correcting Integer Codes

The contents of this chapter are based on the papers mentioned below:

- Das, P. K. and Pokhrel, N. K. Asymmetric CT-burst correcting integer codes. In *2021 5th International Conference on Information Systems and Computer Networks (ISCON)*, 1-5, IEEE, 2021, doi:10.1109/ISCON52037.2021.9702506.
- Pokhrel, N.K. and Das, P.K. Probability of erroneous decoding for integer codes correcting burst asymmetric/unidirectional/symmetric errors within a byte and up to double asymmetric errors between two bytes. *Kuwait Journal of Science*, 2022, doi: 10.48129/kjs.online.

Chapter 2

Asymmetric CT-Burst Correcting Integer Codes

2.1 Overview

As mentioned in Section 1.3, in some systems like optical networks without optical amplifiers, the likelihood of the type $1 \to 0$ is significantly higher than $0 \to 1$ due to the number of received photons never exceeding the number of sent ones. This gives rise to the concept of a Z-channel (refer Figure 1.1). In a binary asymmetric channel, the probability of $0 \to 1$ is zero, so a sent binary message will undergo an error only if $1 \to 0$ occurs. The crossover probability ϵ varies from channel to channel.

To employ a code's error-detecting and correcting mechanism in a communication channel, we frequently require sufficient information about the likelihood of error patterns occurring. To know this, we require the probability of erroneous decoding.

Influenced by these facts, we have presented a class of integer codes capable of correcting asymmetric CT-bursts of length l within a b-bit byte and we name the codes by integer $(CT_lB)_b$ codes. Encoding and decoding of the codes are presented in Section 2.2. In Section 2.3-2.4, we give the implementation and comparison of the codes with existing similar types of codes to justify our study. In Section 2.5, we derive the probability of erroneous decoding for these codes, followed by a ratio for undetected error. This approach can also be used to determine the probability of similar integer codes. Finally Section 2.6 concludes this chapter.

2.2 Construction of codes

For the construction of the codes discussed, we use Definition 1.12 with $m = 2^b - 1, M = 1$ and $N = k + 1$. Because the binary representation of all non-trivial elements in the ring $\mathbb{Z}_{2^b-1}$ is unique, we chose $m = 2^b - 1$. Here, b-bit byte $(x_0, x_1, x_2, \ldots, x_{b-1})$ with $x_i \in \{0, 1\}$ is uniquely represented as $[x_0 2^0 + x_1 2^1 + x_2 2^2 + \ldots + x_{b-1} 2^{b-1}] \pmod{2^b - 1}$. By considering the error pattern $1 \to 0$, $B = (x_0, x_1, \ldots, x_{b-1})$ (sent) and $\bar{B} = (\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_{b-1})$ (received) in Definition 1.5 with $x_i, \bar{x}_i \in \{0, 1\}$, the collection of all asymmetric CT-bursts of length l beginning from the 1^{st} position will be $e^1_{b,l} = \Big\{2^0 + p_1 2^1 + p_2 2^2 + \ldots + p_{l-1} 2^{l-1} \,|\, p_1, p_2, \ldots, p_{l-1} \in \{0, 1\}\Big\}$. Similarly for asymmetric CT-bursts beginning from 2^{nd} position, the collection will be $e^2_{b,l} = \Big\{2^1 + p_1 2^2 + p_2 2^3 + \ldots + p_{l-1} 2^l \,|\, p_1, p_2, \ldots, p_{l-1} \in \{0, 1\}\Big\}$, continuing this pattern, we get $e^i_{b,l} = \Big\{2^{i-1} + p_1 2^i + p_2 2^{i+1} + \ldots + p_{l-1} 2^{i+l-2} \,|\, p_1, p_2, \ldots, p_{l-1} \in \{0, 1\}\Big\}$ as the error set for asymmetric CT-bursts of length l beginning from i^{th} position, where $1 \le i \le b - l + 1$.

For $i > b - l + 1$, asymmetric CT-bursts of length l will have less than l positions to occur. In this case, the set of all possible asymmetric CT-bursts of length l beginning from i^{th} position, where $b - l + 1 < i \le b$, will be $\overline{e^i_{b,l}} = \Big\{2^{i-1} + p_1 2^i + p_2 2^{i+1} + \ldots + p_{b-i} 2^{b-1} \,|\, p_1, p_2, \ldots, p_{b-i} \in \{0, 1\}\Big\}$. Define $e_{b,l} = \bigcup_{i=1}^{b-l+1} e^i_{b,l}$ and $\overline{e_{b,l}} = \bigcup_{j=b-l+2}^{b} \overline{e^j_{b,l}}$, then $\epsilon_{b,l} = e_{b,l} \cup \overline{e_{b,l}}$ gives us the collection of all possible asymmetric CT-bursts of length l within a b-bit byte.

2.2.1 Encoding procedure

To encode integer $(CT_lB)_b$ codes, we choose k distinct coefficients $C_1, C_2, \ldots, C_k$ from the set $\mathbb{Z}_{2^b-1} \setminus \{0, 1\}$ with the help of some computer search results. Consider $N = k + 1$, $M = 1$ and predefined $H = (C_1 \; C_2 \; \ldots \; C_k \; -1)$ in Definition 1.12, then an encoded codeword of integer $(CT_lB)_b$ code will be from the set $\Big\{(B_1 B_2 \ldots B_k C_B) \,|\, (B_1 B_2 \ldots B_k C_B) \Big[C_1 \; C_2 \; \ldots \; C_k \; -1\Big]^T \pmod{2^b - 1} = 0\Big\}$, where

T represents the transpose of matrix. The last b-bit byte C_B is called **check byte** and is written as $C_B = [C_1B_1 + C_2B_2 + \ldots + C_kB_k] \pmod{2^b - 1}$. In simple words, the message $B_1B_2 \ldots B_k$ is encoded as $B_1B_2 \ldots B_kC_B$.

Considering $c = B_1B_2 \ldots B_kC_B$ and $r = \bar{B}_1\bar{B}_2 \ldots \bar{B}_k\bar{C}_B$ to be the sent and received messages respectively, we have error $e = c - r$. Thus the **syndrome** for r will be

$$
\begin{aligned}
S(r) &= [(c - e)H^T] \pmod{2^b - 1} \\
&= [-eH^T] \pmod{2^b - 1} \\
&= -\Big[(B_1B_2 \ldots B_kC_B) - (\bar{B}_1\bar{B}_2 \ldots \bar{B}_k\bar{C}_B)\Big]\Big[C_1 \ C_2 \ \ldots \ C_k \ -1\Big]^T \pmod{2^b - 1} \\
&= -[C_1B_1 + \ldots + C_kB_k - C_B - C_1\bar{B}_1 - \ldots - C_k\bar{B}_k + \bar{C}_B] \pmod{2^b - 1} \\
&= [C_1\bar{B}_1 + \ldots + C_k\bar{B}_k - \bar{C}_B] \pmod{2^b - 1} \\
&= [C_{\bar{B}} - \bar{C}_B] \pmod{2^b - 1}.
\end{aligned}
$$

Keeping this in mind, we introduce the set of syndromes for integer $(CT_lB)_b$ codes as below.

Definition 2.1. *The set of syndromes for integer codes correcting asymmetric CT-bursts of length l within a b-bit byte will be*

$$S_1 = \bigcup_{i=1}^{k+1} [-C_i\epsilon_{b,l}] \pmod{2^b - 1}, \tag{2.1}$$

where $C_{k+1} = -1$, and other coefficient C_i's are picked from $\mathbb{Z}_{2^b-1} \setminus \{0, 1\}$ such that the sets $-C_1\epsilon_{b,l} \pmod{2^b - 1}, -C_2\epsilon_{b,l} \pmod{2^b - 1}, \ldots, -C_k\epsilon_{b,l} \pmod{2^b - 1}$ and $\epsilon_{b,l} \pmod{2^b - 1}$ are mutually disjoint. Appendix A consists of a Python programme used to find the coefficients.

While representing an asymmetric CT-burst, we choose distinct components every time and as each element in the ring $\mathbb{Z}_{2^b-1}$ has a unique binary representation, thus the error set $\epsilon_{b,l}$ will not have any repetition consequently S_1 as well. Theorem below gives the number of elements in the syndrome set.

Theorem 2.2. *A $((k + 1)b, kb)$ integer $(CT_lB)_b$ code can correct asymmetric CT-bursts of length l within a b-bit byte if there exist k distinct coefficient C_i's from the set $\mathbb{Z}_{2^b-1} \setminus \{0, 1\}$ such that $|S_1| = (k + 1)[2^{l-1}(b - l + 2) - 1]$.*

Proof. As the process of choosing coefficient C_i's requires the distinctness of the sets $\epsilon_{b,l} \pmod{2^b - 1}$ and $-C_i\epsilon_{b,l} \pmod{2^b - 1}$ for $1 \leq i \leq k$, also each representation is unique, so all of the sets above will have the same cardinality. Thus to prove the result, we show that $|\epsilon_{b,l}| = 2^{l-1}(b-l+2)-1$. Clearly $|e^i_{b,l}| = \binom{l-1}{0}+\binom{l-1}{1}+\ldots+\binom{l-1}{l-1}$, there are $b-l+1$ number of beginning positions for asymmetric CT-bursts having length l. So $|e_{b,l}| = (b-l+1)\sum_{i=0}^{l-1}\binom{l-1}{i} = (b-l+1)2^{l-1}$. For the remaining beginning positions, the cardinality will be $|\overline{e_{b,l}}| = \binom{l-2}{0} + \binom{l-2}{1} + \ldots + \binom{l-2}{l-2} + \binom{l-3}{0} + \binom{l-3}{1} + \ldots + \binom{l-3}{l-3} + \ldots + \binom{1}{0} = 2^{l-2} + 2^{l-3} + \ldots + 2^0 = 2^{l-1} - 1$. Thus by adding we get $|\epsilon_{b,l}| = (b-l+1)2^{l-1} + 2^{l-1} - 1 = 2^{l-1}(b-l+2) - 1$. This proves our claim. □

2.2.2 Decoding procedure

The decoder constructs a look up table, LUT_2 consisting of all syndrome elements using (2.1) whereas look up table, LUT_1 comprises of the coefficient C_i's. Each entry from LUT_2 is of $2b + \lceil \log_2(k+1) \rceil$ bits, so the size of LUT_2 will be $|S_1| \times (2b + \lceil \log_2(k+1) \rceil)$ bits. Figure 2.1 depicts the bit width of each syndrome entry. Once

Figure 2.1: **Bit width of each syndrome entry**

One syndrome element from S_1	Error location	Error vector e
$\longleftarrow b \longrightarrow$	$\leftarrow \lceil \log_2(k+1) \rceil \rightarrow$	$\longleftarrow b \longrightarrow$

a message is received the decoder calculates syndrome of the received message, and searches the corresponding value of calculated syndrome in LUT_2 which requires η_{TL} table look ups for the binary search such that $1 \leq \eta_{TL} \leq \lfloor \log_2 |S_1| \rfloor + 2$ (see [63]). In case of unavailability, the decoder declares a failure. Following steps are followed for decoding:

- For asymmetric CT-bursts of length l occurring within the check byte:

$$C_B = [\bar{C}_B + e] \pmod{2^b - 1}, \text{ where } e \in \epsilon_{b,l}.$$

- For asymmetric CT-bursts of length l occurring within j^{th} data byte:

$$B_j = [\bar{B}_j + e] \pmod{2^b - 1}, \text{ where } e \in \epsilon_{b,l}.$$

Example 2.3 describes integer $(CT_3B)_8$ code with the help of Table 2.1 generated using (2.1).

Table 2.1: $\boldsymbol{LUT_2}$ **for** $\boldsymbol{(40, 32)}$ **integer** $\boldsymbol{(CT_3B)_8}$ **code**

Sl. No.	**Syndrome (S_1)**	**Error Loc. (i)**	**Error (e)**	**Sl. No.**	**Syndrome (S_1)**	**Error Loc. (i)**	**Error (e)**
1	1	5	1	69	135	3	80
2	2	5	2	70	138	4	48
3	3	5	3	71	139	4	4
4	4	5	4	72	140	2	80
5	5	5	5	73	143	1	56
6	6	5	6	74	145	2	10
7	7	5	7	75	147	3	4
8	8	5	8	76	149	2	56
9	9	3	28	77	156	3	32
10	10	5	10	78	158	2	32
11	12	5	12	79	159	1	48
12	14	5	14	80	160	5	160
13	15	3	160	81	161	4	56
14	16	5	16	82	162	4	12
15	18	3	56	83	167	2	8
16	20	5	20	84	168	4	3
17	21	4	96	85	171	3	192
18	23	4	8	86	174	3	3
19	24	5	24	87	175	1	40
20	25	2	160	88	178	2	7
21	28	5	28	89	183	2	192
22	31	1	112	90	184	4	64
23	32	5	32	91	185	4	20
24	35	2	20	92	186	3	12

Contd...

Sl. No.	Syndrome (S_1)	Error Loc. (i)	Error (e)	Sl. No.	Syndrome (S_1)	Error Loc. (i)	Error (e)
25	36	3	112	93	189	2	6
26	39	3	8	94	190	1	160
27	40	5	40	95	191	1	32
28	42	4	192	96	192	5	192
29	43	2	112	97	195	3	40
30	46	4	16	98	197	4	2
31	48	5	48	99	199	1	28
32	52	4	7	100	200	2	5
33	56	5	56	101	201	3	2
34	57	3	64	102	202	2	28
35	61	2	64	103	205	4	160
36	62	1	224	104	207	1	24
37	63	1	96	105	208	4	28
38	64	5	64	106	211	2	4
39	66	3	7	107	213	3	96
40	67	4	112	108	215	1	20
41	69	4	24	109	219	2	96
42	70	2	40	110	220	4	10
43	72	3	224	111	222	2	3
44	78	3	16	112	223	1	16
45	79	2	16	113	224	5	224
46	80	5	80	114	225	3	20
47	81	4	6	115	226	4	1
48	86	2	224	116	227	1	14
49	92	4	32	117	228	3	1
50	93	3	6	118	230	4	80
51	95	1	80	119	231	1	12
52	96	5	96	120	233	2	2

Contd...

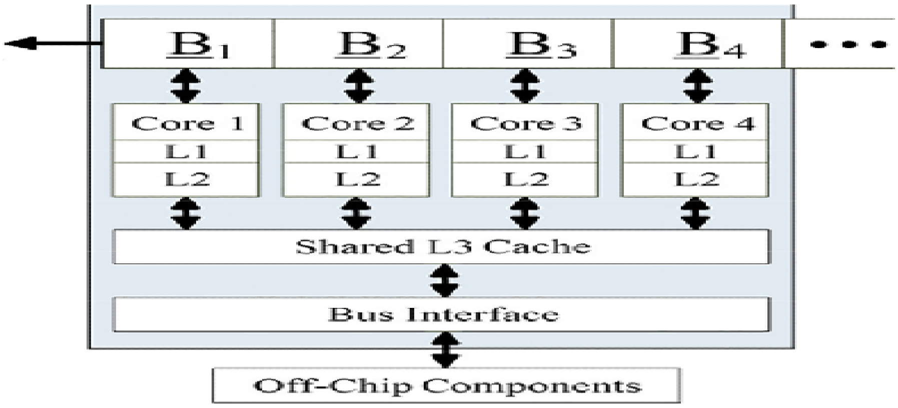
Sl. No.	**Syndrome (S_1)**	**Error Loc. (i)**	**Error (e)**	**Sl. No.**	**Syndrome (S_1)**	**Error Loc. (i)**	**Error (e)**
53	101	2	14	121	234	3	48
54	104	4	14	122	235	1	10
55	110	4	5	123	237	2	48
56	112	5	112	124	239	1	8
57	113	4	128	125	240	3	10
58	114	3	128	126	241	1	7
59	115	4	40	127	243	1	6
60	117	3	24	128	244	2	1
61	120	3	5	129	245	1	5
62	122	2	128	130	246	2	24
63	123	2	12	131	247	1	4
64	126	1	192	132	249	1	3
65	127	1	64	133	251	1	2
66	128	5	128	134	253	1	1
67	132	3	14	135	254	1	128
68	134	4	224				

Example 2.3. *Let* $b = 8$ *and* $l = 3$, *then* $C_1 = 2$, $C_2 = 11$, $C_3 = 27$ *and* $C_4 = 29$, *syndrome elements are listed in Table 2.1. Suppose a message* 11101010 00111100 10100101 11010100 *is transmitted, then check byte* C_B *will be* $C_B = 10000101$. *An asymmetric CT-burst of length 3 within an 8-bit byte may occur in the following ways:*

Case I *(Asymmetric CT-burst in a data byte): If the received message is* 11101010 00111100 10100101 10000100 10000101, *then syndrome* $S = [C_{\bar{B}} - \bar{C}_B] \pmod{2^b - 1} = [126 - 161] \pmod{255} = 220 = [-29 \times 10] \pmod{255}$. *Hence* B_4 *has an error* $e = 10 = 2^1 + 2^3$, *so the corrected data byte will be* $B_4 = [\bar{B}_4 + e] \pmod{255} = [33 + 10] \pmod{255} = 43 = 11010100$.

Case II *(Asymmetric CT-burst in the check byte): If the received message is*

Figure 2.2: **Diagram of a quad-core processor**



11101010 00111100 10100101 11010100 10000<u>000</u>, *then syndrome* $S = [C_{\bar{B}} - \bar{C}_B] \pmod{2^b - 1} = [161 - 1] \pmod{255} = 160$. *Hence the check byte has an error* $e = 160 = 2^5 + 2^7$, *so the corrected check byte will be* $C_B = [\bar{C}_B + e] \pmod{255} = [1 + 160] \pmod{255} = 161 = 10000101$.

2.3 Implementation

From the discussions done so far, it is clear that the encoder/decoder uses look up tables, LUT_1 and LUT_2, where LUT_1 contains the coefficients and LUT_2 the syndrome table. As these operations are supported by all processors, so it is discussed below how the proposed codes are implemented in an octa-core processor (Figure 2.2). The processing core has an integer unit and two private caches: $L1$ and $L2$. $L1$ is of very small size (up to 64 KB) and has very low access latency ($1 - 5$ clock cycles), whereas $L2$ has much larger size (up to 512 KB) but slower latency ($8 - 15$ clock cycles) [48]. As shown in Figure 2.2, $L3$ allows access to all eight cores of the processor. Also, it has the largest memory (up to 32 MB) and highest latency ($25 - 50$ clock cycles)[48].

Table 2.2: **First 32 possible coefficients for some integer $(CT_lB)_b$ codes**

b	l	**Coefficients**
8	3	2, 11, 27, 29

Contd...

b	l	**Coefficients**
8	4	Not possible
8	5	Not possible
16	3	2, 9, 11, 13, 17, 19, 23, 25, 29, 31, 37, 41, 43, 47, 49, 53, 59, 61, 67, 71, 73, 79, 81, 83, 89, 97, 99, 101, 103, 105, 107, 109
16	4	2, 17, 19, 21, 23, 25, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 81, 83, 89, 97, 101, 103, 107, 109, 113, 121, 127, 131, 149
16	5	2, 33, 35, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 97, 101, 107, 113, 117, 127, 137, 149, 157, 163, 179, 227, 233, 251, 271, 283, 289
32	6	2, 65, 67, 69, 71, 73, 77, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199
32	7	2, 129, 131, 133, 137, 139, 143, 145, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 199, 211, 223, 227, 229, 233, 239, 241, 251, 257, 263
32	8	2, 257, 259, 261, 263, 265, 269, 271, 277, 281, 283, 289, 293, 299, 307, 311, 313, 317, 331, 337, 341, 347, 349, 353, 359, 361, 367, 373, 379, 383, 389, 397

From Figure 2.1, it is clear that one syndrome element has $2b + \lceil \log_2(k+1) \rceil$ bits. So the size of LUT_2 will be $|S_1|\,(2b + \lceil \log_2(k+1) \rceil)$ bits. Theoretically after constructing LUT_2 by using (2.1) for the codes, decoder's job is to search the value $S \neq 0$ (syndrome) obtained for the received message with that of the syndromes available in LUT_2. For this, the decoder does a binary search by matching first b bits of S obtained with table entries from the set S_1. The task will be completed in η_{TL} ($1 \leq \eta_{TL} \leq \lfloor \log_2 |S_1| \rfloor + 2$) (refer [63]) table look ups if the elements from the syndrome sets are sorted in increasing order. Table 2.2 consists of some coefficients needed to construct integer $(CT_lB)_b$ codes. Using these coefficients, in Table 2.3, memory consumption is depicted for a few codes of this type.

Table 2.3: **Lookup table sizes for some integer $(CT_lB)_b$ codes**

Codes	b	l	LUT_1 size	LUT_2 size	Number of table look ups
(144,128)	16	4	4× 16B	4.5 KB	$1 \leq \eta_{TL} \leq 11$
(528,512)	16	5	4× 64B	32.48 KB	$1 \leq \eta_{TL} \leq 14$
(512,480)	32	6	4× 60B	0.12 MB	$1 \leq \eta_{TL} \leq 15$
(1024,992)	32	7	4× 124B	0.48 MB	$1 \leq \eta_{TL} \leq 17$
(1056,1024)	32	8	4× 128B	0.96 MB	$1 \leq \eta_{TL} \leq 18$

2.4 Comparison

Since no codes have been developed in this class capable of correcting the discussed errors, we compare the codes with similar error pattern correcting codes. Result 1.43 discusses the number of parity bits required for the code in [68] to correct asymmetric bursts. Thus, by matching the parameters with the proposed codes, we observe that upon the existence of the same number of information bits, codes in [68] have code rate $R_2 = \dfrac{kb}{(k+1)b+s} \leq \dfrac{kb}{(k+1)b+\log_2 k} < \dfrac{kb}{(k+1)b} = R_1$ (R_1: code rate of the proposed codes). That is, the proposed codes can correct similar types of errors with less redundancy.

In [30], CT-burst correcting linear codes with two sub-blocks are discussed. We consider b to be the length of both sub-blocks in [30]. Now taking information bits $= b$, redundancy $= b$ and burst length $= l$ in both sub-blocks, the number of error patterns with fixed length l is $2[(b-l+1)2^{l-1}]$, thus the number of bits in the syndrome table in [30] equals $[2b+b] \times 2[(b-l+1)2^{l-1}]$. Whereas in the proposed integer $(CT_lB)_b$ codes, bits required for constructing the syndrome table equals $[2b + \lceil \log_2(k+1) \rceil] \times 2[(b-l+1)2^{l-1}]$, which is clearly less than that of the linear codes. For instance, consider $b = 8,\ l = 3$, as per Result 1.44-1.45, we have the existence of linear $(16, 8)$ code capable of correcting CT-bursts of length 3 in both blocks. Also, by Table 2.2, we can construct $(16, 8)$ integer $(CT_3B)_8$ code, then we get the number of bits required for storing syndrome table as 1152 and 816 bits respectively for linear and integer codes.

Table 2.4 features the memory consumed and table look ups required by some integer codes capable of correcting different types of errors. It should be noted here that this is just a representation about which code to be used in terms of its cost effectiveness and possible error-correcting capability as the nature of the error is different in all of the cases.

Table 2.4: **Different integer codes with 32 information bytes**

Codes	$\boldsymbol{b}$	$\boldsymbol{l}$	$\boldsymbol{LUT_2}$ **size**	**No of table look ups**
$(CT_lB)_b$	32	8	0.96 MB	$1 \leq \eta_{TL} \leq 18$
From Result 1.19	32	8	3.84 MB	$1 \leq \eta_{TL} \leq 20$
From Result 1.24	32	NA	7.53 MB	$1 \leq \eta_{TL} \leq 21$
From Result 1.36	32	8	8.91 MB	$1 \leq \eta_{TL} \leq 21$

2.5 Probability and BER

In this section, we derive the expression for probability of erroneous decoding and BER, followed by a few graphs for the integer $(CT_lB)_b$ codes. Finally, a method for investigating undetected errors is discussed.

Theorem 2.4. *The probability of erroneous decoding $P_d(CT)$ for a $((k+1)b, kb)$ integer $(CT_lB)_b$ code is*

$$(k+1)\left[b\epsilon(1-\epsilon)^{(k+1)b-1} + \epsilon^2(1-\epsilon)^{(k+1)b-2}\left\{\left(b + \frac{1-\epsilon}{\epsilon}\right)\left(\frac{1-\epsilon}{\epsilon}\right)\left(\left(\frac{1}{1-\epsilon}\right)^{l-1} - 1\right)\right.\right.$$

$$-(l-1)\left(\frac{1}{1-\epsilon}\right)^{l-1}\left(\frac{1-\epsilon}{\epsilon}\right)\Bigg\}\Bigg],\quad \textit{where } \epsilon \textit{ is the crossover probability.}$$

Proof. A received codeword from a $((k+1)b, kb)$ integer $(CT_lB)_b$ code has $(k+1)$ b-bit blocks, thus a received erroneous message having l corrupted bits will have $(k+1)b-l$ non corrupted bits. By considering ϵ as the crossover probability of the Z-channel, the probability of erroneous decoding for a burst of length 1 occurring within a b-bit byte will be $b\epsilon(1-\epsilon)^{(k+1)b-1}$, since there are b number of asymmetric CT-bursts of length 1, thus the probability in this case will be $b\epsilon(1-\epsilon)^{(k+1)b-1}$. Similarly, the probability of erroneous decoding for asymmetric CT-bursts of length 2 will be $(b-1)\epsilon^2(1-\epsilon)^{(k+1)b-2}$, the probability of erroneous decoding for an asymmetric CT-burst of length 3 will be $(b-2)\Big\{\epsilon^2(1-\epsilon)^{(k+1)b-2}+\epsilon^3(1-\epsilon)^{(k+1)b-3}\Big\}$. Continuing this, the probability of erroneous decoding for asymmetric CT-bursts of length l occurring within a b-bit byte will be $(b-l+1)\sum_{i=0}^{l-2}\binom{l-2}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}$. Therefore, by summing up, we get the probability of erroneous decoding for asymmetric CT-bursts up to length l occurring within a b-bit byte as:

$$b\Big\{\epsilon(1-\epsilon)^{(k+1)b-1}\Big\}+(b-1)\Big\{\epsilon^2(1-\epsilon)^{(k+1)b-2}\Big\}+(b-2)\Big\{\epsilon^2(1-\epsilon)^{(k+1)b-2}$$
$$+\epsilon^3(1-\epsilon)^{(k+1)b-3}\Big\}+\ldots+(b-l+1)\Big\{\epsilon^2(1-\epsilon)^{(k+1)b-2}+\binom{l-2}{1}\epsilon^3(1-\epsilon)^{(k+1)b-3}$$
$$+\binom{l-2}{2}\epsilon^4(1-\epsilon)^{(k+1)b-4}+\ldots+\binom{l-2}{l-2}\epsilon^l(1-\epsilon)^{(k+1)b-l}\Big\}$$
$$=b\epsilon(1-\epsilon)^{(k+1)b-1}+(b-1)\sum_{i=0}^{0}\binom{0}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}$$
$$+(b-2)\sum_{i=0}^{1}\binom{1}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}+\ldots+(b-l+1)\sum_{i=0}^{l-2}\binom{l-2}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}$$
$$=b\epsilon(1-\epsilon)^{(k+1)b-1}+\sum_{j=1}^{l-1}\sum_{i=0}^{j-1}(b-j)\binom{j-1}{i}\epsilon^{i+2}(1-\epsilon)^{(k+1)b-i-2}$$
$$=b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}\Bigg[(b-j)\Bigg\{\left(\frac{\epsilon}{1-\epsilon}\right)^0+\binom{j-1}{1}\left(\frac{\epsilon}{1-\epsilon}\right)^1+\ldots$$
$$\ldots+\binom{j-1}{j-1}\left(\frac{\epsilon}{1-\epsilon}\right)^{j-1}\Bigg\}\Bigg]$$
$$=b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}(b-j)\left(1+\frac{\epsilon}{1-\epsilon}\right)^{j-1}$$

$$
\begin{aligned}
=&b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\sum_{j=1}^{l-1}(b-j)\left(\frac{1}{1-\epsilon}\right)^{j-1}\\
=&b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[b\sum_{j=1}^{l-1}\left(\frac{1}{1-\epsilon}\right)^{j-1}-\sum_{j=1}^{l-1}j\left(\frac{1}{1-\epsilon}\right)^{j-1}\right]\\
=&b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[b\left(\frac{\left(\frac{1}{1-\epsilon}\right)^{l-1}-1}{\left(\frac{1}{1-\epsilon}\right)-1}\right)\right.\\
&\left.-\left\{(l-1)\left(\frac{1}{1-\epsilon}\right)^{l-1}\left(\frac{1-\epsilon}{\epsilon}\right)-\left(\frac{1-\epsilon}{\epsilon}\right)^2\left\{\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right\}\right\}\right]\\
=&b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[\frac{b(1-\epsilon)}{\epsilon}\left\{\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right\}\right.\\
&\left.-(l-1)\left(\frac{1}{1-\epsilon}\right)^{l-1}\left(\frac{1-\epsilon}{\epsilon}\right)+\left(\frac{1-\epsilon}{\epsilon}\right)^2\left\{\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right\}\right]\\
=&b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left[\left(\frac{b(1-\epsilon)}{\epsilon}+\left(\frac{1-\epsilon}{\epsilon}\right)^2\right)\left\{\left(\frac{1}{1-\epsilon}\right)^{l-1}-1\right\}\right.\\
&\left.-(l-1)\left(\frac{1}{1-\epsilon}\right)^{l-1}\left(\frac{1-\epsilon}{\epsilon}\right)\right].
\end{aligned}
$$

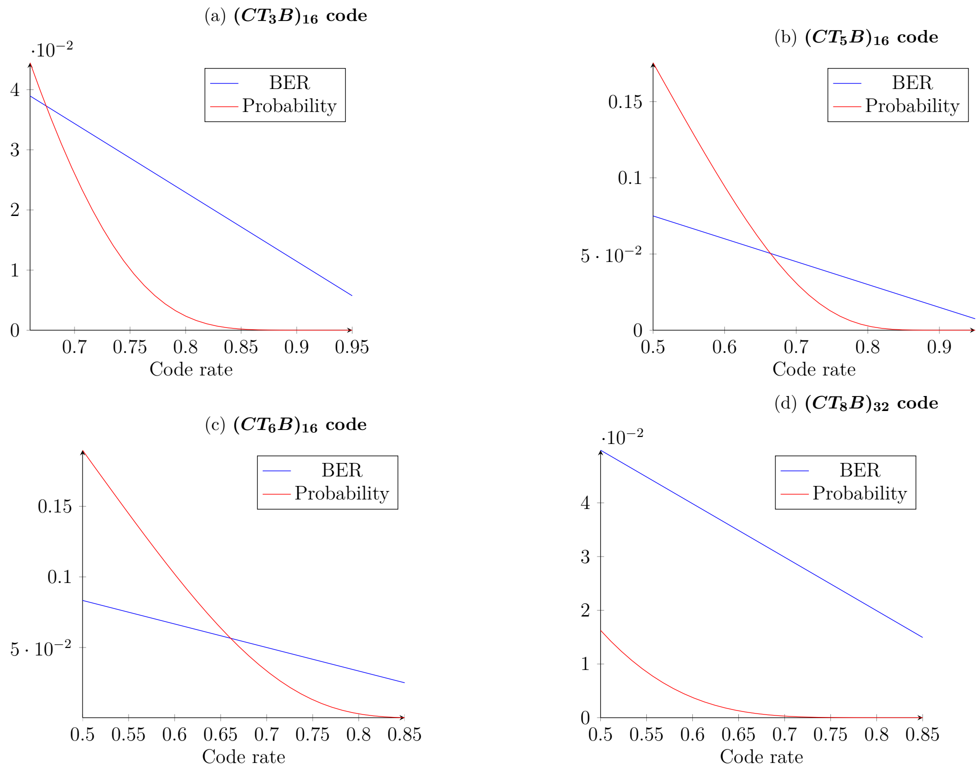
Since the code is capable of correcting one asymmetric CT-burst within a b-bit byte among $k+1$ b-bit bytes at a time, thus the probability of erroneous decoding of $((k+1)b, kb)$ integer $(CT_lB)_b$ codes will be

$$
(k+1)\left[b\epsilon(1-\epsilon)^{(k+1)b-1}+\epsilon^2(1-\epsilon)^{(k+1)b-2}\left\{\left(b+\tfrac{1-\epsilon}{\epsilon}\right)\left(\tfrac{1-\epsilon}{\epsilon}\right)\left(\left(\tfrac{1}{1-\epsilon}\right)^{l-1}-1\right)\right.\right.
$$
$$
\left.\left.-(l-1)\left(\tfrac{1}{1-\epsilon}\right)^{l-1}\left(\tfrac{1-\epsilon}{\epsilon}\right)\right\}\right]. \qquad \square
$$

Bit Error Rate (BER) is the ratio between the number of corrupted bits and the number of bits transmitted. In the proposed codes, the number of corrupted bits differ from 1 to l, so we have considered the average to determine the BER. The rate for length 1 will be $\frac{1}{(k+1)b}$, for length 2, the rate will be $\frac{2}{(k+1)b}$, continuing this, the rate for length l will be $\frac{\frac{2+3+\ldots+l}{l-1}}{(k+1)b}=\frac{2+3+\ldots+l}{(l-1)(k+1)b}$. Thus the BER for $(CT_lB)_b$ codes up to length l will be the average from 1 to l bits corrupted, hence

$$
\begin{aligned}
BER&=\frac{1}{(k+1)bl}\left[1+\sum_{j=2}^{l}\sum_{i=2}^{j}\frac{i}{j-1}\right]\\
&=\frac{1}{(k+1)bl}\left[1+\sum_{j=2}^{l}\frac{2+j}{2}\right]
\end{aligned}
$$

Figure 2.3: **Change in probability and BER for different code rates**



$$= \frac{1}{(k+1)bl}\left[\frac{l^2+5l-2}{4}\right].$$

By considering a few examples and $\epsilon = 0.1$, Figure 2.3 shows the change in probability and BER with respect to different code rates for the proposed codes.

An error is said to be **undetected** if the error is beyond specification and the resulting syndrome is equal to zero. With reference to the proposed class of codes, an asymmetric CT-burst e_r of length r will go undetected if $r > l$ and the resulting syndrome affected by this burst will be 0. For example, in the $(40, 32)$ integer $(CT_3B)_8$ code (refer Table 2.2), asymmetric CT-bursts of length 7, $10101010 = 85$ and $01010101 = 170$ will go undetected since the resulting syndrome $-27 \times 85 \pmod{255} = -27 \times 170 \pmod{255} = 0$. Since the discussed codes are constructed with the help of a computer search result by finding the coefficients C_i, where C_i's do not follow any particular algebraic pattern. So to determine the exact probability of undetected error becomes difficult for these classes of integer codes. Result below gives us the maximum possible ratio of an undetected asymmetric CT-burst with all possible bursts of the type having longer lengths.

Theorem 2.5. *The ratio between the number of undetected asymmetric CT-bursts and asymmetric CT-bursts of length between l and r in a $((k+1)b, kb)$ integer $(CT_lB)_b$ code is at most $\frac{2^r+\sum_{i=1}^{k}2^{p_i+r-b+1}}{(k+1)(2^{r-1}-2^{l-1})}$, here p_i is the highest power of 2 in the binary representation of the coefficient C_i.*

Proof. Let e_r be an asymmetric CT-burst of length $r > l$ in an integer $(CT_lB)_b$ code. For the error e_r to go undetected in the i^{th} b-bit byte, the resulting syndrome should be equal to zero, i.e. $C_i \times e_r \pmod{2^b-1} = 0 \implies 2^b - 1$ divides $C_i \times e_r$. As $e_r = \{2^x(1, 3, \ldots, 2^r - 1) | 0 \leq x \leq b - r\}$, so $2^b - 1$ divides $C_i \times 2^x e'_r$, where e'_r is an odd number between 1 and $2^r - 1$. Since $2^b - 1$ and 2^x are relatively prime, thus $2^b - 1$ divides $C_i \times e'_r$. Accordingly consider $C_i \times e'_r = (2^b - 1) \times M$ and let p_i to be the maximum possible power of 2 in the binary representation of C_i, therefore the maximum possible power of 2 in the binary representation of M will be $p_i + r - 1 - b + 1 = p_i + r - b$. Hence there are $2^{p_i+r-b+1}$ possible (maximum) choices for M. Since an asymmetric CT-burst having length up to l is always corrected, so we only consider asymmetric CT-bursts from length $l+1$ to r in the latter part of the ratio. Number of such bursts of length $l+1, l+2, \ldots, r$ are $2^{l-1}, 2^l, \ldots, 2^{r-2}$ respectively. Consequently, by considering lengths from $l + 1$ to r, we have the number of choices of a burst for a beginning position $= 2^{l-1} + 2^l + \ldots + 2^{r-2} = 2^{l-1}(2^{r-l} - 1)$. Hence the maximum possible ratio for an asymmetric CT-burst up to length r to go undetected within the i^{th} data byte will be $\frac{2^{p_i+r-b+1}}{2^{r-1}-2^{l-1}}$. Since the last data byte has coefficient value $C_{k+1} = -1$ by default, so $p_{k+1} = b - 1$. Therefore, by considering all $k + 1$ b-bit bytes, the required ratio will be $\frac{2^r+\sum_{i=1}^{k}2^{p_i+r-b+1}}{(k+1)(2^{r-1}-2^{l-1})}$. □

2.6 Conclusion

In this chapter, we have presented a class of integer codes capable of correcting asymmetric CT-bursts constructed with the help of computer search results. The probability of erroneous decoding over a Z-channel and the ratio for an error to go undetected are discussed. Similar encoding and decoding can be tried for CT-bursts occurring across two adjoining b-bit bytes, which can work without interleaving.