

This is the peer-reviewed version of the paper:

Radonjic, A., 2020. Integer Codes Correcting Double Errors and Triple-Adjacent Errors Within a Byte. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 28, 1901–1908.

<https://doi.org/10.1109/TVLSI.2020.2998364>



[This work is licensed under the Attribution-NonCommercial-NoDerivatives 4.0 International \(CC BY-NC-ND 4.0\)](https://creativecommons.org/licenses/by-nc-nd/4.0/)

Integer Codes Correcting Double Errors and Triple-Adjacent Errors within a Byte

Aleksandar Radonjic

Abstract—This paper presents a class of integer codes that are suitable for use in optical computer networks where the data is transmitted serially. The presented codes are constructed with the help of a computer and have three desirable properties. First, they use integer and lookup table operations, which makes them suitable for software implementation. Second, depending on the application requirements, the proposed codes can be used as lower rate error correction (EC) codes or as high-rate error detection (ED) codes. In the EC mode, which is suited for real-time applications, the receiver can correct all single and double errors as well as all triple-adjacent (TA) errors within one b -bit byte. On the other hand, if the integrity of data is of high importance, the receiver may operate in the ED mode. In that case, it is able to detect all quadruple errors, all double TA errors within one b -bit byte and all double TA errors within two b -bit bytes. Finally, it is important to note that the presented codes can be interleaved without delay and without using any additional hardware. Owing to this, it is possible to construct simple codes capable of detecting/correcting multiple TA and random errors.

Index Terms—Integer codes, optical computer networks, error control, single errors, double errors, triple-adjacent errors.

I. INTRODUCTION

Optical computer networks (OCNs) are usually classified in terms of the area they cover. Thus, for example, optical LANs (local area networks) connect computers that are physically close together, optical MANs (metropolitan area networks) are used to connect optical LANs, while optical WANs (wide area networks) are designed to connect computers on a national or continental scale (Fig. 1a) [1]. Due to these differences, these networks operate at different speeds: optical LANs and MANs typically run at 1 Gbps or 10 Gbps, while optical WANs rarely have data rates above 1 Gbps (these networks are mostly built using leased lines) [1], [2].

Although these facts are well documented in the literature, it is very rarely mentioned that OCNs share several common features. The first of them concerns the high computing power of the network nodes (NNs). Namely, from [3] it is known that exterior nodes (computers and servers) are always equipped with general purpose processors (GPPs), while interior nodes (routers and switches) contain network processors (NPs). The only difference between these chips is that NPs integrate some additional coprocessors [3] (e.g. for cryptographic operations). Hence, it can be said that each NN has hardware to perform integer, floating-point, logical and memory-based operations (e.g. lookup table (LUT) operations).

The second common characteristic of almost all OCNs is that they use Ethernet packets for internode communication. In optical LANs/MANs these packets are used directly, while in optical WANs they are encapsulated into different frames (e.g. GFP frame [2]). Each Ethernet packet, as it is known, contains between 368 and 12000 bits of data, and ends with a 32-bit CRC code. Owing to this, the receiver is able to detect up to three random errors and all bursts of length up to 32 bits.

The third and last common feature relates to the behaviour of channel errors. In many studies, such as [4]-[8], it has been shown that single errors are 10 to 1000 times more frequent than any other type of errors. Besides this, in [4]-[8] it was shown that the burst error probability decreases exponentially with increasing burst length (Fig. 1b). If we add to this the fact that some OCNs may have high error rates (between 10^{-4} and 10^{-3}) [7] we easily come to the conclusion that the most likely errors may occur twice. Therefore, it is reasonable to assume that, in OCNs, the vast majority of corrupted packets will contain either single, double or triple-adjacent (TA) errors.

Bearing this in mind, one may conclude that CRC codes are not the optimal solution for OCNs. First of all, these codes are based on finite field (FF) arithmetic, which is not supported by GPPs. For this reason, all NNs are equipped with hardware that performs FF operations. The second disadvantage is that CRCs cannot efficiently adapt to the application requirements. Namely, they are designed to detect the errors, but not to correct them. This means that each erroneous packet will be recovered through retransmission, which may be unacceptable for real-time (RT) applications.

Motivated by these facts, in this paper, we extend our previous work [9]-[14], and present a class of integer codes suitable for use in OCNs where the data is transmitted serially. Compared to CRCs, our codes have three advantages. First, they use integer and LUT operations, which makes them suitable for implementation in software. Second, depending on the application requirements, the presented codes can be used as high-rate error detection (ED) codes or as lower rate error correction (EC) codes. In the EC mode, which is suited for RT applications, the receiver can correct all single and double errors, as well as all TA errors confined to one b -bit byte (TA₁ errors). On the other hand, if the integrity of data is of high importance, the receiver may operate in the ED mode. In that case, it is able to detect all quadruple errors, all double TA errors corrupting one b -bit byte (DTA₁ errors) and all double TA errors corrupting two b -bit bytes (DTA₂ errors). Finally, the codes presented in this paper can be interleaved without delay and without using any additional hardware. Owing to this, it is possible to construct very simple codes capable of correcting/detecting multiple TA and random errors.

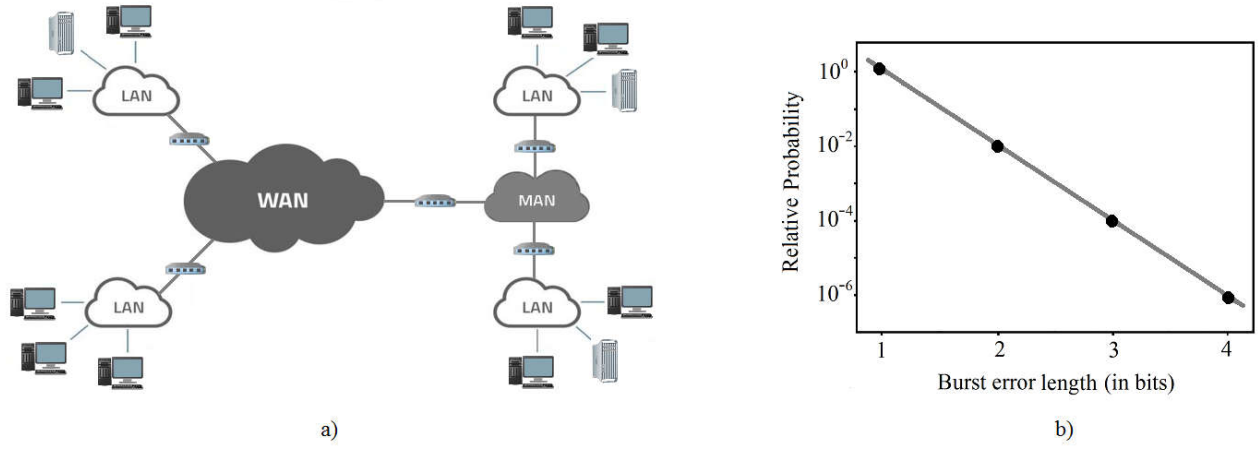


Fig. 1. Optical computer networks: (a) the topology and (b) the typical error behavior.

The organization of this paper is as follows: Section 2 deals with the construction of integer double error correcting and TA_1 error correcting (DEC-(TAEC) $_b$) codes. The error control procedure for these codes is described in Section 3. Section 4 is devoted to the application of the proposed codes in OCNs, while Section 5 compares these codes with linear DEC-TAEC codes. Finally, Section 6 concludes the paper.

II. CODES CONSTRUCTION

Definition 1. [14] Let $Z_{2^b-1} = \{0, 1, \dots, 2^b - 2\}$ be the ring of integers modulo $2^b - 1$ and let $B_i = \sum_{n=0}^{b-1} a_n \cdot 2^n$ be the integer representation of a b -bit byte, where $a_n \in \{0, 1\}$ and $1 \leq i \leq k$. Then, the code $C(b, k, c)$, defined as

$$C(b, k, c) = \left\{ x \in Z_{2^b-1}^{k+1} \cdot \sum_{i=1}^k C_i \cdot B_i \equiv B_{k+1} \pmod{2^b-1} \right\} \quad (1)$$

is an $(kb + b, kb)$ integer code, where $x = (B_1, B_2, \dots, B_k, B_{k+1}) \in Z_{2^b-1}^{k+1}$ is the codeword vector, $c = (C_1, C_2, \dots, C_k, 1) \in Z_{2^b-1}^{k+1}$ is the coefficient vector and $B_{k+1} \in Z_{2^b-1}$ is an integer.

Definition 2. Let $x = (B_1, B_2, \dots, B_k, B_{k+1}) \in Z_{2^b-1}^{k+1}$, $y = (\underline{B}_1, \underline{B}_2, \dots, \underline{B}_k, \underline{B}_{k+1}) \in Z_{2^b-1}^{k+1}$ and $e = (\underline{B}_1 - B_1, \underline{B}_2 - B_2, \dots, \underline{B}_k - B_k, \underline{B}_{k+1} - B_{k+1}) = (e_1, e_2, \dots, e_k, e_{k+1}) \in Z_{2^b-1}^{k+1}$ be respectively, the sent codeword, the received codeword and the error vector. Then, the syndrome S of the received codeword is defined as

$$S = \sum_{i=1}^k C_i \cdot \underline{B}_i - \underline{B}_{k+1} \pmod{2^b-1} = \sum_{i=1}^k e_i \cdot C_i \pmod{2^b-1} \quad (2)$$

From (1) and (2) it can be noticed that the byte with all 1s has the same integer value as the byte with all 0s (since $2^b - 1 \equiv 0 \pmod{2^b-1}$). This means that we cannot construct codes that correct byte errors. However, we can construct codes with DEC-(TAEC) $_b$ capability. The first step in that direction is to define sets of syndromes that correspond to single, double and TA_1 errors.

Definition 3. The set of syndromes corresponding to single errors is defined as

$$s_1 = \bigcup_{i=1}^{k+1} \{ \pm 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-1 \} \quad (3)$$

Definition 4. The set of syndromes corresponding to double errors is defined as

$$s_2 = d_1 \cup d_2 \quad (4)$$

where

$$d_1 = \bigcup_{i=1}^{k+1} \{ (\pm 2^r \pm 2^s) \cdot C_i \pmod{2^b-1} : 0 \leq r < s \leq b-1 \} \quad (5)$$

$$d_2 = \bigcup_{i=1}^k \bigcup_{j=i+1}^{k+1} \{ \pm 2^r \cdot C_i \pm 2^s \cdot C_j \pmod{2^b-1} : 0 \leq r, s \leq b-1 \} \quad (6)$$

Definition 5. The set of syndromes corresponding to TA_1 errors is defined as

$$s_3 = \bigcup_{i=1}^{k+1} \{ (\pm 2^{2^i} \pm 2^1 \pm 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \} \quad (7)$$

If we compare the sets s_1 , s_3 and d_1 , we will observe that their intersection is not empty. Moreover, it can be proved that the sets s_1 and s_3 are subsets of d_1 .

Theorem 1. For any $b \geq 3$ it holds $s_1 \subseteq d_1$ and $s_3 \subseteq d_1$.

Proof. To prove that $s_1 \subseteq d_1$, observe that the set d_1 can be express as the union

$$d_1 = \bigcup_{i=1}^{2^b-2} P_i$$

where

$$P_1 = \bigcup_{i=1}^{k+1} \{ \pm (2^1 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-2 \}$$

$$P_2 = \bigcup_{i=1}^{k+1} \{ \pm (2^1 + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-2 \}$$

$$P_3 = \bigcup_{i=1}^{k+1} \{ \pm (2^2 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \}$$

$$P_4 = \bigcup_{i=1}^{k+1} \{ \pm (2^2 + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \}$$

$$P_5 = \bigcup_{i=1}^{k+1} \{ \pm (2^3 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-4 \}$$

$$P_6 = \bigcup_{i=1}^{k+1} \{ \pm (2^3 + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-4 \}$$

\vdots

$$P_{2^b-7} = \bigcup_{i=1}^{k+1} \{ \pm (2^{b-3} - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq 2 \}$$

$$P_{2^b-6} = \bigcup_{i=1}^{k+1} \{ \pm (2^{b-3} + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq 2 \}$$

$$P_{2b-5} = \bigcup_{i=1}^{k+1} \left\{ \pm (2^{b-2} - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq 1 \right\}$$

$$P_{2b-4} = \bigcup_{i=1}^{k+1} \left\{ \pm (2^{b-2} + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq 1 \right\}$$

$$P_{2b-3} = \bigcup_{i=1}^{k+1} \left\{ \pm (2^{b-1} - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : r = 0 \right\}$$

$$P_{2b-2} = \bigcup_{i=1}^{k+1} \left\{ \pm (2^{b-1} + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : r = 0 \right\}$$

From the above it is easy to show that

$$\begin{aligned} P_{2b-3} &= \bigcup_{i=1}^{k+1} \left\{ \pm (2^{b-1} - 1) \cdot C_i \pmod{2^b-1} \right\} \equiv \\ &\equiv \bigcup_{i=1}^{k+1} \left\{ \mp 2^{b-1} \cdot C_i \pmod{2^b-1} \right\}, \end{aligned}$$

$$s_1 \equiv P_1 \cup P_{2b-3}.$$

Hence, it is clear that $s_1 \subseteq d_1$. To prove the second part of the theorem, note that the set s_3 can be express as the union

$$s_3 = \bigcup_{j=1}^8 Q_j$$

where

$$Q_1 = \bigcup_{i=1}^{k+1} \left\{ (-2^2 - 2^1 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

$$Q_2 = \bigcup_{i=1}^{k+1} \left\{ (-2^2 - 2^1 + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

$$Q_3 = \bigcup_{i=1}^{k+1} \left\{ (-2^2 + 2^1 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

$$Q_4 = \bigcup_{i=1}^{k+1} \left\{ (-2^2 + 2^1 + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

$$Q_5 = \bigcup_{i=1}^{k+1} \left\{ (2^2 - 2^1 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

$$Q_6 = \bigcup_{i=1}^{k+1} \left\{ (2^2 - 2^1 + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

$$Q_7 = \bigcup_{i=1}^{k+1} \left\{ (2^2 + 2^1 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

$$Q_8 = \bigcup_{i=1}^{k+1} \left\{ (2^2 + 2^1 + 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\}$$

By comparing the sets $Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, P_1, P_3$ and P_4 we conclude that

$$(Q_2 \cup Q_7) \equiv P_4$$

$$(Q_3 \cup Q_6) \equiv P_3$$

$$(Q_4 \cup Q_5) \subseteq P_1.$$

Also, if we compare the sets Q_1, Q_8, P_5 and P_{2b-7} we see that

$$\begin{aligned} (Q_1 \cup Q_8) &= \left\{ \pm (2^3 - 2^0) \cdot 2^r \cdot C_i \pmod{2^b-1} : 0 \leq r \leq b-3 \right\} \equiv \\ &\equiv \left(\bigcup_{i=1}^{k+1} \left\{ \left[\pm (2^3 - 2^0) \cdot 2^{b-3} \cdot C_i \right] \pmod{2^b-1} \right\} \cup P_5 \right) = \\ &= \left(\bigcup_{i=1}^{k+1} \left\{ \left[\mp (2^{b-3} - 2^0) \cdot C_i \right] \pmod{2^b-1} \right\} \cup P_5 \right) \\ &\subseteq (P_{2b-7} \cup P_5). \end{aligned}$$

As a result, it follows that $s_3 \subseteq d_1$. \square

Now, when we know that $s_1 \subseteq d_1$ and $s_3 \subseteq d_1$, we can prove the following theorem.

Theorem 2. The codes defined by (1) can correct all single, double and TA_1 errors iff there exists k mutually different coefficients C_i such that

$$1. |d_1| = \lceil 2 \cdot (b-1)^2 - 2 \rceil \cdot (k+1),$$

$$2. |d_2| = 2 \cdot b^2 \cdot k \cdot (k+1),$$

$$3. d_1 \cap d_2 = \emptyset,$$

where $|A|$ is the cardinality of A .

Proof. Condition 1 of this theorem says that double errors confined to a b -bit byte generate $\lceil 2 \cdot (b-1)^2 - 2 \rceil \cdot (k+1)$ nonzero syndromes. To prove this, let us first observe that $P_3 \subseteq P_2$ and $P_{2b-5} \subseteq (P_{2b-2} \cup P_2)$. Now, suppose that the coefficients C_i have values such that

$$|P_{2j-1}| = |P_{2j}| = 2 \cdot (b-j) \cdot (k+1), \quad j = 1, 2, \dots, b-1,$$

$$P_1 \cap P_2 \cap P_4 \cap P_5 \cap \dots \cap P_{2b-6} \cap P_{2b-4} \cap P_{2b-3} \cap P_{2b-2} = \emptyset.$$

In that case, it is easy to show that

$$|d_1| = \sum_{j=1}^{2b-2} |P_j| - |P_3| - |P_{2b-5}| = \lceil 2 \cdot (b-1)^2 - 2 \rceil \cdot (k+1).$$

Similarly, Condition 2 implies that double errors corrupting two b -bit bytes generate $2 \cdot b^2 \cdot k \cdot (k+1)$ nonzero syndromes. To prove this, observe that the set d_2 can be expressed as

$$d_2 = \bigcup_{i=1}^k R_i$$

where

$$R_1 = \bigcup_{j=2}^{k+1} \left\{ \pm 2^r \cdot C_1 \pm 2^s \cdot C_j \pmod{2^b-1} : 0 \leq r, s \leq b-1 \right\}$$

$$R_2 = \bigcup_{j=3}^{k+1} \left\{ \pm 2^r \cdot C_2 \pm 2^s \cdot C_j \pmod{2^b-1} : 0 \leq r, s \leq b-1 \right\}$$

$$R_3 = \bigcup_{j=4}^{k+1} \left\{ \pm 2^r \cdot C_3 \pm 2^s \cdot C_j \pmod{2^b-1} : 0 \leq r, s \leq b-1 \right\}$$

\vdots

$$R_k = \left\{ \pm 2^r \cdot C_k \pm 2^s \pmod{2^b-1} : 0 \leq r, s \leq b-1 \right\}$$

Obviously, if the coefficients C_i have values such that

$$R_1 \cap R_2 \cap \dots \cap R_k = \emptyset,$$

$$|R_1| = 4 \cdot b^2 \cdot k,$$

$$|R_2| = 4 \cdot b^2 \cdot (k-1),$$

$$|R_3| = 4 \cdot b^2 \cdot (k-2),$$

\vdots

$$|R_k| = 4 \cdot b^2,$$

then

$$|d_2| = \sum_{i=1}^k |R_i| = 4 \cdot b^2 \cdot \sum_{i=1}^k (k+1-i) = 2 \cdot b^2 \cdot k \cdot (k+1).$$

Finally, Condition 3 is a necessary condition for distinguishing double errors confined to one b -bit byte from those corrupting two b -bit bytes. So, $(kb + b, kb)$ integer DEC-(TAEC) $_b$ codes must satisfy all the conditions 1 to 3. Conversely, if the codes satisfy conditions 1 to 3, then we can distinguish double errors within one b -bit byte from those corrupting two b -bit bytes.

Algorithm 1 The pseudocode for finding the coefficients C_i

Input: $b, \xi = \{0\}, c = \{1\}$;
Output: $c = \{1, C_1, C_2, \dots, C_k\}$;
for $r = 0$ **to** $b - 2$
 for $s = r + 1$ **to** $b - 1$
 $\xi = \{\pm 2^r \pm 2^s \pmod{2^b - 1}\}$;
 end
end
 $i = 1$;
for $m = 2$ **to** $2^b - 2$
 $l_1 = \{0\}$;
 for $r = 0$ **to** $b - 2$
 for $s = r + 1$ **to** $b - 1$
 $l_1 = \{(\pm 2^r \pm 2^s) \cdot m \pmod{2^b - 1}\}$;
 end
 end
 if $(|l_1| = 2 \cdot (b - 1)^2 - 2)$ **and** $(l_1 \cap \xi = \emptyset)$
 $l_2 = \{0\}$;
 for $k = 1$ **to** $|c|$
 for $r = 0$ **to** $b - 1$
 for $s = 0$ **to** $b - 1$
 $l_2 = \{(\pm 2^r \cdot c(k) \pm 2^s \cdot m) \pmod{2^b - 1}\}$;
 end
 end
 end
 if $(|l_2| = 4 \cdot |c| \cdot b^2)$ **and** $(l_2 \cap \xi = \emptyset)$ **and** $(l_2 \cap l_1 = \emptyset)$
 $\xi \leftarrow \xi \cup l_1 \cup l_2$;
 $C_i = m$;
 $c \leftarrow c \cup C_i$;
 $i \leftarrow i + 1$;
 end
 end
 end
end

We can also correct all single, double and TA_1 errors. Hence, these codes are $(kb + b, kb)$ integer DEC-(TAEC) $_b$ codes. \square

As a corollary of Theorem 2, we can state the following.

Theorem 3. The error set for $(kb + b, kb)$ integer DEC-(TAEC) $_b$ codes has

$$|\xi| = |s_2| = |d_1| + |d_2| = 2 \cdot [b \cdot (k + 1) - 1]^2 - 2$$

nonzero elements.

Proof. The proof follows from Theorem 2. \square

Theorem 4. For any $(kb + b, kb)$ integer DEC-(TAEC) $_b$ code it holds that

$$k \leq \left\lfloor \frac{2^{\frac{b-1}{2}} - b + 1}{b} \right\rfloor.$$

Proof. From Theorem 3 it is known that the error set ξ has $2 \cdot [b \cdot (k + 1) - 1]^2 - 2$ nonzero elements. On the other hand, Definition 1 states that the total number of nonzero syndromes is equal to $2^b - 2$. Obviously, if we combine Definition 1 with Theorem 3 we get the inequality

$$2 \cdot [b \cdot (k + 1) - 1]^2 - 2 \leq 2^b - 2$$

where from it follows that

$$k \leq \left\lfloor \frac{2^{\frac{b-1}{2}} - b + 1}{b} \right\rfloor. \quad \square$$

In Section I it was mentioned that the presented codes can be also used as high-rate ED codes. However, before showing this, we need to prove the following theorem.

TABLE I
NUMBER OF COEFFICIENTS FOR SOME INTEGER DEC-(TAEC) $_b$ CODES.

	$b = 9$	$b = 10$	$b = 11$	$b = 12$	$b = 13$	$b = 14$	$b = 15$	$b = 16$
Theory	0	1	2	2	4	5	7	10
Experiment	0	0	1	1	1	2	3	3

TABLE II
FIRST 96 COEFFICIENTS FOR SOME INTEGER DEC-(TAEC) $_b$ CODES.

$b = 16$							
53	231	1067					
$b = 24$							
45	201	477	1109	1319	3129	3453	4847
9581	10117	11837	15411	17897	28827	44061	52265
74329	119841	174283	302403	674075	830035		
$b = 32$							
45	201	477	1109	1319	3129	3453	4847
9581	10117	11837	15411	17897	18439	23781	29749
34757	36419	44865	46009	51889	68223	81619	93047
108053	112279	113181	117189	164183	167119	169211	196783
201311	209395	256657	264427	275489	282429	310935	354225
386703	428269	432035	446911	514953	599285	634607	690403
748103	774457	834335	892067	893141	1013237	1067671	1087365
1103047	1122449	1191261	1248189	1297563	1342681	1458509	1570785
1701685	1789337	1904439	2007495	2143923	2240111	2300025	2332779
2346995	2583965	2597467	3176613	3200875	3218123	3333741	3677993
3759663	4008735	4245743	4301929	4539051	4637371	5001505	5168905
5659385	6365959	7104157	7276563	8858289	8994505	9282467	9856101

Theorem 5. Any $(kb + b, kb)$ integer DEC-(TAEC) $_b$ code, when operating as the ED code, is able to detect all quadruple errors, all DTA_1 errors and all DTA_2 errors.

Proof. From coding theory, it is known that any DEC code can detect all quadruple errors. On the other hand, from the proof of Theorem 1 we know that some double errors, within a b -bit byte, produce the same syndromes as TA_1 errors. From this we easily conclude that DTA_2 errors generate the same syndromes as some quadruple errors corrupting two b -bit bytes. In the same way, it can be concluded that DTA_1 errors produce the same syndromes as some quadruple errors corrupting one b -bit byte. Therefore, any $(kb + b, kb)$ integer DEC-(TAEC) $_b$ code, when operating in the ED mode, is able to detect all quadruple errors, all DTA_1 errors and all DTA_2 errors. \square

To illustrate the applicability of the above theory, we have conducted an exhaustive computer search. Our first goal was to find out how the number of the C_i 's depends on the byte length (Table I), while the second goal was to find the C_i 's that allow us to construct some practical codes (Table II). The pseudocode of our search method is presented in Algorithm 1.

III. ERROR CONTROL PROCEDURE

The error control procedure for the proposed codes is very similar to those described in [9]-[14]. In short, if it operates in the ED mode, the decoder has only one task: to check whether the received data are correct or not. As a result, the incoming packet will be either dropped ($S \neq 0$) or accepted ($S = 0$). However, if it operates in the EC mode, the decoder will try to recover the original data. The first step in this direction is to obtain the EC data from the syndrome table (ST). After that, in the next step, the decoder will execute one of the following operation(s):

TABLE III
THE SYNDROME TABLE FOR THE (22, 11) INTEGER DEC-(TAEC)₁₁ CODE.

	Element of ζ	i	E_1	j	E_2		Element of ζ	i	E_1	j	E_2		Element of ζ	i	E_1	j	E_2
1	1	2	1	0	0	291	640	2	1407	0	0	585	1395	1	2016	0	0
2	2	2	2	0	0	292	641	1	1032	0	0	586	1396	1	512	2	1919
3	3	2	3	0	0	293	642	1	1919	2	1023	587	1404	1	128	2	1023
4	4	2	4	0	0	294	643	1	1919	2	1024	588	1405	1	128	2	1024
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
287	630	1	2033	0	0	581	1383	1	2039	2	1023	877	2043	2	4	0	0
288	634	1	256	2	1919	582	1384	1	2039	2	1024	878	2044	2	3	0	0
289	637	1	128	2	256	583	1391	1	16	2	64	879	2045	2	2	0	0
290	639	1	32	2	32	584	1394	1	60	0	0	880	2046	2	1	0	0

• for errors corrupting one b -bit byte
 $B_i = \underline{B}_i + E_1 \pmod{2^b - 1}, 1 \leq i \leq k+1;$ (8)
 $E_1 \in \{\mp 2^r \mp 2^s \pmod{2^b - 1} : 0 \leq r < s \leq b-1\}$

• for errors corrupting two b -bit bytes
 $B_i = \underline{B}_i + E_1 \pmod{2^b - 1}, 1 \leq i \leq k;$ (9)
 $B_j = \underline{B}_j + E_2 \pmod{2^b - 1}, i < j \leq k+1;$ (10)
 $E_1, E_2 \in \{\mp 2^r \pmod{2^b - 1} : 0 \leq r \leq b-1\}$

To generate the ST it is necessary to substitute the values of b and C_i into (5)-(6). In this way, exactly $|\zeta|$ (Theorem 3) relationships between the nonzero syndrome (element of ζ), error location(s) (i, j) and error vector(s) (E_1, E_2) will be established (Fig. 2). So, when $S \neq 0$, the decoder's task will be to find the entry with the first b bits as that of the syndrome S . If the ST is sorted, this task will be finished after n_{TL} table lookups and n_{TL} comparisons ($1 \leq n_{TL} \leq \lfloor \log_2 |\zeta| \rfloor + 2$) [9].

To make this procedure more clear, let us consider it on example of the (22, 11) DEC-(TAEC)₁₁ code. (Remark: the syndrome table for this code has $|\zeta| = 880$ entries, and thus, it will be shown partially.)

Example 1. Let $b = 11, k = 1$ and $C_1 = 45$. Now, suppose that we want to transmit 9 bits of data, $B_1 = 10010011101_2 = 1181$. In that case, the integer value of the second (last) byte will be equal to

$$B_{k+1} = B_2 = 45 \cdot 1181 \pmod{2047} = 1970 = 11110110010_2$$

and the codeword will have the form, $x = (B_1, B_2) = (1181, 1970) = (10010011101_2, 11110110010_2)$.

Scenario 1: Suppose that the 6th and 11th bit are received in error, $y = (\underline{B}_1, \underline{B}_2) = (10010\underline{1}11100_2, 11110110010_2) = (1212, 1970)$. As explained previously, after calculating the value of the syndrome S

$$S = C_1 \cdot \underline{B}_1 - \underline{B}_2 \pmod{2047} = 1212 \cdot 45 - 1970 \pmod{2047} = 1395$$

the decoder will lookup the ST to find the entry with the first 11 bits as that of the syndrome S (Table III). After that, in the next step, it will execute the operation

$$B_1 = 1212 + 2016 \pmod{2047} = 1181.$$

Scenario 2: Assume that the 4th and 12th bit are received in error, $y = (\underline{B}_1, \underline{B}_2) = (10000011101_2, \underline{0}1110110010_2) = (1053, 946)$. As in the previous case, the decoder will calculate

$$S = C_1 \cdot \underline{B}_1 - \underline{B}_2 \pmod{2047} = 1053 \cdot 45 - 946 \pmod{2047} = 1405$$

and conclude that the value $S = 1405$ indicates an error within the first and second byte (Table III). As a result, the following procedure will take place

$$B_1 = 1053 + 128 \pmod{2047} = 1180,$$

$$B_2 = 946 + 1024 \pmod{2047} = 1970.$$

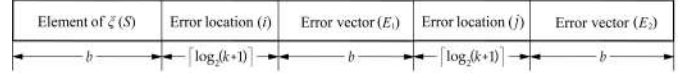


Fig. 2. Bit-width of one syndrome table entry.

IV. APPLICATION IN MODERN OCNs

If we analyze the procedures described above, we will note that the decoder uses very simple operations: integer additions, integer multiplications and table lookups. As these operations are supported by all processors, it is interesting to analyze how the proposed codes can be used in modern OCNs. Without loss of generality, we will suppose that all nodes (computers, routers, etc.) are equipped with six-core processors having the same specifications as in [14] (Fig. 3a). Likewise, we will assume that each node has two modes of operation: the ED mode, in which the data are protected by the six-byte interleaved (3104, 3072) code, and the EC mode, in which the data are protected with the six-byte interleaved (704, 672) code. In that case, regardless of the mode of operation, each node will compute the values of six syndromes [14]:

• Core 1

$$S_1 = \sum_{i=1}^k C_i \cdot \underline{B}_{6(i-1)+1} - \underline{B}_{6k+1} \pmod{2^{32}-1} \quad (11)$$

• Core 2

$$S_2 = \sum_{i=1}^k C_i \cdot \underline{B}_{6(i-1)+2} - \underline{B}_{6k+2} \pmod{2^{32}-1} \quad (12)$$

⋮

• Core 6

$$S_6 = \sum_{i=1}^k C_i \cdot \underline{B}_{6(i-1)+6} - \underline{B}_{6k+6} \pmod{2^{32}-1} \quad (13)$$

The only difference is that, in the EC mode ($k = 21$), the NNs perform much less operations than in the ED mode ($k = 96$). However, when viewed from the throughput standpoint, this is not a disadvantage. Namely, from [14] we know that, in the case of operating in the EC mode, each node requires one second to decode

$$G_{EC} = \frac{(3.3 \cdot 10^9) \cdot 192 \cdot k}{9.5 \cdot k + 35 \cdot n_{TL} + 3} \quad (14)$$

data bits. If we apply this result to the presented theory, we can easily conclude that, for $k = 21$, each node achieves a throughput of 14.19 Gbps (Fig. 3b). On the other hand, if they operate in the ED mode, the NNs will not lookup the STs or perform any EC operation. As a result, the expression (14) reduces to

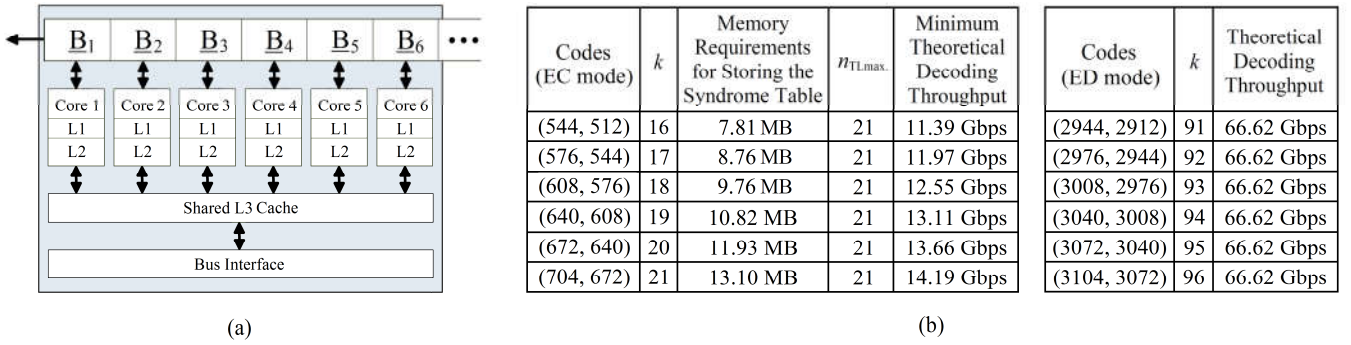


Fig. 3. (a) Block diagram of six-core processor and (b) theoretical decoding throughputs for some six-byte interleaved integer DEC-(TAEC)₃₂ codes.

$$G_{ED} = \frac{(3.3 \cdot 10^9) \cdot 192 \cdot k}{9.5 \cdot k + 1} \quad (15)$$

where from it is easy to calculate that, for $k = 96$, each node achieves a throughput of 66.62 Gbps. Having this in mind, we can draw some general conclusions regarding the application of the proposed codes:

1. All considered codes have the potential to be used in 10G or slower OCNs (Fig. 3b). In the EC mode, they are able to protect between 3072 and 4032 bits. On the other hand, if operating in the ED mode, they can protect up to 18432 bits.

2. All considered codes are six times interleaved at the byte level. This means that the decoder can correct (detect) various types of errors, including all (double) TA errors.

3. All considered codes share the same coefficients ($C_1 = 45$, $C_2 = 201$, ..., $C_{96} = 9856101$) (Table II). As a result, and depending on the application requirements, the NNs can easily switch from one mode to another. In the analyzed case, for example, the NNs only need to know whether they will use 21 (the EC mode) or 96 coefficients (the ED mode). On the basis of this information they will also know whether they need to lookup the STs or not.

V. COMPARISON WITH LINEAR DEC-TAEC CODES

In the coding literature, one can find only two codes with DEC-TAEC capability [15], [16]. The first was obtained by modifying the DEC-BCH codes and the second by modifying the DEC-OLS codes. In both cases, the modifications were done at the cost of increased hardware complexity, but without adding additional check bits.

As for the DEC-TAEC BCH codes, they have parameters $(2^u - 1, 2^u - 1 - 2 \cdot u)$, where $u \geq 4$. This means that the codes of highest-rate are (15, 7), (31, 21) and so on. On the other hand, it is known that the DEC-OLS codes are very inefficient in terms of redundancy. The parameters of these codes are $(m^2 + 4 \cdot m, m^2)$, where $m \geq 1$, which means that they require many more check bits than the DEC-TAEC BCH codes. Unlike the linear DEC-TAEC codes, the proposed ones are characterized by the parameters $(kb + b, kb)$. Although there is a significant difference between experimental and theoretical results (Table I), it can be said that the proposed codes are relatively efficient in terms of redundancy. This is also seen from the fact that for data lengths up to 2048 bits they require 3 to 6 check bits more compared to the DEC-TAEC BCH codes, i.e. 9 to 154 check-bits less than the DEC-TAEC OLS codes (Table IV).

When it comes to the processing of data bits, the difference between the mentioned codes is even more pronounced. First

TABLE IV

CHECK-BIT LENGTHS OF THE PROPOSED AND LINEAR DEC-TAEC CODES.

Codes	Data word length (bits)						
	32	64	128	256	512	1024	2048
Codes from [15]	12	14	16	18	20	22	24
Proposed codes	15	17	20	22	25	27	30
Codes from [16]	24	32	48	64	92	128	184

TABLE V

COMPARISON OF THE PROPOSED CODES AND LINEAR DEC-TAEC CODES.

Main characteristics	DEC-TAEC BCH codes	Proposed codes	DEC-TAEC OLS codes
Error correction capabilities	Correction of single, double and TA errors	Correction of single, double and TA _{b1} errors	Correction of single, double and TA errors
Processing of data bits	Finite field operations	Integer and LUT operations	Finite field operations
Interleaving	Causes extra delay and requires additional hardware	Do not cause extra delay and do not require additional hardware	Causes extra delay and requires additional hardware
Adaptation to the application requirements	Very complex	Simple	Complex
Preferred type of implementation	Hardware	Software	Hardware

of all, we have seen that the proposed codes use integer and LUT operations, which makes them suited for implementation on modern processors (Table V). Moreover, the analysis from Section 4 showed that they have the potential to achieve throughputs of several tens of Gbps. This result would be even better if multiple integer units (instead of one) per core were used (since Haswell all Intel processors have four integer units per core [17]). On the other hand, it is known that the linear DEC-TAEC codes use FF operations to encode/decode data bits. However, as these operations are not supported by GPPs, the codes from [15], [16] must be implemented in dedicated hardware (the software-based decoders need several tens of clock cycles to process one bit [18], [19]). In this regard, it should be noted that the DEC-TAEC OLS codes have several advantages over the DEC-TAEC BCH codes. One is that they can be decoded much faster than the DEC-TAEC BCH codes. Another advantage is that the DEC-TAEC OLS decoder can be implemented in modular form. This means that each added module provides extra correction ability allowing the receiver to switch between different EC modes. Although this feature is similar to that of the proposed codes, it cannot be efficiently exploited in modern OCNs. The reasons for this lie in the high

complexity of the decoder (it is necessary to process up to 12000 data bits) as well as high redundancy of the DEC-TAEC OLS codes.

Finally, the last difference between the mentioned codes concerns the error control capabilities. As shown in Section 2, the proposed codes were originally designed to correct all single, double and TA₁ errors. However, in Section 4 we saw that they have the ability to be interleaved without delay and without using any additional hardware. Thanks to this, it is possible to construct codes capable of detecting/correcting (multiple) random and TA errors. In addition, these codes can easily change the code rate and the mode of operation, which is very useful when transmitting different types of data. On the other hand, the linear DEC-TAEC codes have the ability to correct single, double and TA errors, but cannot easily change the code rate and/or the mode of operation. More precisely, they could be used for that purpose if different codecs were integrated into one chip. This would, however, lead to a significant increase in the price of network equipment. The similar applies to the use of interleaved codes, which are further characterized by delays in the processing of data bits.

VI. CONCLUSION

In this paper, we presented a class of integer codes that are suitable for use in optical computer networks where the data is transmitted serially. The presented codes use integer and table lookup operations, which makes them well suited for software implementation. Besides this, the presented codes have the ability to change the code rate and the mode of operation. In the error correction mode, which is suitable for real-time applications, the decoder can correct all single and double errors as well as all triple-adjacent errors within one b -bit byte. On the other hand, in the error detection mode, which is suited for non real-time applications, the decoder can detect all quadruple errors, all double triple-adjacent errors within one b -bit byte and all double triple-adjacent errors within two b -bit bytes. Finally, the proposed codes can be interleaved without delay and without using any additional hardware. Owing to this, it is possible to construct very simple codes capable of correcting (detecting) various types of errors, including all (double) triple-adjacent errors.

REFERENCES

- [1] R. Ramaswani, K. Sivarajan and G. Sasaki, *Optical Networks: A Practical Perspective*, 3rd ed., Elsevier, Inc., 2010.
- [2] P. Ciccarelli *et al.*, *Networking Basics*, 2nd ed., John Wiley & Sons, Inc., 2013.
- [3] R. Giladi, *Network Processors: Architecture, Programming, and Implementation*, Elsevier, Inc., 2008.
- [4] *Observations of Error Characteristics of Fiber Optic Transmission Systems*, CCITT Study Group XVIII Contrib. D21, San Diego, CA, USA, Jan. 1989.
- [5] W. D. Grover, "Effect of Error Correcting Code Using DS3 Framing Bits on Measured Dribble Error Pattern of 565 Mb/s Fibre Optic Transmission System," *Elect. Lett.*, vol. 28, no. 20, pp. 1869-1870, Sept. 1992.
- [6] T. Ono *et al.*, "Bit Error Statistical Analysis of Optical Transmission Systems," in D.W. Faulkner and A.L. Harmer (Eds.), pp. 43-49, IOS Press, 2000.
- [7] D. Mello, E. Offer, and J. Reichert, "Error Arrival Statistics for FEC Design in Four-Wave Mixing Limited Systems," in *Proc. Opt. Fiber Commun. Conf. (OFC)*, Mar. 2003, pp. 529-530.
- [8] L. James, "Error Behaviour in Optical Networks," Ph.D. dissertation, Dept. Eng., Univ. Cambridge, Cambridge, U.K., 2005.
- [9] A. Radonjic and V. Vujicic, "Integer Codes Correcting Burst Errors within a Byte," *IEEE Trans. Comput.*, vol. 62, no. 2, pp. 411-415, Feb. 2013.
- [10] A. Radonjic and V. Vujicic, "Integer Codes Correcting Spotty Byte Asymmetric Errors," *IEEE Commun. Lett.*, vol. 20, no. 12, pp. 2338-2341, Dec. 2016.
- [11] A. Radonjic and V. Vujicic, "Integer Codes Correcting High-Density Byte Asymmetric Errors," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 694-697, Apr. 2017.
- [12] A. Radonjic, "(Perfect) Integer Codes Correcting Single Errors," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 17-20, Jan. 2018.
- [13] A. Radonjic and V. Vujicic, "Integer Codes Correcting Burst and Random Asymmetric Errors within a Byte," *J. Franklin Inst.*, vol. 355, no. 2, pp. 981-996, Jan. 2018.
- [14] A. Radonjic and V. Vujicic, "Integer Codes Correcting Sparse Byte Errors," *Cryptogr. Commun.*, vol. 11, no. 5, pp. 1069-1077, Sept. 2019.
- [15] W. Wu *et al.*, "Method of Correcting Adjacent Errors By Using BCH-Based Error Correction Coding," US 2013/0262957 A1, Oct. 3, 2013.
- [16] S. Liu *et al.*, "Reducing the Cost of Triple Adjacent Error Correction in Double Error Correction Orthogonal Latin Square Codes," *IEEE Trans. Device Mater. Rel.*, vol. 16, no. 2, pp. 269-271, Jun. 2016.
- [17] A. Fog, "The Microarchitecture of Intel, AMD and via CPUs: An Optimization Guide for Assembly Programmers and Compiler Makers," Technical University of Denmark, Mar. 8, 2020. [Online]. Available: <https://www.agner.org/optimize/microarchitecture.pdf>
- [18] J. Cho and W. Sung, "Efficient Software-Based Encoding and Decoding of BCH Codes," *IEEE Trans. Comput.*, vol. 58, no. 7, pp. 878-889, July 2009.
- [19] A. Subbiah and T. Ogunfunmi, "A Flexible Hybrid BCH Decoder for Modern NAND Flash Memories Using GPGPUs," *Micromachines*, vol. 10, no. 6, pp. 1-15, Jun. 2019.