Egyptian Informatics Journal 23 (2022) 529–545

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Permutation-based frame synchronization method for data transmission systems with short packets

Jamil Al-Azzeh [a,*], Emil Faure [b,c], Anatoly Shcherba [b], Bohdan Stupka [b]

[a] Al-Balqa' Applied University, Amman 11134, Jordan
[b] Cherkasy State Technological University, 18006 Cherkasy, Ukraine
[c] The State Scientific and Research Institute of Cybersecurity Technologies and Information Protection of the State Service for Special Communications and Information Protection of Ukraine, 03142 Kyiv, Ukraine

ARTICLE INFO

Article history:
Received 28 November 2021
Revised 7 May 2022
Accepted 22 May 2022
Available online 3 June 2022

Keywords:
Permutation
Syncword
Frame synchronization
Short-packet communications
Factorial coding
Noise

ABSTRACT

This paper describes the research, development, and implementation of a frame synchronization method for data transmission systems with short packets, specifically those that use nonseparable factorial coding. The key concept of the method is using a permutation as a syncword. This permutation must satisfy the following condition: the minimum Hamming distance to all its circular shifts must be the maximum. The frame synchronization method employs correlation processing and majority processing of data fragments transmitted through the communication channel, where the fragment length is equal to the syncword length. As a result, the proposed frame synchronization method can realize frame synchronization in data transmission systems with adverse noise conditions. Theoretical and practical assessment of the number of accumulated fragments required to establish frame synchronization indicates that implementing the proposed method reduces the amount of received data, which subsequently reduces the time required to establish a connection between the transmitter and receiver. In addition, interleaving accumulated fragments further reduces the required number of received fragments.

1. Introduction

An efficient frame synchronization system that can detect frame boundaries at high bit error rates (BER) allows communication under high levels of natural or intentional noise. For example, such a system could facilitate communication under electronic warfare conditions. In addition, in such communication systems, combining channel error protection and unauthorized access protection into a single data structure realizes confidential information exchange.

This study reviews research aimed at developing a theory to construct telecommunication systems with nonseparable factorial coding. We continue the research started in [1], reveal, describe in detail, and improve the permutation based frame synchronization method.

Previous studies [2–5] have highlighted the basic principles of constructing nonseparable factorial codes and their characteristics. However, to the best of our knowledge, the procedure to establish frame synchronization for such systems with nonseparable factorial coding has not been studied extensively.

The frame synchronization [6–8] procedure is a required component of all network protocols [9–14]. Detecting frame boundaries at the receiver can be difficult due to low signal-to-noise ratios, which can be caused by electronic warfare and channel fading. Note that redundancy and forward error correction are common techniques to overcome channel fading.

When receiving a framed data stream, frame synchronization identifies incoming frame alignment signals (sync sequences or syncwords). Conventional frame formats ensure that a syncword

Abbreviations: ACF, Autocorrelation function; BER, Bit error rate; CDF, Cumulative distribution function; NOMA, Non-orthogonalnonorthogonal multiple access; SFD, Start of the frame delimiter.

* Corresponding author.
E-mail address: jamil.azzeh@bau.edu.jo (J. Al-Azzeh).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.05.005
1110-8665/

is sent in each frame and is utilized for channel estimation and frame synchronization.

Several methods have been proposed to synchronize codeword frames without using syncwords. For example, a previous study proposed a method [15] that is a variation of the brute-force approach. This method buffers two frame lengths of symbols and attempts decoding at each possible offset until an offset is determined for which decoding is successful. However, this method is computationally more complex and does not perform as well as frame synchronizers that utilize syncwords.

Another study [16] suggested XORing syncwords with data stream to increase the efficiency of frame synchronization. This technique reduces the amount of service information transmitted through the communication channel to establish message frame synchronization, thereby realizing more efficient allocation of communication channel resources. However, applying such an approach does not seem applicable in telecommunication systems with block and factorial codes [17–19] compared to data transmission systems with convolutional codes [20,21] because syncword superimposition on the information block can result in deformation of the data block and irrecoverable information loss. Thus, this method is not intended for frame synchronization in systems with factorial coding.

Frame synchronization methods, which do not rely on preamble symbols that introduce overhead in terms of energy consumption and channel utilization, deserve special attention. The core concept of such methods [22,23] is to omit the preamble and adapt the frame format; however, the adapted frame format provides a syncword [start of the frame delimiter (SFD)] in the frame structure.

Telecommunication systems with nonseparable factorial coding already utilize a nonstandard and redundant frame structure that does not provide a separate SFD field In addition, the codeword structure of a nonseparable factorial code allows it to function as a transport mechanism in short-packet communications [24–32], which is a feature of 5G wireless networks, sensor data networks, machine to machine communications. In such systems, the overhead for syncwords cannot be ignored [33–35].

A previous study [36] developed a framing method for data transmission systems that use nonseparable factorial coding. However, it has several disadvantages, which are summarized as follows.

- The convergence time of the synchronization procedure is not optimal because this method does not consider all redundant features of the factorial code's codeword structure.
- The probability of false synchronization is high. This feature becomes more noticeable as bit error probability increases and accumulation l value decreases. For example, for bit error probability $p_0 = 0.495$, the upper estimate of the probability of false synchronization is in the range $10^{-2} - 10^{-5}$ when accumulation factor $l = 3 \ldots 51$ fragments with a length of $n = M \cdot l_r = 24$ bits (for a permutation length of $M = 8$, where each symbol is encoded using a fixed length binary code). The results of an experimental study [36] indicated that the relative frequency of establishing false synchronization was $6.1 \cdot 10^{-3}$ for 10,000 tests and the indicated parameters.

The goal of this study is to increase the probability of frame synchronization for a given accumulation and reduce the probability of false synchronization under high noise conditions for short-packet data systems using nonseparable factorial coding.

To achieve this goal, it is necessary to solve the following tasks.

- A method and corresponding algorithm must be developed to establish frame synchronization for short-packet data systems using nonseparable factorial coding.
- The probability of establishing frame synchronization under noise conditions of various intensity must be assessed.
- A model that can implement the algorithm developed to establish frame synchronization must be constructed.
- A comparative analysis of the probability indicators of framing acquisition must be performed under various noise intensity conditions.

In this study, we will evaluate the efficiency of the permutation based frame synchronization method in a model data transmission system with the following constraints,

- The communication channel is binary symmetric with independent bit errors.
- The bit error probability in the communication channel is $p_0 \leqslant 0.495$.
- The probability of establishing correct synchronization is $P_{true} \geqslant 0.9997$ for any given value of bit error probability $p_0 \leqslant 0.495$.
- The probability of establishing false synchronization is.$P_{false} \leqslant 3 \cdot 10^{-4}$. for any given bit error probability value $p_0 \leqslant 0.495$.

This study proposed to use as a syncword a permutation where the minimum Hamming distance to all its circular shifts was the maximum. Furthermore, a correlation processing of data fragments received from the communication channel was introduced to recognize a syncword. Interleaving accumulated fragments accelerated the synchronization convergence. To evaluate the permutation-based frame synchronization method, a Python software model was constructed.

This paper is organized as follows: Section 2 provides the concepts for syncword selection and recognition, data stream correlation processing, theoretical assessment of probabilistic indicators, and details an outline of the permutation-based frame synchronization method; Section 3 presents a case study of the frame synchronization method; Section 4 gives the results of the permutation-based frame synchronization method evaluation through simulation; and Section 5 summarizes the findings and concludes the paper.

2. Materials and methods

As described in the literature [2], the codewords of a nonseparable factorial code are permutations. Here, each permutation π symbol is encoded with a fixed length binary code. The codeword length is $l_r = \lceil \log_2 M \rceil$, where M is a permutation length.

The proposed frame synchronization method, and an existing method [36] use a permutation π of length M as a syncword. In addition, these methods use majority processing of the received data. However, in contrast to the previous method [36], the proposed method employs fragment correlation processing.

The principle of constructing a frame synchronization system is demonstrated using an example syncword based on permutation π of length $M = 8$ (the sequence of decimal characters in a set $\{0, 1, 2, 3, 4, 5, 6, 7\}$).

Here, each symbol in the set is encoded with a fixed length binary code. The codeword length is $l_r = \lceil \log_2 M \rceil = 3$ ($n = M \cdot l_r = 24$), as shown in Table 1.

2.1. Selecting a syncword

Note that permutation π used as a syncword must satisfy the following condition: the minimum Hamming distance to all its circular shifts is the maximum.

Table 1
Permutation character encoding scheme.

Decimal notation	0	1	2	3	4	5	6	7
Binary notation	000	001	010	011	100	101	110	111

Let $\pi_i(j)$ be the permutation π_i circular shifted to the left by j bits, and let d_{ij} be the Hamming distance from π_i to its circular shift $\pi_i(j)$, where $0 \leqslant i \leqslant M!-1$ and $1 \leqslant j \leqslant n-1$. In addition, let $d_i = \min_j (d_{ij})$ and $d = \max_i (d_i) = \max_i \left(\min_j (d_{ij}) \right)$. Then the syncword can be any permutation π_i where $\forall d_{ij} \geqslant d$.

Remark 1. Due to the n-periodicity of the distances d_{ij} relative to the circular shifts, it follows that for $\forall j,k,\ j \neq k,\ 1 \leqslant j,k \leqslant n-1$ and $d(\pi_i(j),\pi_i(k)) \geqslant d$ is true for the Hamming distance $d(\pi_i(j),\pi_i(k))$ between circular shifts $\pi_i(j)$ and $\pi_i(k)$ if $d_{ij} \geqslant d$ for $\forall j \in [1, n-1]$.

Theorem 1. *If $n > 2$, there does not exist a permutation π_i for which $2d_i > n+1$, i.e., $d \leqslant (n+1)/2$.*

Proof. The value d_i is equivalent to the coding distance of the code, in which combinations are formed by all n circular shifts of the binary-represented permutation π_i. Thus, for such a code, it is necessary to determine the Plotkin bound relative to the code distance d_i. To achieve this, the ratios available in the proof of Theorem 1 (Plotkin bounds) are applicable [37]. For every code (n, M, d), where n is the code combination length, M is the code size, d is the code distance, the inequality $M(M-1)d \leqslant nM^2/2$ holds if M is an even number. For code combinations in the form of circular shifts of the binary-represented permutation π_i, $M = n$ is true. Then, $n(n-1)d_i \leqslant n^3/2$ or $2d_i \leqslant n+1+1/(n-1)$ using $n > 1$. Here, $1/(n-1) < 1$ if $n > 2$; thus, $2d_i \leqslant n+1$.

With odd number values of M, the inequality $M(M-1)d \leqslant n\left(M^2-1\right)/2$ is applied. If $M = n$, then $n(n-1)d_i \leqslant n(n^2-1)/2$ or $2d_i \leqslant n+1$ if $n > 1$.

Thus, if $n > 2$, then $d \leqslant (n+1)/2$. □.

Remark 2. For $M = 8$ and $n = 24$, the inequality $d \leqslant (n+1)/2$ is formulated as $d \leqslant 12$.

The distribution of the d_i values for every permutation of length $M = 8$ is shown in Fig. 1.

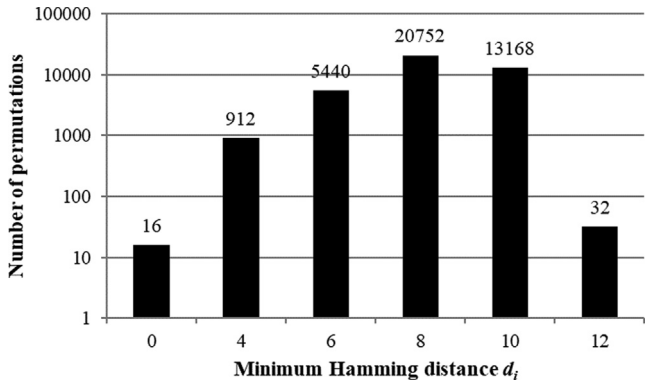

Fig. 1. Relationship between the distribution of permutations number and distance d_i.

From the distribution information shown in Fig. 1, it follows that $d = \max_i (d_i) = 12$ for $M = 8$. Here, the number of permutations with $d = 12$ is 32.

Remark 3. The number d_i corresponding to permutation π_i is invariant relative to its symbol (codeword) circular shifts, the inversions of bits, and the reverse order of bits. Based on Remark 3, the 32 permutations with $d = 12$ can be obtained appropriately from a single permutation, as shown in Table 2.

In the next step, we constructed a normalized autocorrelation function (ACF) for the permutation given in Table 2. Here, we calculate the autocorrelation coefficients as follows:

$$\rho_j = \frac{1}{n}(n - 2d_{ij})$$

Fig. 2 shows the normalized ACF graph for the permutation given in Table 2.

The number of shifts from 0 to 1 and from 1 to 0 in the binary notation of the permutation in Table 2 is 12. From these data, it is apparent that the performance indicators to establish and maintain clock synchronization are slightly lower than those presented in Table 2 in the literature [36]. This indicates that 33% more time is required to accumulate a certain predetermined number of significant modulation moments for the permutation in Table 2 in this paper compared to the permutations in Table 2 in the literature [36].

2.2. Syncword recognition

Here, we describe the majority processing of the received data to increase data reliability, which was also implemented in the previous study [36]. Note that this processing method allows for repetition and accumulation of syncwords. Let l be an accumulation factor defining the number of accumulated bit fragments, where the fragment length is equal to the syncword length. In addition, the accumulation factor is an odd number. Majority processing defines the refined sequence R. According to this procedure, each error with multiplicity up to $(l-1)/2$ inclusive that occurs in the corresponding bits of the received fragments is corrected. Thus, the maximum total number of bit errors that majority processing can correct is $((l-1)/2) \cdot n = ((l-1)/2) \cdot M\lceil \log_2 M \rceil$ among the received ln bits.

2.3. Correlation processing

After majority processing, the receiver defines Hamming distances to every syncword circular shift from the sequence R. If some Hamming distance is less than or equal to d_{lim} for some syncword circular shift, the receiver identifies the resulting sequence R with this shift.

Using the permutation in Table 2 as the syncword, the Hamming distance between the syncword and its circular shifts is not

Table 2
Permutation π_i, where $\forall d_{ij} \geqslant d$.

Decimal notation	Binary notation
(0, 1, 7, 3, 2, 5, 4, 6)	(000, 001, 111, 011, 010, 101, 100, 110)

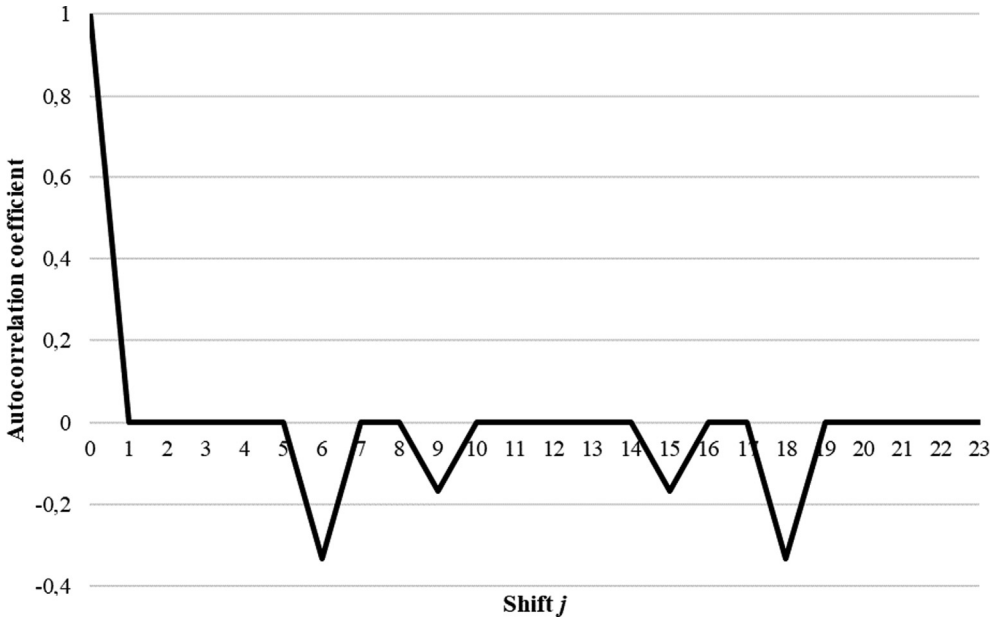

Fig. 2. Normalized ACF graph for permutation given in Table 2.

less than 12; thus, each error in the refined sequence R with multiplicity up to $d_{lim} = 5$ is corrected.

2.4. Probability of bit error in sequence R

Here, assume that the communication channel is binary symmetric with independent bit errors. Then, the result of majority processing allows us to reduce the probability of the bit error to the value as follows:

$$p_0^* = \sum_{i=(l+1)/2}^{l} C_l^i p_0^i (1-p_0)^{l-i} \tag{1}$$

In cases where $l \geqslant 1027$ (this value would be necessary when the bit error probability is high), it is impossible to perform the calculation using Eq. (1) by standard means. Therefore, we define an approximation for the bit error probability p_0^*.

Theorem 2. *The probability of bit error after majority processing of l fragments can be approximated as follows:*

$$p_0^* \cong \Phi(x) + \frac{1-2p_0}{6\sqrt{l p_0(1-p_0)}} \cdot (x^2-1) \cdot \varphi(x) \tag{2}$$

where $\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt$ denotes the integral Laplace function, $\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ is the differential Laplace function, $x = -\sqrt{l}\frac{0.5-p_0}{\sqrt{p_0(1-p_0)}}$, and the channel bit error probability $p_0 \in (0; 0.5)$.

In this case, the approximation accuracy is given as follows:

$$|\varepsilon| \leqslant \frac{1}{12\pi p_0(1-p_0)} \cdot \frac{l}{(l-2)(l-3)} + \frac{(1-2p_0)^2}{9\pi p_0(1-p_0)} \cdot \frac{1}{l} + \\ + \frac{1-2p_0}{4\sqrt{2\pi p_0(1-p_0)}} \cdot \frac{l}{(l-2)(l-3)\sqrt{l-4}} + \frac{(1-2p_0)^2}{9\pi p_0(1-p_0)} \cdot \frac{l}{(l-1)(l-2)(l-3)}. \tag{3}$$

Proof. The distribution $F_l(x)$ can be approximated as follows (refer to Equations (36) and (40) in the literature [38]):

$$F_l(x) \cong \Phi(x) - \frac{\theta_3}{3!\sqrt{l}} \cdot H_2(x) \cdot \varphi(x)$$

where θ_3 is the third moment of the Chebyshev–Hermite distribution; $H_2(x) = x^2 - 1$ is the Chebyshev–Hermite polynomial, $H_k(x) = (-1)^k \varphi^{(k)}(x)/\varphi(x)$.

For the Bernoulli distribution, where zero is the mean and one is the variance, i.e., $\theta_3 = \alpha_3$, where $\alpha_3 = -(1-2p_0)/\sqrt{p_0(1-p_0)}$ is the third moment (asymmetry) of the distribution. Eq. (2) follows from these expressions.

We obtain the approximation accuracy [Eq. (3)] by substituting the expression for the absolute fourth moment of the Bernoulli distribution function $\beta_4 + 3 = 1/(p_0(1-p_0))$ and inserting the expressions for the third, fourth, and fifth absolute moments of the standard normal distribution divided by $\sqrt{2\pi}$, $B_3 = 2/\pi$, $B_4 = 3/\sqrt{2\pi}$, $B_5 = 8/\pi$ into Equation (40) from the literature [38]. □.

Using Eq. (3), we can define and present an estimate of the approximation accuracy [Eq. (2)] with $l \geqslant 1027$ for different p_0 values (Table 3).

2.5. Correct synchronization probability

Correct synchronization probability is the probability of errors in $v \leqslant d_{lim}$ bits in R:

$$P_{true}(n, d_{lim}, p_0, l) = \sum_{v=0}^{d_{lim}} C_n^v \left(p_0^*\right)^v \left(1-p_0^*\right)^{n-v} \tag{4}$$

For the syncword in Table 2, $P_{true}(24, 5, p_0, l) = \sum_{v=0}^{5} C_{24}^v \left(p_0^*\right)^v \left(1-p_0^*\right)^{24-v}$.

Graphs $P_{true}(24, 5, p_0, l)$ for $p_0 \in \{0.1, 0.2, 0.3, 0.4, 0.43, 0.45, 0.47, 0.475, 0.48, 0.485, 0.49, 0.495\}$ depending on $l = 1, 3, 5, \ldots, 1001$ are shown in Fig. 3.

Note that the bit error probability is not given a priori; thus, the frame synchronization procedure should be adaptive. Based on the dependencies shown in Fig. 3, the accumulation factor can vary widely to achieve a given probability of correct synchronization. In other words, to achieve $P_{true} \geqslant 0.9997$ when $p_0 = 0.1$, the accumulation factor $l = 3$ is required, when $p_0 = 0.3$, factor $l = 19$ is required, and when $p_0 = 0.45$, factor $l = 305$ is required. The

Table 3
Estimation of approximation accuracy of Eq. (2) with $l \geqslant 1027$.

p_0	0,4	0,43	0,45	0,47	0,475	0,48	0,485	0,49	0,495
$\lvert\varepsilon\rvert \leqslant$	$1.15 \cdot 10^{-4}$	$1.1 \cdot 10^{-4}$	$1.07 \cdot 10^{-4}$	$1.05 \cdot 10^{-4}$		$1.04 \cdot 10^{-4}$			

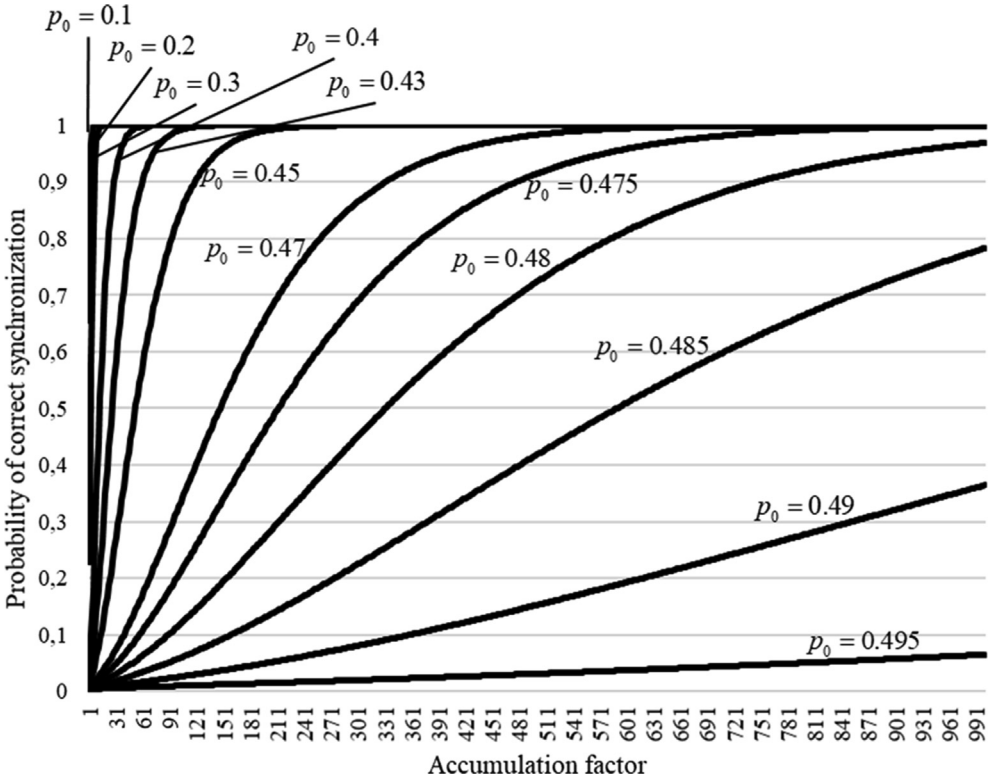

Fig. 3. Dependencies of correct synchronization probability on accumulation factor for different bit error probabilities.

permutation-based frame synchronization method provides accumulation of n-bit fragments from the channel, and the corresponding majority and correlation processing. The maximum bit error probability value and minimum synchronization probability value designed for the synchronization system determine the maximum l value. If there is no synchronism at maximum l, the synchronization system generates the channel failure signal. As indicated below, for $n = 24$, $d_{lim} = 5$, and under the given constraints $P_{true} \geqslant 0.9997$ and $p_0 \leqslant 0.495$, the maximum accumulation factor l value is limited to 30,603.

In addition, while devising the procedure to establish frame synchronization, it is necessary to count false synchronization occurrences. Here, false synchronization probability increases with a higher probability p_0 and with a lower accumulation factor l. In this case, the frame synchronization procedure should ensure that the probability of establishing false synchronism cannot exceed a given threshold for any arbitrarily high probability of a bit error.

Having defined the probability of correct synchronization, we discuss the false synchronization probability in the following.

2.6. False synchronization probability

The upper estimate for false synchronization probability can be calculated approximately as follows:

$$P_{false}(n, d_{lim}, p_0, l) \leqslant 1 - P_{true}(n, d_{lim}, p_0, l) \tag{5}$$

This estimation considers the possibility of converting the transmitted syncword into any $2^n - \sum_{v=0}^{d_{lim}} C_n^v$ sequence. Simultaneously, a false synchronization event will occur if a communication channel error transforms the transmitted syncword into any of the sequences located at a Hamming distance not exceeding d_{lim} from the possible syncword circular shifts (in geometric interpretation, within the spheres of n-dimensional space with centers at the points corresponding to the syncword circular shifts and radius d_{lim}). Here, the number of such possible sequences is $(n-1)\sum_{v=0}^{d_{lim}} C_n^v$. While calculating the relation $(n-1)\sum_{v=0}^{d_{lim}} C_n^v / \left(2^n - \sum_{v=0}^{d_{lim}} C_n^v\right)$, for example, for $M = 8$, $n = 24$, and $d_{lim} = 5$, we obtain the value 0.076. Note that the points inside the spheres of radius d_{lim} are not equally probable when $0 < p_0 < 0.5$; thus, the assessment in Eq. (5) is imprecise. Next, we calculate the exact probability rate of false synchronization.

Theorem 3. *False synchronization probability*.

$$P_{false}(n, d_{lim}, p_0, l) = \sum_{j=1}^{n-1} \left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}} C_{d_{ij}}^v \left(\sum_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^w \left(p_0^*\right)^{v+w} \left(1-p_0^*\right)^{n-(w+v)} \right) \right) \tag{6}$$

Proof. The probability of an error leading to an incorrect decision and establishing false synchronization in the refined sequence R is the probability of occurrence of any error vector that transforms syncword π_i into any of its circular shifts $(1 \leqslant j \leqslant n-1)$ in the refined sequence R with accuracy up to d_{lim} bits.

As discussed previously, d_{ij} is the Hamming distance from syncword π_i to its circular shift by j bits $\pi_i(j)$, $1 \leqslant j \leqslant n-1$. In this case, the error translates permutation π_i into its circular shift $\pi_i(j)$ if it contains v errors in d_{ij} bits of difference between these two sequences, while $d_{ij} - d_{\lim} \leqslant v \leqslant d_{i,j}$. In addition, in the remaining $(n - d_{ij})$ bits, the occurrence of w more bit errors is possible, whereas $0 \leqslant w \leqslant v - (d_{ij} - d_{\lim})$. The latter restraint is due to the fact that, for false synchronization, the permutation modified by the error should not differ by greater than d_{lim} bit from the circular shift of permutation π_i.

The probability of the above event described is given as follows:

$$P_{false}(n, d_{lim}, p_0, l) = \sum_{j=1}^{n-1} \left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}} \left(\begin{array}{c} C_{d_{ij}}^{v} (p_0^*)^v (1-p_0^*)^{d_{ij}-v} \times \\ \times \sum\limits_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^{w} (p_0^*)^w (1-p_0^*)^{n-d_{ij}-w} \end{array} \right) \right) \tag{7}$$

By grouping the factors in Eq. (7), we obtain an expression to calculate the false synchronization probability [Eq. (6)]. □.

Remark 4. For the syncword in Table 2 with $d_{lim} = 5$, values $d_{ij} = 12$ are registered in 19 cases, values $d_{ij} = 14$ are registered in two cases, and values $d_{ij} = 16$ are registered in two cases (refer to the normalized ACF graph in Fig. 2). Thus, after grouping the terms, Eq. (6) for the syncword in Table 2 holds:

$$\begin{aligned} P_{false}(24,5,p_0,l) = {} & 19 \sum_{v=7}^{12} C_{12}^{v} \left(\sum_{v=7}^{12} C_{12}^{v} (p_0^*)^{v+w} (1-p_0^*)^{24-v-w} \right) \\ & + 2 \sum_{v=9}^{14} C_{14}^{v} \left(\sum_{w=0}^{v-9} C_{10}^{w} (p_0^*)^{v+w} (1-p_0^*)^{24-v-w} \right. \\ & \left. + 2 \sum_{v=11}^{16} C_{8}^{w} (p_0^*)^{v+w} (1-p_0^*)^{24-v-w} \right) \end{aligned} \tag{8}$$

Remark 5. When performing numerical calculations for the false synchronization probability $P_{false}(n, d_{lim})$, Eq. (8) is expressed in the model example of this study.

Graphs $P_{false}(24,5,p_0,l)$ for $p_0 \in \{0.1, 0.2, 0.3, 0.4, 0.43, 0.45, 0.47, 0.475, 0.48, 0.485, 0.49, 0.495\}$ depending on $l = 1, 3, 5, \ldots, 1001$ are shown in Fig. 4.

Analysis of the graphs shown in Fig. 4 leads to the following. While increasing accumulation factor l and calculating the distances to all syncword circular shifts, the probability of false synchronization exceeds the maximum permissible value with an unknown bit error probability p_0. For example, if the analysis of the refined sequence begins with accumulation factor $l = 3$ and $p_0 = 0.4$, the probability of false synchronization for $l = 3$ is $P_{false} = 0.046$. Obviously, this probability is unacceptable. Thus, the following approach is employed to reduce this probability.

2.7. Reducing false synchronization probability

The receiver groups the fragments received from the channel into K blocks by l fragments. Here, the refined sequences R_k, where $k \in [1, K]$, are calculated independently for each block. Synchronism is established if each of the sequences R_k corresponds to the same circular shift of syncword.

Then, the correct synchronization probability [refer to Eq. (4)] is obtained as follows:

$$P_{true}(n, d_{lim}, p_0, l, K) = \left(\sum_{v=0}^{d_{lim}} C_n^v (p_0^*)^v (1-p_0^*)^{n-v} \right)^K \tag{9}$$

From Eq. (6), the probability of false synchronization changes to:

$$P_{false}(n, d_{lim}, p_0, l, K) = \sum_{j=1}^{n-1} \left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}} C_{d_{ij}}^{v} \sum_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^{w} (p_0^*)^{v+w} (1-p_0^*)^{n-v-w} \right)^K \tag{10}$$

Eq. (10) can be explained as follows. The probability of converting permutation π_i into its circular shift $\pi_i(j)$ is equal [refer to Eq. (6)]$P_{false\,j}(n, d_{lim}, p_0, l) = \sum_{v=d_{ij}-d_{lim}}^{d_{ij}} C_{d_{ij}}^{v} \left(\sum_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^{w} (p_0^*)^{v+w} (1-p_0^*)^{n-(w+v)} \right)$. The probability that the channel noise will transform a permutation π_i into its circular shift $\pi_i(j)$ in K blocks is equal to $P_{false\,j}^K(n, d_{lim}, p_0, l)$. Thus, the false synchronization probability for the K blocks is given as:

$$\begin{aligned} P_{false}(n, d_{lim}, p_0, l, K) = {} & P_{false\,1}^K(n, d_{lim}, p_0, l) \\ & + P_{false\,2}^K(n, d_{lim}, p_0, l) + \ldots + \\ & + P_{false\,(n-1)}^K(n, d_{lim}, p_0, l) \\ = {} & \sum_{j=1}^{n-1} P_{false\,j}^K(n, d_{lim}, p_0, l). \end{aligned}$$

Graphs of $P_{true}(24,5,p_0,l,K)$ and $P_{false}(24,5,p_0,l,K)$ for $p_0 = 0.495$ and $K \in \{1,2,3,4\}$ depending on accumulation factor $l = 1, 3, 5, \ldots, 30603$ are shown in Fig. 5 and Fig. 6.

Dependency graphs $P_{true}(24,5,p_0,l,K)$ and $P_{false}(24,5,p_0,l,K)$ for $p_0 = 0.495$ and $K \in \{1,2,3,4\}$ depending on the number of accumulated fragments, $L = l \cdot K = 1, 3, 5, \ldots, 30603$ are shown in Fig. 7 and Fig. 8.

Fig. 6 and Fig. 8 show that the values are $P_{false}(24,5,p_0,l,4) \leqslant 2.68 \cdot 10^{-9}$, $P_{false}(24,5,p_0,l,3) \leqslant 8.15 \cdot 10^{-7}$, $P_{false}(24,5,p_0,l,2) \leqslant 2.48 \cdot 10^{-4}$, and $P_{false}(24,5,p_0,l,1) \leqslant 7.55 \cdot 10^{-2}$ for $p_0 = 0.495$.

In the following, we perform an upper estimate for the probability $P_{false}(n, d_{lim}, p_0, l, K)$ calculated using Eq. (10).

Theorem 4. *The evaluation of false synchronization probability $P_{false}(n, d_{lim}, p_0, l, K)$ can be expressed as follows:*

$$P_{false}(n, d_{lim}, p_0, l, K) \leqslant \max_s \{p(n, d_{lim}, K, s)\}, \tag{11}$$

where

$$p(n, d_{lim}, K, s) = \sum_{j=1}^{n-1} \left(\sum_{v=d_{ij}-d_{lim}}^{d_{ij}} C_{d_{ij}}^{v} \sum_{w=0}^{v-d_{ij}+d_{lim}} C_{n-d_{ij}}^{w} F(n, v+w, s) \right)^K, \tag{12}$$

$$F(n, v+w, s) = \begin{cases} (s/Nn)^{v+\omega} (1 - s/Nn)^{n-(v+\omega)}, & N(v+w) \leqslant s; \\ ((s+1)/Nn)^{v+\omega} (1 - (s+1)/Nn)^{n-(v+\omega)}, & \\ N(v+w) \geqslant s+1; & \end{cases}$$

N is a number of segments into which the segments $[z/n; (z+1)/n]$ are divided;
z is a natural number, $d - d_{lim} \leqslant z \leqslant \lceil n/2 \rceil - 1$;
s is a natural number, $N(d - d_{lim}) \leqslant s \leqslant \lceil Nn/2 \rceil - 1$.

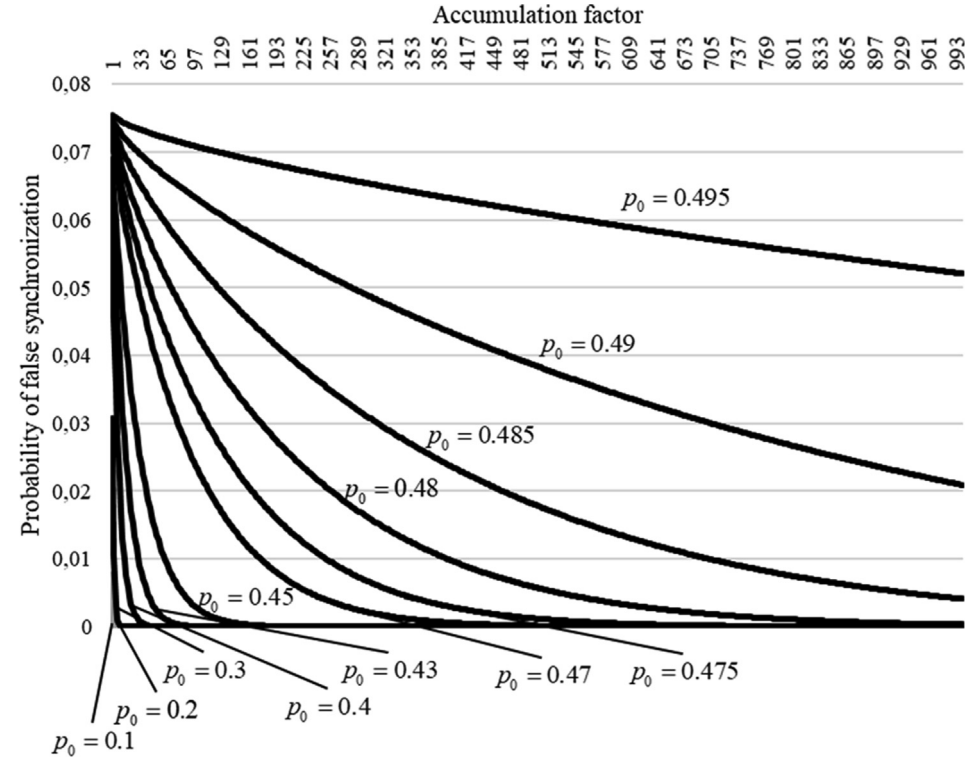

Fig. 4. Dependencies of false synchronization probability for different bit error probabilities on the accumulation factor.

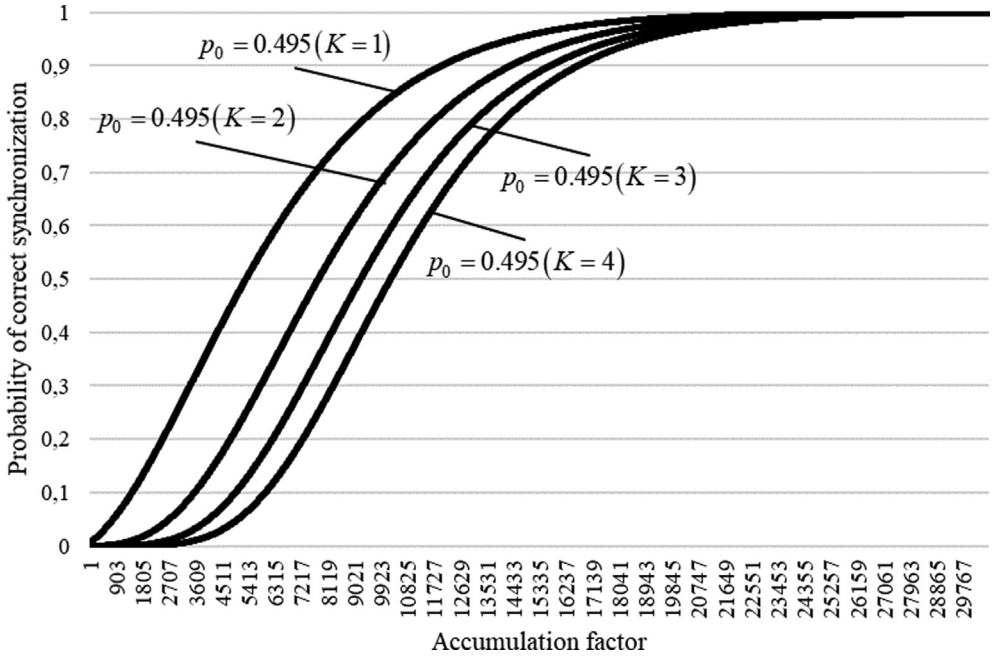

Fig. 5. Dependencies of correct synchronization probability on accumulation factor for bit error 0.495 and various values of $K \in \{1, 2, 3, 4\}$.

Proof. Here, consider the behavior of the function $f(n, v+w, p_0^*) = (p_0^*)^{v+w}(1-p_0^*)^{n-v-w}$. The graphs $f(n, v+w, p_0^*)$ for $n = 24$ and $v+w = \{7, 9, 12, 15, 17\}$ depending on p_0^* are shown in Fig. 9, where the range of values $p_0^* \in (0, 1)$ is employed to improve the visualization of the relationship.

In the following step, it is necessary to determine the point p_0^* at which function $f(n, v+w, p_0^*)$ obtains the maximum value. Here, we assume that $g(n, v+w, p_0^*) = \ln f(n, v+w, p_0^*) = (v+w)\ln p_0^* + (n-v-w)\ln(1-p_0^*)$. Then, $g'(n, v+w, p_0^*) = (v+w-np_0^*)/p_0^*(1-p_0^*)$, and $g'(n, v+w, p_0^*) = 0$ when $p_0^* = (v+w)/n$. Thus, we obtain $\max(f(n, v+w, p_0^*)) = f(n, v+w, (v+w)/n)$.

For the model example in this study, the maximum of the functions $f(n, v+w, p_0^*)$, $d - d_{lim} \leqslant v+w < n/2$ is located in $p_0^* \in [7/24; 1/2)$. When $v+w \geqslant n/2$, the functions increase monotonically over the interval $(7/24; 1/2)$.

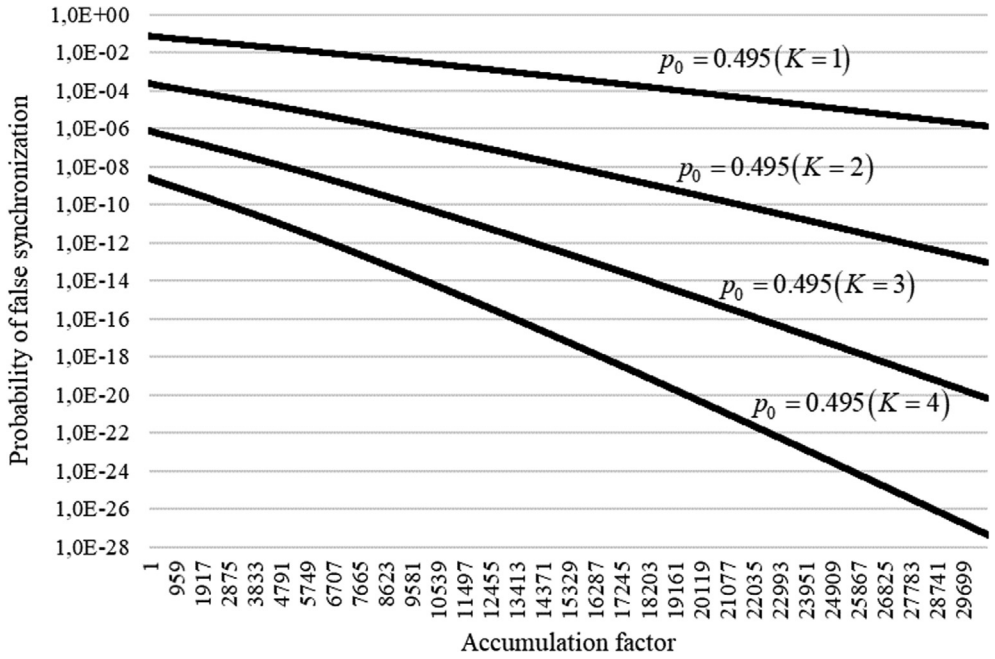

Fig. 6. Dependencies of false synchronization probability on accumulation factor for bit error 0.495 and various values of $K \in \{1, 2, 3, 4\}$.

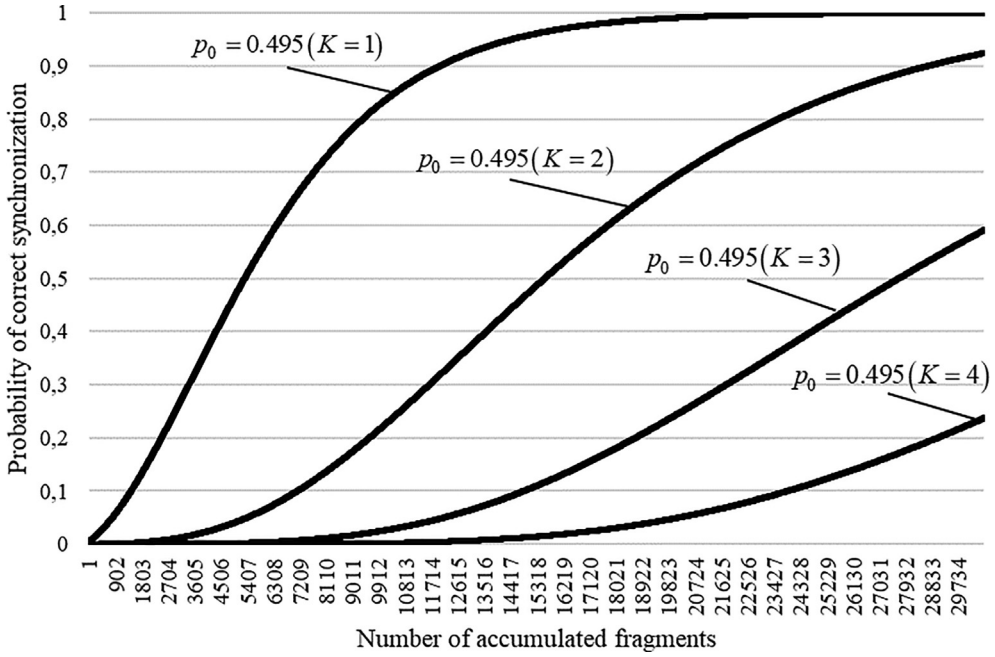

Fig. 7. Dependencies of correct synchronization probability on the number of accumulated fragments for bit error 0.495 and various values of $K \in \{1, 2, 3, 4\}$.

Each function $f(n, v + w, p_0^*)$ is monotonic in the intervals $(z/n; (z+1)/n)$, where z is a natural number $d - d_{lim} \leqslant z \leqslant \lceil n/2 \rceil - 1$. Each segment $[z/n; (z+1)/n]$ is divided into N segments $[z/n + i/Nn; z/n + (i+1)/Nn] 0 \leqslant i \leqslant N - 1$. For each natural number $s, N(d - d_{lim}) \leqslant s \leqslant \lceil Nn/2 \rceil - 1$, and parameter values $p_0^* \in [s/Nn; (s+1)/Nn]$, the following constraints are applied:

$$f(n, v + w, p_0^*) = (p_0^*)^{v+w}(1 - p_0^*)^{n-v-w} \leqslant F(n, v + w, s) \qquad (13)$$

where $F(n, v + w, s) = \{ (s/Nn)^{v+\omega}(1 - s/Nn)^{n-(v+\omega)}, N(v + w) \leqslant s;$ $((s+1)/Nn)^{v+\omega}(1 - (s+1)/Nn)^{n-(v+\omega)}, N(v + w) \geqslant s + 1..$

Estimates (12) for $p(n, d_{lim}, K, s)$ and (11) for $P_{false}(n, d_{lim}, p_0, l, K)$ are formed sequentially based on relations (10) and (13). □.

For the example model, we split segment $p_0^* \in [7/24, 1/2]$ into $\lceil Nn/2 \rceil - N(d - d_{lim}) = \lceil 2 \cdot 24/2 \rceil - 2(12 - 5) = 10$ segments with a step of 1/48. Then, $s = 14, 15, \ldots, 23$.The upper estimates of the probability $P_{false}(24, 5, p_0, l, K)$ for $K \in \{1, 2, 3, 4\}$ are derived based on Eqs. (11) and (12), as shown in Table 4.

The given values indicate that the initial value of K can be selected even for $p_0^* \to 1/2$ such that the probability of false synchronization satisfies the given requirements.

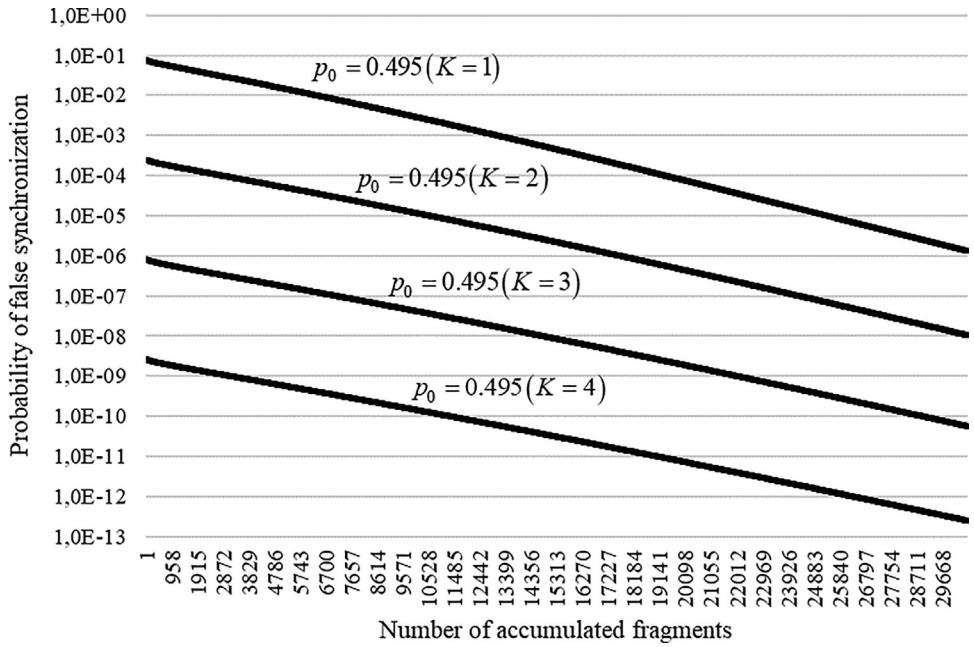

Fig. 8. Dependencies of false synchronization probability on the number of accumulated fragments for bit error 0.495 and various values of $K \in \{1, 2, 3, 4\}$.

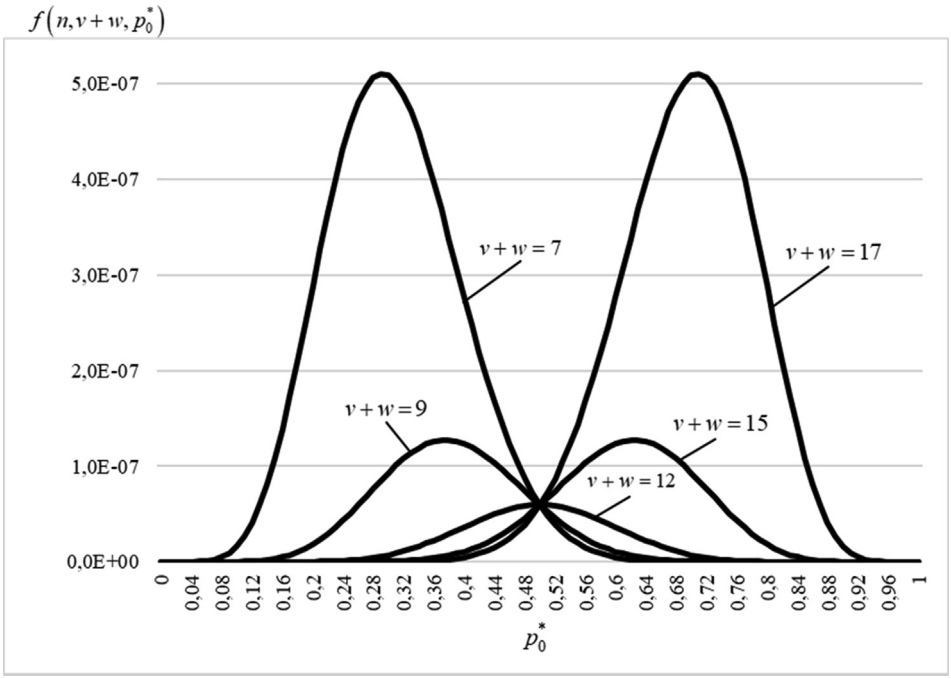

Fig. 9. Dependencies of $f(n, v+w, p_0^*)$ from p_0^* for $n = 24$ and $v+w = \{7, 9, 12, 15, 17\}$.

Remark 6. Eqs. (9) and (10), as well as Eqs. (4) and (6), determine the probabilities of correct and false synchronization "in the point," i.e., for a separate test with given values of accumulation factor l and the number of blocks K, while ignoring the procedure of sequential l increase and K variation.

2.8. Assessments of interval probabilities of synchronization

The synchronization result for the current value of l depends on the synchronization results for the previous values of l. This is due to the fact that the statistics in the accumulated segments within a single block may differ slightly.

The probability P_{true_final} of correct synchronization after the accumulation factor has reached the value of l is estimated by the correct synchronization probability for a fixed value of l as follows:

$$P_{true\_final}(n, d_{lim}, p_0, l, K) \geqslant P_{true}(n, d_{lim}, p_0, l, K) \quad (14)$$

The probability P_{false_final} of false synchronization after the accumulation factor has reached the value of l for $K = 1$ is estimated from the above as follows:

Table 4
Upper estimates of false synchronization probability for $n = 24$, $d_{lim} = 5$.

K	1	2	3	4
$P_{false}(24, 5, p_0, l, K) \leqslant$	$8.03 \cdot 10^{-2}$	$2.80 \cdot 10^{-4}$	$9.78 \cdot 10^{-7}$	$3.42 \cdot 10^{-9}$

$$P_{false\_final}(n, d_{lim}, p_0, l, K) = \sum_{i=1}^{K} P_{false\_sum}(n, d_{lim}, p_0, l_{\min}(i), l_{\max}(i), i), \tag{15}$$

where

$$P_{false\_sum}(n, d_{lim}, p_0, l_{\min}(i), l_{\max}(i), i) \leqslant \sum_{j=l_{\min}(i)}^{l_{\max}(i)} P_{false}(n, d_{lim}, p_0, j, i), \tag{16}$$

$P_{false\_sum}(n, d_{lim}, p_0, l_{\min}(i), l_{\max}(i), i)$ determines the false synchronization probability for i blocks;$l_{\min}(i) \leqslant l \leqslant l_{\max}(i)$;$l_{\min}(K) = 1$;$l_{\max}(1) = l$.

Remark 7. $P_{true_final}(n, d_{lim}, p_0, l, K)$and $P_{false\_final}(n, d_{lim}, p_0, l, K)$ are in fact estimates of the cumulative distribution function (CDF) [39] of the number $L_{fr} = Kl$ prior to establishing the correct or false synchronism, respectively.

2.9. Selecting K and l

The values of K and the segment $[l_{\min}(i), l_{\max}(i)]$ bounds are selected to satisfy the inequality $P_{false\_final}(n, d_{lim}, p_0, l, K) \leqslant P_{false\_\max}$, where $P_{false\_\max}$ is the limit value for the probability of false synchronization, and $p_{0\_\max}$ is the limit value for the bit error probability.

Based on Eq. (15), the constraints for the individual terms $P_{false\_final}(n, d_{lim}, p_0, l, K)$ are defined as follows:

$$P_{false\_sum}(n, d_{lim}, p_0, l_{\min}(i), l_{\max}(i), i) \leqslant \gamma_i \cdot P_{false\_\max} \tag{17}$$

In addition, $\gamma_i \geqslant 0$, $\sum_{i=1}^{K} \gamma_i = 1$.

Note that defining the bounds for segments $[l_{\min}(i), l_{\max}(i)]$ begins with $i = 1$.

Then $l_{\max}(1) = \min(l) : P_{true}(n, d_{lim}, p_0, l, 1) \geqslant P_{true\_\min}$. Thus, via an analogy with Eq. (14), the following inequality holds:$P_{true\_final}(n, d_{lim}, p_0, l_{\max}(1), K) \geqslant P_{true\_min}$. For the example model with $n = 24$, $d_{lim} = 5$ for $P_{true\_\min} = 0.9997$, and $p_{0\_\max} = 0.495$, we obtain $l_{\max}(1) = 30603$.

The values of $l_{\min}(i) = \min(l) : P_{false\_sum}(n, d_{lim}, p_0, l, l_{\max}(i), i) \leqslant \gamma_i \cdot P_{false\_\max}$.

The values $l_{\min}(i)$ define the upper bounds $l_{\max}(i+1)$: $l_{\max}(i+1) = [l_{\min}(i) \cdot (i/(i+1))] - prtsgn$ where $prtsgn = \begin{cases} 0 \text{ if } [l_{\min}(i) \cdot (i/(i+1))] \text{ is odd}; \\ 1 \text{ otherwise}. \end{cases}$.

Here, the method used to select the values of K, $l_{\min}(i)$, $l_{\max}(i)$, $i \in [1; K]$ applies to any given constraint.

For the example model, $P_{false\_\max} = 3 \cdot 10^{-4}$. According to Table 4 and from Eq. (16), we obtain $P_{false\_sum}(24, 5, 0.495, 1, 30603, 4) \leqslant 1.05 \cdot 10^{-4}$. Thus, $K = 4$, and $P_{false\_sum}(24, 5, 0.495, 1, l_{\max}(4), 4) \ll \sum_{i=1}^{3} P_{false\_sum}(n, d_{lim}, p_0, l_{\min}(i), l_{\max}(i), i)$.

Here, assume that $\gamma_1 = 1/8$, $\gamma_2 = 3/4$, and $\gamma_3 = 1/8$. Then, the boundaries of the segments $[l_{\min}(i), l_{\max}(i)]$ take the values given in Table 5.

The number of accumulated fragments is $L_{fr} = Kl$, and the number of accumulated bits is $L = Kln$.

Table 5
Segment boundaries $[l_{\min}(i), l_{\max}(i)]$.

i	1	2	3	4
$l_{\min}(i)$	30,549	10,457	2779	1
$l_{\max}(i)$	30,603	15,273	6971	2083

Fig. 10 shows the graphs for the dependencies of estimates of the correct and false synchronization probabilities calculated using Eqs. (14) and (15) on the number of accumulated fragments under the given constraints for $K = \{1, 2, 3, 4\}$ and the values defined for the transition points.

The graphs in Fig. 10 define the adaptive synchronization process at an unknown level of bit error probability $p_0 \leqslant 0.495$ for $P_{true} \geqslant 0.9997$ and $P_{false} \leqslant 3 \cdot 10^{-4}$.

Based on the approach employed to determine frame synchronism, we formulate the stages stipulated by the proposed method in the following.

2.10. Permutation-based frame synchronization method

The permutation-based frame synchronization method is realized through the following stages.

- The transmitter sequentially issues a syncword into the communication channel. The syncword is a permutation π of length M, where the minimum Hamming distance d to all of its circular shifts is the maximum. For example, for $M = 8$, such a syncword is the permutation $\pi = (000, 001, 111, 011, 010, 101, 100, 110)$ with precision up to its circular shift by the number of bits that is a multiple of the symbol length in its binary representation $l_r = 3$, bit inversion, and their reverse order.
- The receiver accumulates K blocks received from a communication channel comprising l fragments of n bits. Here, the K and l values are transformed according to the above procedure.
- For each block, the refined sequences R_k, where $k \in [1, K]$, are calculated independently. Each bit of this sequence is calculated on a majority basis based on the corresponding bits of the received fragments. Thus, if "ones" dominate in the i-th bits of a fragment, the i-th bit of the refined sequence is assigned a value of "one." In contrast, if they contain more "zeros," it is assigned a value of "zero."
- The Hamming distances for all syncword circular shifts are calculated for each refined sequence R_k. Here if one distance is less than or equal to $d_{lim} = \lfloor (d-1)/2 \rfloor$, the refined sequence is identified with the circular shift to which this distance corresponds.
- Synchronism is established if each of the sequences R_k, where $k \in [1, K]$, are identified by the same syncword circular shift; otherwise, the operations to detect the syncword are repeated beginning from the second stage in the current list.
- The number of accumulated fragments can be increased sequentially to a predetermined threshold $l_{\max}(1)$. Here, if synchronism has not been established after reaching this threshold, the search procedure is terminated, and a channel failure message is output by the system.

3. Evaluation

To evaluate the permutation-based frame synchronization method described in Subsection 2.10, a Python software model was developed to transmit information with data factorial coding. The simulation parameters are given in Table 6.

The algorithm for the model frame synchronization system is shown in Fig. 11.

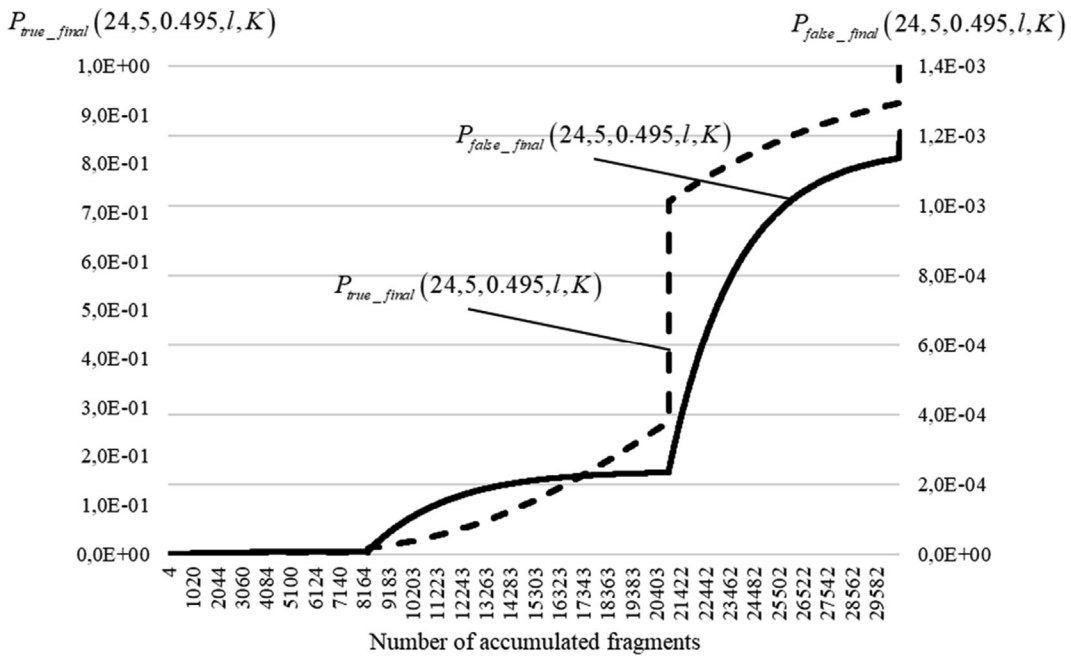

Fig. 10. Dependencies of estimates of probabilities of correct and false synchronization on the number of accumulated fragments for the adaptive synchronization process.

Table 6
Simulation parameters.

Communication channel	Binary symmetric
Bit errors	Independent
BER	$0.4 \leqslant p_0 < 0.498$
Number of tests	10^3, 10^4
Syncword	$\pi = (0,1,7,3,2,5,4,6) = (000,001,111,011,010,101,100,110)$
K	4
$l_{\min}(i)$ values	30549, 10457, 2779, 1
$l_{\max}(i)$ values	30603, 15273, 6971, 2083

In this evaluation, the permutation given in Table 2 was adopted as a syncword. Segment boundaries $l_{\min}(i)$, $l_{\max}(i)$ are defined above for $P_{true} \geqslant 0.9997$, $P_{false} \leqslant 3 \cdot 10^{-4}$ for any given value of bit error probability $p_0 \leqslant 0.495$. In addition, 10,000 tests were conducted to accurately determine the statistical indicators of frame synchronization.

4. Results and discussion

The graphs in Fig. 12 show the dependencies of the relative frequency $W_{true}(24,5,p_0,l,K)$ of establishing correct frame synchronism on the fixed value of accumulation factor l for BER $p_0 = 0.4$ and number of blocks $K \in \{1;2;3;4\}$. Here, each value of $W_{true}(24,5,p_0,l,K)$ was verified experimentally across 1000 tests.

The markers shown in Fig. 12 also illustrate the corresponding graphs of the theoretical dependencies P_{true} from l according to Eq. (9).

Note that the theoretical and experimental dependencies shown in Fig. 12 have the same distribution. In addition, the obtained p-values for the Pearson criterion interpreted according to the methodology described in the literature [40] are close to one. The same conclusions were made for the probability of false synchronization. This indicates the adequacy of the constructed model.

Fig. 13 shows the graph of the experimentally obtained CDF $W_{true\_final}(24,5,p_0,l,K) = \sum_{K^*l^* \leqslant Kl} W_{true}(24,5,p_0,l^*,K^*)$ on the number of processed fragments $L_{fr} = Kl$ for $p_0 = 0.495$ across 10,000 tests.

In Fig. 13, the dashed line shows the CDF lower estimate calculated by Equation (14), and the dash-dotted line represents the experimentally determined CDF for the previous method [36] at $p_0 = 0.495$.

Remark 8. No false synchronizations were observed in 10,000 tests during the experimental analysis of the developed method, for BER $p_0 = 0.495$.

The graphs in Fig. 13 confirm the relatively rough estimate obtained by Equation (14). The results also indicate that the proposed method allows reaching the set probabilistic indicators of frame synchronization faster than the previous method [36]. However, certain value ranges for the number of accumulated fragments were determined ($L_{fr} = 1 \ldots 14984$ in the given graph), where the relative frequency of correct synchronization for the previous method [36] is higher. In this case, the previous method [36] does not satisfy the requirements for the probability of false synchronism establishment. Note that there is a potential for combined application of both the previous and proposed methods; however, this is beyond the scope of the current study.

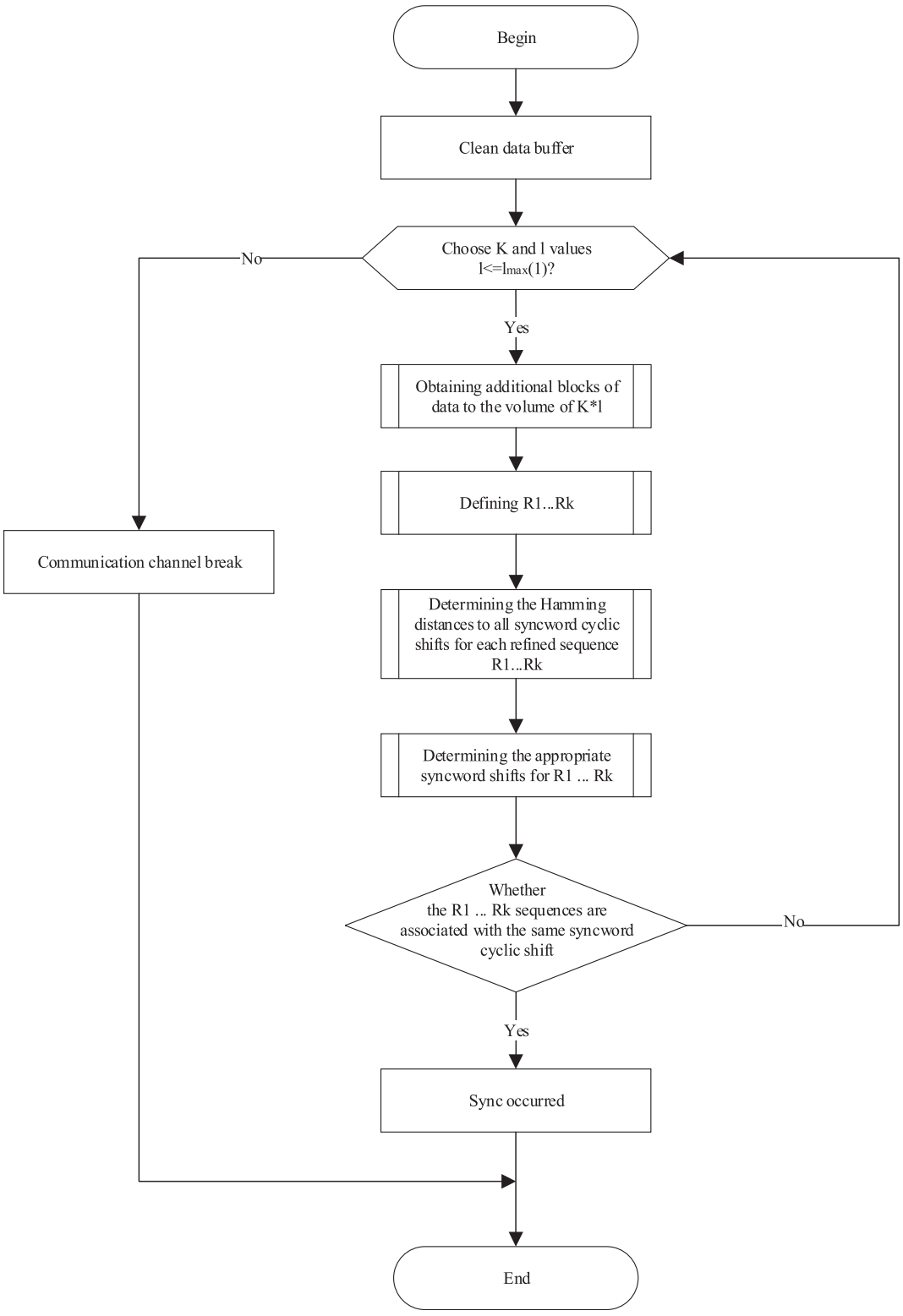

Fig. 11. Algorithm of the frame synchronization system.

It necessary to consider the behavior of the CDF $W_{true\_final}(24, 5, p_0, l, K)$ on $L_{fr} = Kl$ for the BER $p_0 \leqslant 0.498$ (Fig. 14). Here, the K and l values were determined for $p_0 = 0.495$, as shown in Table 5. For each value of BER, 1000 tests were conducted.

The graphs in Fig. 14 determine the probabilistic indicators to establish correct synchronism using the frame synchronization system at various values of BER p_0.

Fig. 14(b) shows that when the BER exceeds the limit value (i.e., $p_0 = 0.495$ in the example model), the requirements for correct synchronization probability may be violated. Thus, $W_{true\_final}(24, 5, 0.496, 30603, 1) = 0.997$, $W_{true\_final}(24, 5, 0.497, 30603, 1) = 0.920$, and $W_{true\_final}(24, 5, 0.498, 30603, 1) = 0.519$. In this case, the relative frequency of establishing false frame synchronism is $W_{false\_final}(24, 5, 0.496, 30603, 1) = 0.003$,

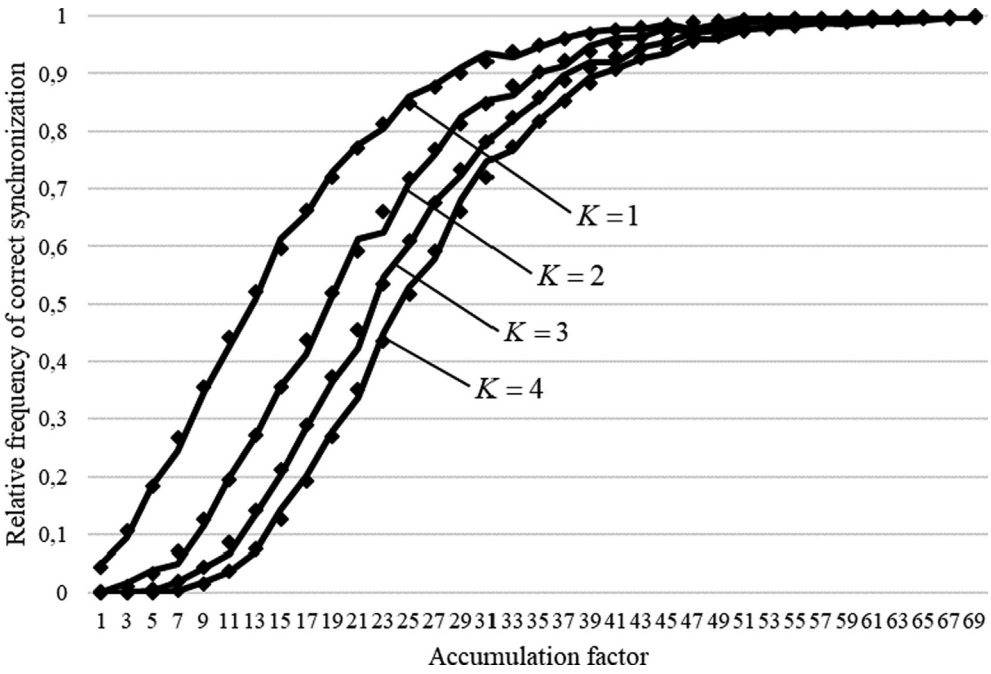

Fig. 12. Dependencies of $W_{true}(24, 5, p_0, l, K)$ on the fixed value of l.

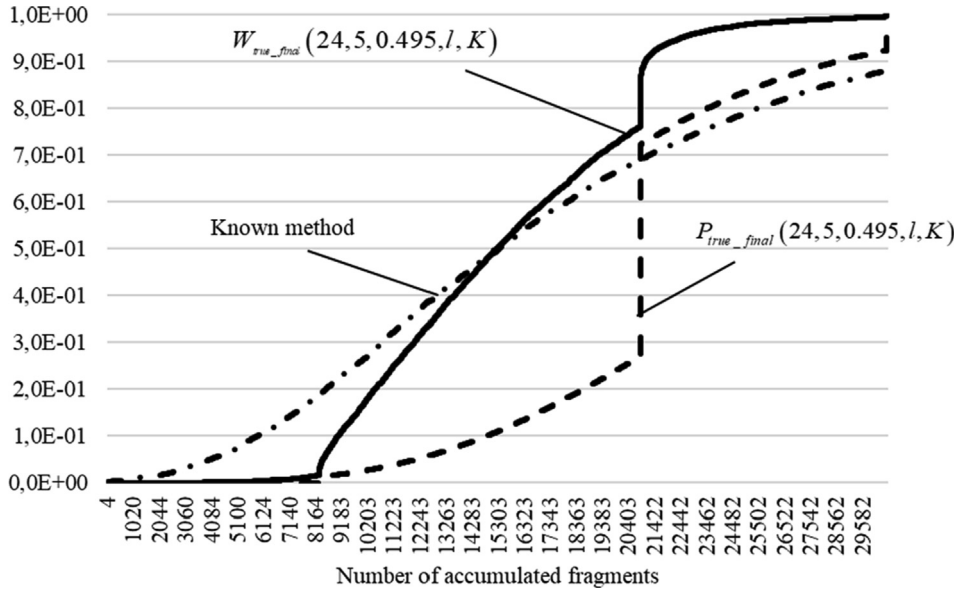

Fig. 13. CDF for BER $p_0 = 0.495$.

$W_{false\_final}(24, 5, 0.497, 30603, 1) = 0.001$, and $W_{false\_final}(24, 5, 0.498, 30603, 1) = 0.033$. In addition, the relative frequency of communication channel failure (a situation in which $L_{fr} = Kl = 30603$ synchronization was not found) is 0.003 for $p_0 = 0.496$, 0.080 for $p_0 = 0.497$, and 0.465 for $p_0 = 0.498$.

It is important to note that for maximum efficiency of the proposed method, bit error probability must be predicted as accurately as possible. However, the procedure used to select the moments to alter the K and l values can be adaptive, depends on bit error probability, and can be determined, for example, according to the average number of accumulated fragments before frame synchronism is realized.

At this stage, it is possible to define and demonstrate (Fig. 15) the dependency of the average value of the number of accumulated fragments prior to establishing frame synchronism $\bar{L}_{fr}$ on BER p_0.

Fig. 15 illustrates the exponential nature of the increase in the average value of the number of accumulated fragments prior to establishing frame synchronism (depending on the BER). This dependency can be employed for adaptive changes in the value change moments of the K and l value in the frame synchronization system.

Note that according to the permutation-based frame synchronization method described in Subsection 2.10, the refined sequences R_k, $k \in [1, K]$, are calculated independently. Each bit of

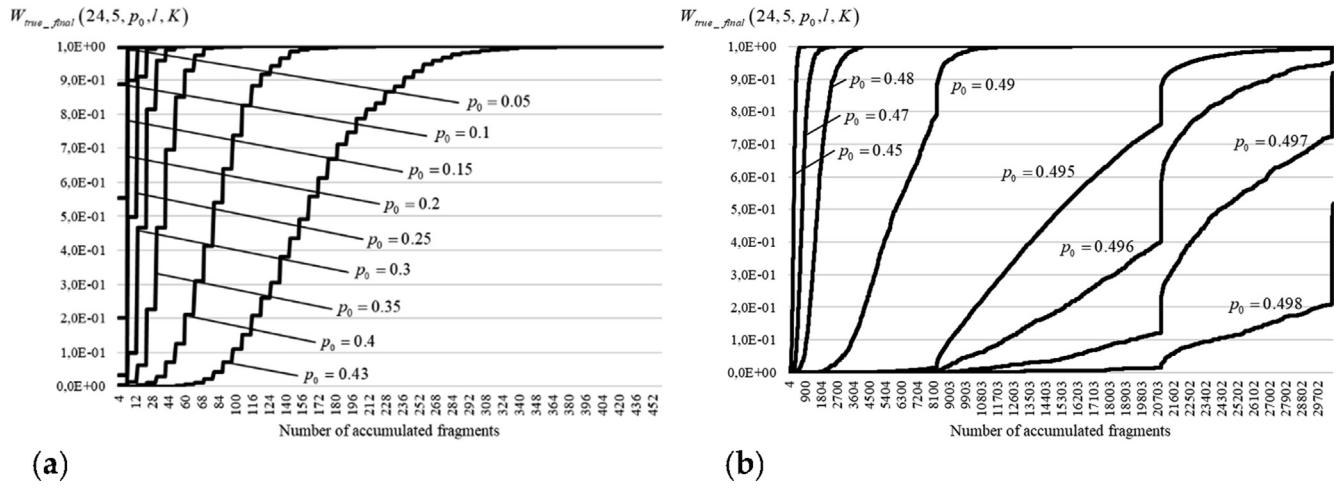

Fig. 14. CDF for BER: (**a**)$p_0 \leqslant 0.43$; (**b**)$0.45 \leqslant p_0 \leqslant 0.498$.

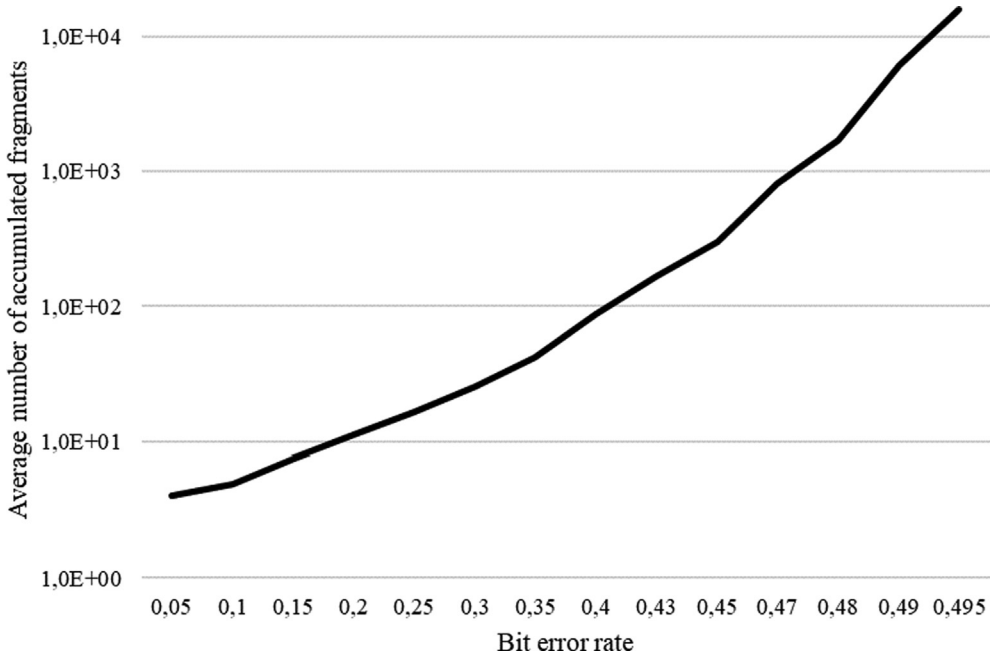

Fig. 15. Average number of accumulated fragments before establishing frame synchronism on BER.

these sequences is calculated on a majority basis based on the corresponding bits of the received fragments.

In the constructed model, for which the results are shown in Figs. 13–15, the fragments that define sequences R_k stay the same when the values of K and l are varied. When increasing l, new fragments just supplement the sets of fragments that define sequences R_k (for the constant K, each next value of l results in each set of fragments being supplemented with only two new fragments). Thus, the number of supplemented fragments is relatively small, e.g., 8 fragments for $l = 50$ and $K = 4$, what is only 4%. Thus, the result of the majority processing of received bits is highly correlated with the result obtained in the previous step for a smaller l value.

The following approach is proposed to reduce this correlation.

The frame synchronization system accumulates all the received fragments in one buffer and then randomly distributes the fragments equally between K sets of fragments that define sequences R_k. Here, if synchronization is not established in this step, the receiver also accepts new fragments and writes them to a general buffer. Then, the synchronization system again distributes the fragments randomly between the K sets of fragments. This approach to fragment processing is referred to as interleaving because the specified random distribution of fragments between K sets can be achieved by interleaving fragments in a common array in the buffer.

Fig. 16 shows the graph of the experimentally obtained CDF $W_{true\_final}(24,5,p_0,l,K)$ for the proposed processing approach with interleaving compared to the approach without interleaving. Here, all model parameters are the same as those in the previous example model, and the results are shown in Fig. 13.

The graphs in Fig. 16 show the effectiveness of interleaving fragments prior to calculating the R_k values. We found that the interleaving process reduces the required number of fragments received from the communication channel to establish synchro-

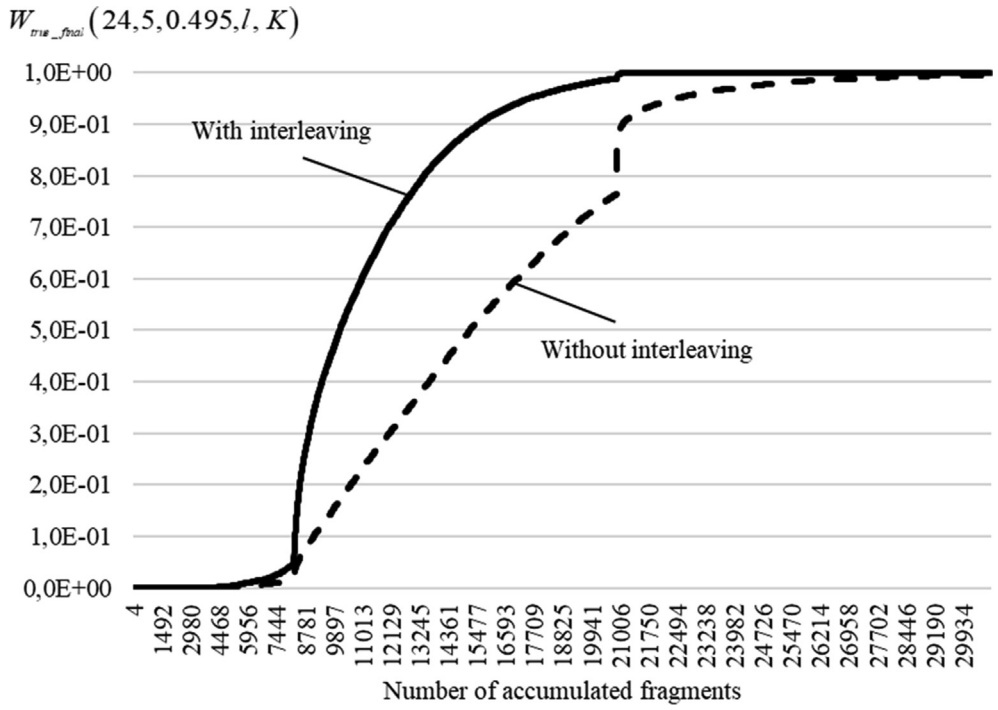

Fig. 16. CDF for BER $p_0 = 0.495$ (with interleaving).

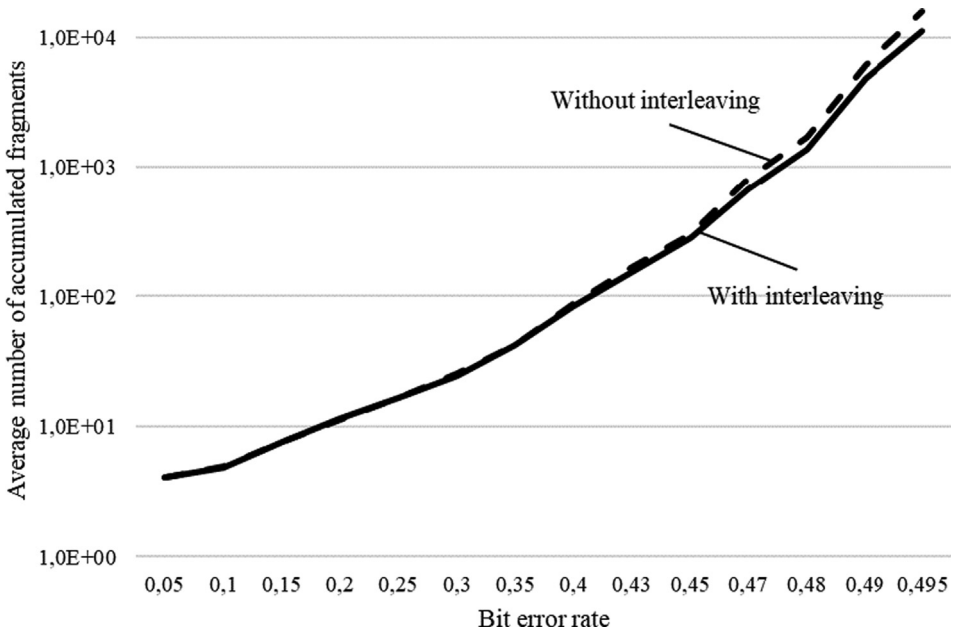

Fig. 17. Average number of accumulated fragments before establishing frame synchronism on BER (with interleaving).

nization. As a result, the time required to establish a communication session is reduced.

In Fig. 17, the dependencies of the average value of the number of accumulated fragments before establishing frame synchronism $\bar{L}_{fr}$ on BER p_0 are shown for the proposed method with interleaving compared to without interleaving.

The graph in Fig. 17 demonstrates the positive effect of interleaving. This effect becomes particularly noticeable with increasing BER. For example, for $p_0 = 0.495$, the average number of accumulated fragments obtained with the interleaving approach is reduced by 29.4%, i.e., 11,060 fragments compared to 15,667 fragments.

However, realizing this effect requires sufficient buffer memory resources to implement the proposed frame synchronization method with interleaving of accumulated fragments. The size of the buffer used to store the received fragments depends on the

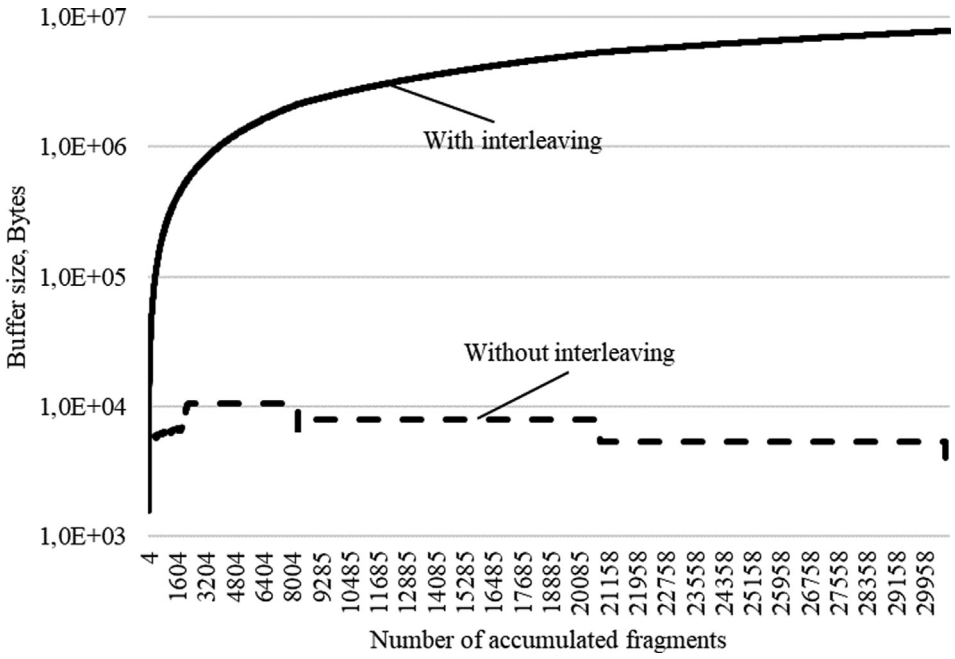

Fig. 18. Buffer size required to store increasing numbers of accumulated fragments.

number of fragments. For the implemented Python model, Fig. 18 shows the buffer sizes required for increasing numbers of accumulated fragments.

Thus, the procedure used to interleave accumulated fragments involves a tradeoff between memory costs and the time required to realize synchronization and establish a communication session.

5. Conclusions

The paper has presented a permutation-based frame synchronization method for data communications with short packets. In this method, the binary representation of permutation used as a syncword must satisfy the a single condition, i.e., the minimum Hamming distance to all its circular shifts is the maximum.

The results of this study suggest that, through the correlation processing and majority processing, the proposed frame synchronization method effectively realizes frame synchronization in data transferring systems under adverse noise conditions. The procedure for selecting the synchronization system parameters is implemented in a model data transmission system with the bit error probability $p_0 \leqslant 0.495$, the probability of correct synchronization $P_{true} \geqslant 0.9997$, and the probability of false synchronization $P_{false} \leqslant 3 \cdot 10^{-4}$.

We found that the proposed method reduces the amount of received data, which, in turn, reduces the time required to establish a connection. This leads to an increase the time to transfer user data.

In addition, we found that interleaving the accumulated fragments realizes further reduction of the required number of received fragments at the cost of increased memory usage on the receiver. For the given example with a BER of $p_0 = 0.495$, the average number of accumulated fragments was reduced from 15,667 to 11,060 fragments.

The proposed frame synchronization method can be implemented effectively in systems that use nonseparable factorial coding, but is not limited to these systems. A potential application of the permutation-based frame synchronization method is nonorthogonal multiple access (NOMA) [41–43], which is a promising radio access method in next-generation wireless communications, particularly in the Internet of Things context. The NOMA principles allow multiple users to be superimposed on the same resource. As a result, the noise immunity of such systems is reduced. Therefore, resource management and interference mitigation methods for ultradense networks, including the realization of reliable frame synchronization, require improved efficiency.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors are grateful to the Honorary Prof. of Cherkasy State Technological University, Dr. Valerii Shvydkyi for helpful remarks and discussion of the obtained results. This research was funded from the Ministry of Education and Science of Ukraine under grant 0120U102607. E.F. was supported by the nominal scholarship to young scientists - Doctors of Sciences for 2021, founded by the Verkhovna Rada of Ukraine, under grant 0121U112454.

References

[1] Faure E, Shcherba A, Stupka B. Permutation-based frame synchronisation method for short packet communication systems. In: Proc. 11th IEEE Intl. Conf. on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Cracow, Poland, 22-25 Sept.. p. 1073–7.
[2] Faure EV. Factorial coding with data recovery. Visnyk Cherkaskogo Derzhavnogo Tehnologichnogo Universitetu 2016;2:33–9.
[3] Faure EV. Factorial coding with error correction. Radio Electron Comput Sci Control 2017;3(3):130–8.
[4] Faure EV, Shcherba AI, Kharin AA. Factorial code with a given number of inversions. Radio Electron Comput Sci Control 2018;2(2):143–53.
[5] Faure EV, Shvydkyi VV, Lavdanskyi AO, Kharin OO. Methods of factorial coding of speech signals. Radio Electron Comput Sci Control 2019;4(4):186–98.
[6] Ling F. Synchronization in digital communication systems. Cambridge: Cambridge University Press; 2017.
[7] Massey J. Optimum frame synchronization. IEEE Trans Commun 1972;20(2):115–9.

[8] Scholtz R. Frame synchronization techniques. IEEE Trans Commun 1980;28 (8):1204–13.
[9] Hercog D. Communication protocols, principles, methods and specifications. Springer International Publishing; 2020.
[10] König H. Protocol engineering. Berlin, Heidelberg: Springer Berlin Heidelberg; 2012.
[11] Popovic M. Communication protocol engineering. 2nd ed. New York, USA: CRC Press; 2018.
[12] Kartaschoff P. Synchronization in digital communications networks. Proc IEEE 1991;79(7):1019–28.
[13] Hansheng W, Xiaoyi Q, Lieguang Z, Fuqin X. Coding, decoding, and recovery of clock synchronization in digital multiplexing system. IEEE Trans Commun 2003;51(5):825–31.
[14] Khlaponin Y, Khalifa EK, Khlaponin D, Selyukov A, Tolbatov A, Tolbatov V, et al. Method of improving the stability of network synchronization in multiservice macro networks. CEUR Workshop, Proceedings 2020;2654:786–97.
[15] Hamkins J. Frame synchronization without attached sync markers. Aerospace Conf 2011;2011:1–7.
[16] Bregman BF, Kvashennikov VV, Method of code frame synchronization of messages, R.F. Patent 2 459 366, 2006.
[17] Al-Azzeh JS, Ayyoub B, Faure E, Shvydkyi V, Kharin O, Lavdanskyi A. Telecommunication systems with multiple access based on data factorial coding. Int J Commun Antenna Propag 2020;10(2):102–13.
[18] Faure E, Shcherba A, Vasiliu Y, Fesenko A. Cryptographic key exchange method for data factorial coding. CEUR Workshop, Proceedings 2020;2654:643–64.
[19] Shcherba A, Faure E, Lavdanska O. Three-pass cryptographic protocol based on permutations. In: Proc. 2020 2nd IEEE Intl Conf. on Adv. Trends in Inf. Theor., Kyiv, Ukraine, 25 Nov.. p. 281–4.
[20] Lin S, Costello D. Error control coding. 2nd ed. NJ, USA: Prentice Hall; 2004.
[21] Kong JJ, Parhi KK. Interleaved convolutional code and its viterbi decoder architecture. EURASIP J Adv Signal Process 2003;13:2003.
[22] Bloessl B, Dressler F. mSync - Frames without preambles. Proceedings of the 21st ACM international conference on mobile computing and networking (MobiCom 2015)", 4th ACM Software Radio Imple-mentation Forum (SRIF 2015). Paris, France: ACM; 2015.
[23] Bloessl B, Dressler F. mSync: physical layer frame synchronization without preamble symbols. IEEE Trans Mob Comput 2018;17(10):2321–33.
[24] Durisi G, Koch T, Popovski P. Toward massive, ultrareliable, and low-latency wireless communication with short packets. Proc IEEE 2016;104(9):1711–26.
[25] Lee B, Park S, Love DJ, Ji H, Shim B. Packet structure and receiver design for low latency wireless communications with ultra-short packets. IEEE Trans Commun 2018;66(2):796–807.
[26] Bana A, Trillingsgaard KF, Popovski P, de Carvalho E. Short packet structure for ultra-reliable machine-type communication: tradeoff between detection and decoding. In: Proceedings of the IEEE Intl conf. on Acoust., Speech and Signal Process. (ICASSP), Calgary, AB, Canada, 15-20 April 2018. p. 6608–12.
[27] Zaman A, Hassan Z, Odarchenko R, Hassan S, Ahmed S, Bilal M, et al. Wireless underground sensor networks: packet size optimization survey. CEUR Workshop, Proceedings 2020;2616:353–65.
[28] Salamat Ullah S, Liew SC, Liva G, Wang T. Implementation of short-packet physical-layer network coding. IEEE transactions on mobile computing, 1–1, 2021.
[29] Salamat Ullah S, Liew SC, Liva G, Wang T. Short-Packet Physical-Layer Network Coding. IEEE Trans Commun 2020;68(2):737–51.
[30] Wu J, Kim W, Shim B. Pilot-less one-shot sparse coding for short packet-based machine-type communications. IEEE Trans Veh Technol 2020;69(8):9117–20.
[31] Östman J, Durisi G, Ström EG, Coşkun MC, Liva G. Short packets over block-memoryless fading channels: pilot-assisted or noncoherent transmission? IEEE Trans Commun 2019;67(2):1521–36.
[32] Liu X, Zhang X. Rate and energy efficiency improvements for 5G-based IoT with simultaneous transfer. IEEE Internet Things J 2019;6(4):5971–80.
[33] Nguyen ATP, Le Bidan R, Guilloud F. Trade-off between frame synchronization and channel decoding for short packets. IEEE Commun Lett 2019;23 (6):979–82.
[34] Nguyen ATP, Le Bidan R, Guilloud F. Superimposed frame synchronization optimization for finite blocklength regime. In: Proceedings of the IEEE wireless communications and networking conference workshop (WCNCW), Marrakech, Morocco, 15-18 April 2019. p. 1–6.
[35] Nguyen ATP, Guilloud F, Le Bidan R. On the optimization of resources for short frame synchronization. Ann Telecommun 2020;75(11–12):635–40.
[36] Faure EV, Shvydkyi VV, Shcherba AI, Kharin OO, Stupka BA. Method of cyclic synchronization based on permutations. Visnyk Cherkaskogo Derzhavnogo Tehnologichnogo Universitetu 2020;4:67–76.
[37] MacWilliams FJ, Sloane NJA. The theory of error-correcting codes. Amsterdam, Netherlands: North-Holland; 1977.
[38] Senatov VV. On the real accuracy of approximation in the central limit theorem. Siber Adv Math 2017;27(2):133–52.
[39] Deisenroth MP, Faisal AA, Ong CS. Mathematics for Machine Learning. Cambridge University Press; 2020.
[40] Rukhin A, Soto J, Nechvatal J, Smid M, Barker E, A statistical test suite for random and pseudorandom number generators for crypto-graphic applications, Spec," Pub, pp. 800-822 rev. 1a. Gaithersburg, MD: NIST, 2010.
[41] Abozariba R, Naeem MK, Patwary M, Seyedebrahimi M, Bull P, Aneiba A. NOMA-based resource allocation and mobility enhancement framework for IoT in next generation cellular networks. IEEE Access 2019;7:29158–72.
[42] Liu X, Zhang X. NOMA-based resource allocation for cluster-based cognitive industrial internet of things. IEEE Trans Ind Inf 2020;16(8):5379–88.
[43] Liu X, Zhai XB, Lu W, Wu C. QoS-guarantee resource allocation for multibeam satellite industrial internet of things with NOMA. IEEE Trans Ind Inf 2021;17 (3):2052–61.