Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 11 Issue 12, December - 2024

Noise-Tolerant Multimedia Data Encoding for Enhanced Network Performance

Ahmad AbdulQadir AlRababah
Faculty of Computing and Information Technology,
King Abdulaziz University, Rabigh, KSA.
Ahd_68@yahoo.com

***Abstract*— Noise-resistant encoding plays a critical role in improving the reliability of message transmission across computer networks by ensuring that transmitted data can be correctly reconstructed despite the presence of errors introduced by noise. However, traditional error-correcting codes, such as Hamming or Reed-Solomon codes, are generally designed to correct errors within packets and are not well-suited for dealing with the complete loss of packets, which is a common issue in IP networks, particularly when handling multimedia data streams.**

In IP networks, especially those used for multimedia transmissions like video and audio, packet loss is a significant challenge. Since multimedia data is typically transmitted in real-time, lost packets can result in severe degradation of the quality of the transmitted content. Traditional error-correcting mechanisms do not adequately address this issue, as they are primarily focused on correcting bit-level errors rather than packet-level losses. To address this gap, an analysis was conducted to explore the potential of erasure codes, which are specifically designed to recover missing packets in network transmissions. This makes erasure codes a more suitable option for multimedia applications in IP networks, where packet loss is more prevalent than bit-level errors.

Keywords*— *Multimedia; error-resilient coding; digital broadcasting; erasure codes; LT codes; non-random erasure codes.

I. INTRODUCTION

Multimedia data, like any other messages transmitted over computer networks, consist of files of finite size. These files are sent over the network as a sequence of packets, the size of which is determined by network standards and existing architectures [1,5]. Ideally, each packet should be received by the recipient without errors. Multimedia data, such as video, audio, and images, are transmitted over computer networks as finite-sized files [6,15]. These files cannot be sent in a single piece due to network constraints, so they are divided into smaller units called packets. Each packet contains a portion of the data, along with headers that include important metadata, such as the source and destination addresses, sequence numbers, and error-detection information. The size of these packets is determined by network standards and architectures, such as Ethernet frames or TCP/IP segments, which optimize for efficiency and compatibility across diverse devices and systems[2,4,18].

Ideally, every packet should arrive at the recipient without any errors, ensuring the integrity of the multimedia file. This is particularly important for multimedia applications where errors can cause noticeable artifacts, such as pixilation in videos, distortion in audio or incomplete image rendering. To achieve reliable transmission, error-detection and correction mechanisms are integrated into network protocols[3,10].These mechanisms include checksums, cyclic redundancy checks (CRC), and automatic retransmissions for packets detected as corrupted. Despite these safeguards, factors like network congestion, interference, and hardware failures can occasionally lead to packet loss or errors, which need to be mitigated to maintain the quality of multimedia transmissions[7]. However, some packets may be delivered with erroneous bits, while others may be lost in the network. In the theory of error-correcting codes, such packets are considered erased, and the basic mathematical model used to describe a noisy channel in computer networks is the erasure channel model for packets [13]. To detect errors in the packets, the sender employs coding; in error correcting code theory, when packets are lost or arrive with uncorrectable errors, they are considered "erased." To handle this, a mathematical model called the erasure channel is used, which describes how packets can be lost or corrupted during transmission. To detect and correct these errors or losses, the sender adds extra information to the data using coding techniques, which helps the receiver identify and recover the original data[9,21].

For any code, there is a probability that an error will go undetected. An error will not be detected if the transmitted code word is transformed into another code word in the channel. This transformation is referred to as the transformation of code words. In the erasure channel model, the probability of a packet being mistakenly accepted due to code word transformation is neglected [8,20]. For most network applications (IP broadcasting, IP multicast service, simultaneous data transmission from multiple sites on the Internet, etc.), this neglect is justified due to the extremely low probability of transformation. Each network application has its own specifics, and in this article, IP broadcasting is considered as the network application [11,16]. In communication systems utilizing error-detection codes, an undetected error can occur if a transmitted code word is transformed by the channel into another valid code word, a phenomenon called the transformation of code words [14]. This issue arises because error-detection codes rely on the assumption that any alteration to a code word results in a pattern outside the set of valid code words. In practice, the probability of such undetected errors is exceedingly low, particularly in the context of an erasure channel model, where errors typically manifest as detectable packet losses or erasures rather than transformations[17].

For most network applications, such as IP broadcasting, IP multicast services, or simultaneous data transmission across multiple sites, this negligible probability is considered insignificant and is therefore often ignored in error modeling. This simplification is practical because modern error-detection and correction mechanisms are highly effective, ensuring that the likelihood of undetected transformations is minimal and does not compromise system reliability. In IP broadcasting, specifically, where data is simultaneously transmitted to multiple recipients, this assumption holds strong due to the robust design of coding schemes and the inherently low risk of undetectable transformations impacting overall data integrity[3,19].

New data delivery methods and architectures of broadcasting systems over IP networks (hereinafter briefly referred to as IPTV, Internet Protocol Television) have introduced fundamentally new challenges. One of the most pressing and relevant issues is the problem of packet loss in real-time IPTV systems, or 'live' broadcasting. Data loss leads to degradation in the quality of television programs, including loss of sound, the breaking up of images into annoying 'blocks,' and desynchronization of audio and video[2,23]. The core issue is that 'live' broadcasting requires the use of IP protocols with no guaranteed packet delivery, such as UDP (User Datagram Protocol) and RTP (Real-Time Protocol), since retransmission of lost packets (known as the ARQ, Automatic Repeat request scheme) is not feasible due to the significant delays this operation would introduce, which are unacceptable in 'live' television broadcasting. Furthermore, retransmission requests, especially in a 'one-to-many' topology, would place additional loads on the communication channel and could potentially lead to further degradation of the service. Packet loss in unreliable protocols occurs due to collisions, network delays, bottlenecks in bandwidth, and other similar issues [8, 22].

Let us note that classical error-correcting codes are unable to solve this problem. Convolutional codes correct individual bit errors. Block codes, including Reed-Solomon codes, can correct bursts of errors within a single packet. However, in the case of the complete loss of a packet, they are ineffective[11].

Currently, we can discuss the emergence of a new class of error-correcting codes designed for erasure channels erasure codes. These codes can encode a message of finite size into a potentially unlimited stream of independent packets [15]. This characteristic fundamentally differentiates this new class of codes from classical block or convolutional error-correcting codes that operate at a fixed rate.

For codes within this new class, the term 'rateless' has been introduced, indicating that they do not have a predetermined encoding or decoding rate. These codes are also referred to as digital fountain codes due to their ability to produce an endless supply of encoded packets that can be drawn from a finite message, similar to water flowing from a fountain[19].

Historically, the first and most conceptually significant rateless code is the LT code, created by Michael Luby in 2002. The code derives its name from the 'Luby Transform' (LT), which refers to the specific mathematical transformation used in its encoding process[20].

The most significant distinction between erasure codes and block or convolutional codes is the ability to recover an entire packet in the event of its loss. Currently, erasure codes have already found applications in commercial products for computer networks, supplied, for example, by the company Digital Fountain. The statement emphasizes the unique strength of erasure codes in recovering lost data packets compared to traditional error-correcting codes[4,16]. This capability makes erasure codes particularly valuable in commercial networking applications, where packet loss can impact performance and user experience. Analysis of the existing sources has shown that there are only isolated attempts to apply erasure codes in IPTV. It is not yet feasible to speak of a more or less comprehensive IPTV technology using erasure codes. One can only highlight the mobile television standard DVB-H (Digital Video Broadcasting - Handheld), which involves the use of Raptor erasure codes, developed based on LT codes with consideration of the specifics of mobile broadcasting[4,23].

This article is dedicated to the analysis of the possibility of applying erasure codes in IPTV systems. The article proposes new error-correcting erasure codes to reduce multimedia data loss in IP broadcasting.

Journal of Multidisciplinary Engineering Science and Technology (JMEST)
ISSN: 2458-9403
Vol. 11 Issue 12, December - 2024

II. PRINCIPLES OF CONSTRUCTING ERASURES CODES: LT CODES

Assume that we have an original message consisting of *K* original symbols or packets. The lengths of all symbols are equivalent and equal to *L*. The code symbols in erasure codes are generated as a result of the "exclusive or" (XOR) operation over d original symbols. The value of d is referred to as the degree of the code symbol. This value can take on values from 1 to some maximum value *G*. The original symbols used to generate a certain code symbol *i* will be referred to as neighbors. We also introduce the probability distribution of degrees *p(d): p(d)* is the probability that a code symbol will have degree *d*. The degree distribution can be specified both analytically and in a tabular form as a set of values *p*(1), *p*(2),...,*p*(*G*−1), *p*(*G*).

The following is a description of the process for generating code symbols for LT codes according to our translation:

- Randomly select the degree *d* of the code symbol using the probability distribution of degrees *p*(*d*);
- Randomly choose *d* different original symbols as the neighbors of the code symbol;
- Set the value of the code symbol equal to the result of the XOR operation on the *d* selected neighbors.

The question arises: how many code symbols need to be generated to recover the original message? Since the code symbols are independent of each other, the generation process can potentially continue indefinitely until all users receive a sufficient number of code symbols to fully recover the original message. This is the "fountain" property of LT codes. An example of the generating graph of an LT code is presented in Figure 1.

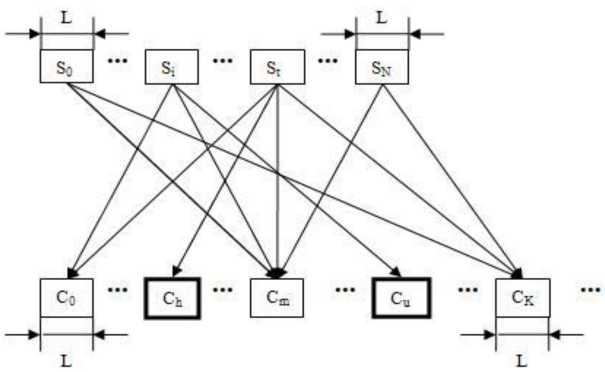

Fig. 1. *Example of a generating graph of an LT code*

Let to present the decoding algorithm for LT codes in our interpretation. Denote the queue of all code symbols received by the receiver as OKS. We will introduce the concept of OOS, which is the queue of processed symbols whose degree is equal to 1 at a certain iteration of decoding. The queue of decoded, or more correctly, restored symbols will be referred to as OVS.

Initially, the OVS is empty. Among the OKS, we find all symbols with a degree of 1 and move them to the OOS. We then iteratively process each i-th symbol in the OOS (OOSi_), performing the following actions:

• It is restored the original symbol, which is the only "neighbor" for OOSi in the corresponding symbol in OVS, i.e., we copy the contents of the OVSi symbol into the corresponding original symbol OVSk ;

• Found among the code symbols in the OOS the symbols in which the i-th symbol is included as a neighbor. We perform the XOR operation on the contents of the found symbol in OKS (denote it as OKSj) and the current i-th symbol, reducing the degree of OKSj by 1.

• If the degree of OKSj becomes equal to 1, and such a symbol is not present in either OBC or OOC, we move OKSj to the OOS.

• OOSi removed from the OOS.

The process is considered successfully completed if all original symbols are recovered. The algorithm terminates unsuccessfully if, at any step, the queue of processed symbols (OOS) becomes empty. In this case, some original symbols will be lost. To prevent an unsuccessful situation, it is essential to properly design the distribution of the degrees of code symbols. This is a critical task in the design of erasure codes. For LT codes, M. Luby proposed the Robust Soliton Distribution. The robust distribution μ(d) is defined by the formula:

$$\mu(d) = \frac{p(d)+\tau(d)}{\sum_d p(d)+\tau(d)},$$

where,

$$p(d) = \begin{cases} p(1) = 1/K, \\ \dfrac{1}{d(d-1)}, \quad d = 2,\dots,K, \end{cases}$$

$$\tau(d) = \begin{cases} \dfrac{S}{K}\dfrac{1}{d}, \ d = 1,2,\dots,(K/S)-1 \\ \dfrac{S}{K}\log(S/\delta), \ d = K/S \\ 0, \ \ d > K/S \end{cases}.$$

In the last formula, S is defined by the expression:

$$S(K,\delta) = c\ln(K/\delta)\sqrt{K}.$$

The parameters of the robust distribution are: K – the number of original symbols; c – the distribution parameter; defined as a positive number, it is recommended to set 0<c<1; the decoding process will be successfully completed with a probability of 1−δ. An example of the robust distribution is shown in Fig. 2.

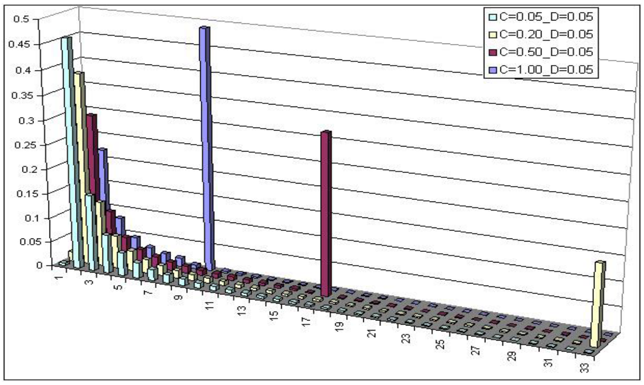

Fig. 2. *Robust distribution of the degrees of code symbols μ(d). Distribution parameters: K=10000, c=0,05...1,0, δ=0,05*

III. ANALYSIS OF THE POSSIBILITY OF USING LT CODES IN IPTV

Let's formulate the main properties of LT codes.

- Code symbols are generated independently of each other; this is the so-called "fountain" property of LT codes- any received code symbols can be used by the receiver to recover the original message.
- Original symbols have equal priority.
- The length of the original symbols can be arbitrary, but all original symbols have equal lengths. The lengths of the code symbols are also equal to the length of the original symbols.
- Due to the fact that the theory of LT codes is based on the statistical problem of balls and bins, the effectiveness of LT codes becomes apparent at sufficiently large values of the parameter K i.e., with a large number of original symbols. The inventors of LT codes recommend values of K on the order of 10,000.

The listed properties of LT codes are highly valuable and allow for the practical use of LT codes for any type of transmitted data. An important feature is the fountain property of the codes, especially for the "many-to-many" problem. However, the mentioned properties of erasure codes are insufficient for successfully addressing the tasks of transmitting digital multimedia data, particularly for online broadcasting systems.

Let's outline the main problems associated with the application of LT codes in IPTV systems.

- Unacceptably Long Delay: The delay of the television signal between the receiver and transmitter becomes unacceptable due to the need to buffer a large number of source packets when dealing with values of K on the order of 10,000. While this is not an issue in delayed media file uploads, it results in delays of several tens of seconds in real-time mode.

- Equal Priority of Source Packets: The equal priority of source packets is not the best solution for IPTV systems, as packets containing video frame data of various types and audio data contribute differently to the overall quality perception of television programs. Losses in audio data pose the greatest problem, while the loss of key I-frames leads to significant degradation in video quality. Multimedia data, such as video and audio files, are transmitted over computer networks as finite-sized files that are divided into packets for efficient delivery. These packets, sized according to network standards and architectures like Ethernet or TCP/IP, contain not only the data payload but also essential metadata, such as error-detection codes and sequence numbers. In an ideal scenario, every packet would reach the recipient without errors to preserve the integrity of the multimedia content. This is critical for applications like IPTV, where any packet loss can degrade the viewing experience.

In IPTV systems, treating all packets with equal priority is not optimal because different types of data packets contribute differently to the overall quality perception. For instance, packets carrying audio data are crucial, as losses in audio can significantly disrupt the viewer's experience, making the content unintelligible, similarly, packets containing key I-frames (intra-coded frames), which serve as reference points for decoding subsequent video frames, are vital for maintaining video quality. The loss of I-frame packets can lead to severe video degradation, causing visible glitches or long interruptions in playback. On the other hand, the loss of packets from less critical P-frames or B-frames (predictive and bidirectional predicted frames) has a less noticeable impact. Therefore, prioritizing packets based on their importance, such as giving higher priority to audio and I-frame packets can enhance the overall quality and reliability of IPTV systems, especially in environments prone to network congestion or packet loss.

- The property of fountain-like behavior does not play a significant role in online IPTV systems, as broadcasting in such systems is subject to strict timing requirements for receiving, decoding, and reproducing frames, and there is no possibility of waiting for subsequent (coded packets). In online IPTV systems, the concept of fountain-like behavior—where data can be received and reconstructed in any order without strict sequencing—has limited relevance. This is because IPTV systems operate under strict real-time constraints for the delivery, decoding, and playback of video and audio frames. In such systems, frames must be received in a precise sequence and within tight timing windows to ensure smooth and uninterrupted playback. Unlike file transfer protocols, where data packets can be reassembled later without strict timing, IPTV cannot afford delays or out-of-order processing since it would result in noticeable playback interruptions or quality degradation. As a result, the flexibility of fountain-like protocols is not applicable, as IPTV requires a continuous and synchronized stream of packets with no allowance for waiting on missing or delayed data.

Nevertheless, the property of fountain-like behavior can be utilized to relieve the channels of broadcasting head servers, as in the case of using erasure codes, any node in the broadcasting network can operate simultaneously in both the mode of receiving coded symbols and their immediate retransmission to nearby nodes. The property of fountain-like behavior can be leveraged in broadcasting networks, such as those used for IPTV, to improve efficiency and reduce the load on central broadcasting servers. This behavior is particularly useful when combined with erasure codes, which allow data to be encoded into multiple redundant symbols. In this approach, any node in the broadcasting network such as a router or a peer device can simultaneously perform two functions: receiving the encoded symbols and immediately retransmitting them to other nearby nodes.

This capability distributes the transmission workload across the network, reducing the reliance on the head servers that originate the broadcast. Instead of every node needing to connect directly to the central server to receive packets, nodes can share the responsibility of distributing data among themselves. As a result, the central server's bandwidth requirements are significantly reduced, and the system becomes more scalable and resilient to congestion or bottlenecks. This method is particularly advantageous in large-scale broadcasting networks, where the volume of data and the number of recipients can put substantial pressure on the central servers.

Erasure codes offer a fundamental advantage in IPTV systems due to their ability to recover lost packets, making them highly promising for applications where data loss or packet erasure is a concern. This property allows IPTV systems to maintain the integrity of the transmitted multimedia content, even when packets are lost due to network issues or congestion. Erasure codes work by encoding the original data into additional redundant packets, which can be used to recover lost or missing information. However, while the potential benefits are clear, several challenges specific to IPTV systems need to be addressed for these codes to be effectively applied. These include ensuring low latency and real-time decoding, as IPTV systems require strict timing for seamless playback. Moreover, the system must be able to prioritize critical data, such as audio and key video frames, to minimize the perceptible degradation in quality. Addressing these application-specific issues will be crucial in fully realizing the advantages of erasure codes in IPTV, ensuring both reliability and high-quality user experience under varying network conditions..

IV. NON-RANDOM ERASURE CODES

To address the aforementioned problems, new erasure codes were proposed, called "non-random erasure codes" (NREC). The motivation for creating NREC was that at low values of K, the statistical properties that form the basis of the LT code theory do not function. Therefore, it is necessary to control the number of occurrences of source symbols in code packets, ensuring greater robustness and increasing the efficiency of the code.

The NREC code utilizes a modified robust distribution (Robust Soliton Distribution μ'(d). The robustness of the modified distribution is further enhanced by increasing the number of packets with degree 3 by 30% of the number of packets with degree 1 (derived empirically). In the current implementation of NREC, the number of code symbols is limited to a specific value N, determined based on the maximum allowable surplus of code packets relative to the number of K source symbols as N=Kβ, where β is a value slightly greater than 1 (set based on the characteristics of the transmission channel).

It will be denoted the queue of source symbols as OIS, as before. The queue of coded symbols will be denoted as OCS. We will use indices to refer to symbols in the OIS and OCS. The number of occurrences of a source symbol f in code symbols as a "neighbor" will be called the measure of protection of the source packet. We will denote the measure of protection with a variable R[f], f=0,...,K–1..

The algorithm for encoding NSSC (Non-Random Erasure Codes) is provided below.

Step 1. Determine the number of code symbols intended for encoding with degrees $d_0,...,d_{max}$ according to:

$$n[d_i] = \mu\,(d_i) \cdot N,$$

Where N is the target number of code symbols.

Step 2. Select $n[d_0]$ source packets with the highest priority. Encode them as reference symbols with degree 1. This action can be represented in pseudo-C code as:

```
for(i=0; i < n[d0]; i++)
{
OCS [i] = OIS [fi];
},
```

Where, *fi* denotes the index of the source symbol from the OIS (Original Input Symbols) queue.

Step 3. Perform the encoding of all symbols encoded in Step 1 according to the following procedure:

```
for(i =0; i < n[d0]/3; i++)
{
```

OCS[i+n[d0]]=OIS[f3i] XOR OIS[f3i+1] XOR OIS[f3i+2]; R[f3i]=2; R[f3i+1]=2; R[f3i+2]=2;
}

Step 4. Let W_R represent the subset of all original symbols that, at a given encoding step, have a protection measure of R. Among them, define the subset of symbols with the maximum protection measure as W_{MAX}, and the subset with the minimum protection measure as W_{MIN}.
Set $i = 2$.
Step 4.1. Perform actions 4.1.1 and 4.1.2 $n[di]$ times:

Step 4.1.1. Generate the next k-th code symbol as the result of the XOR operation over one symbol from W_{MAX}, one from $W_{\text{MAX}}{-}1$, ..., one from W_{L-1}, and one symbol from W_{MIN}, where $L = MAX - di$, i.e., perform the following action:

$$OCS[k]=W_{MAX}[j_0]\ XOR\ W_{MAX-1}[j_1]\ XOR...\ XOR\ W_{L-1}[jL-1]\ XOR\ W_{MIN}[j_L];$$

Step 4.1.2. Increment by one the values of $\mathrm{R}[j_t]$ for all j_t, where $t = 0...L$ (here, the indices jt represent the indices of the original packets in OIS).

Step 4.2. Update the subsets W_R, as well as the maximum and minimum values of R. Increment i by one. If i equals $d_{max}+1$, or the total number of coded symbols has reached the desired number N, exit. Otherwise, proceed to Step 4.1.

It is important to note that the higher the priority of an original packet selected from WMIN, the lower the degree of the coded packet in which it will first be involved in encoding.
The most important aspect is that the rows of the matrix of original packet indices J should ideally be linearly independent, or at the very least, have as few combinations of linearly dependent vector rows as possible.

$$J=\begin{bmatrix} j_0^1 & \dots & j_{d_2-1}^1 & 0 & \dots & 0 \\ j_0^2 & \dots & \dots & j_{d_3}^2 & 0 & 0 \\ & & & & & \\ \dots & \dots & \dots & \dots & \dots & \dots \\ j_0^{N-n[d_0]} & \dots & \dots & \dots & \dots & j_{d_{\max}}^{N-n[d_0]} \end{bmatrix}.$$

The generator graph of the NSSC code is a graph with a systematic structure. An example of such a graph is shown in Fig. 3.

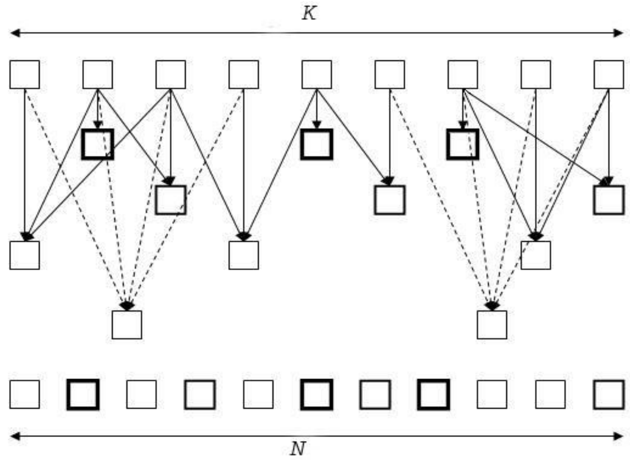

Fig. 3. *Example of the generator graph of the NSSC code.*

V. CONCLUSION
The article explores the potential use of erasure codes in IPTV systems. It was found that erasure codes are capable of recovering data in the event of packet loss. However, for IPTV systems, erasure codes need to have specific properties to be effective. The codes we developed, called Non-Random Erasure Codes (NSSC), possess these required properties. It was both theoretically and experimentally demonstrated that NSSC codes show higher efficiency compared to LT codes when the value of K (the number of original symbols) is around 100 to 1000.
This detailed conclusion underscores the critical need for specialized erasure codes tailored to the unique requirements of IPTV systems. Unlike traditional data transmission scenarios, IPTV involves real-time delivery of multimedia content with strict timing and quality constraints. Specialized erasure codes play a pivotal role in ensuring reliability and minimizing the impact of packet loss during transmission. Among these, the use of Non-Systematic Systematic Codes (NSSC) stands out, particularly for scenarios involving smaller values of K (the number of original data packets). NSSC offers significant advantages by optimizing the encoding and decoding processes to meet the stringent timing demands of IPTV systems. By efficiently managing redundancy and enabling rapid recovery of lost packets, NSSC enhances both the quality of service and the viewer experience, ensuring that critical packets, such as those carrying audio data or key video frames, are reliably delivered with minimal delay. This makes NSSC an invaluable tool for addressing the challenges inherent in modern IPTV broadcasting.

References

[1] Luby M. LT Codes // Proc. of the 43rd Annual IEEE Symp. on Foundations of Computer Science (FOCS). – 2002. – P. 271–282.

[2] Liu, Yunan, et al. "Hierarchical information passing based noise-tolerant hybrid learning for semi-supervised human parsing." Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 35. No. 3. 2021.

[3] Ahmad AlRababah, "Implementation of Software Systems Packages in Visual Internal Structures" Journal of Theoretical and Applied Information Technology, Volume 95, Issue 19 (2017), Pages: 5237-5244.

[4] Chintalapati, Shireesha, and M.V. Raghunadh. "Automated attendance management system based on face recognition algorithms." International Conference on Computational Intelligence and Computing Research. IEEE, 2015.

[5] Jha, Abhishek. "Classroom attendance system using facial recognition system." The International Journal of Mathematics, Science, Technology and Management, 2014.

[6] Riya, G. Lakshmi, et al. "Implementation of attendance management system using SMART-FR." International Journal of Advance Research Computer and Communication Engineering, 2015.

[7] Akash G, Rupali B and Shobhana S. "SDLC (Software Development Life Cycle)". Published 2014.https://www.slideshare.net/akash250690/sdlc-models-38873234.

[8] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[9] M. Yousef, K. F. Hussain, and U. S. Mohammed, “Accurate, data-efficient, unconstrained text recognition with convolutional neural net-works,” Pattern Recognition, vol. 108, p. 107482, 2020.

[10] Kharati, Ehsan, Mohammad Khalily-Dermany, and Hamidreza Karmajani. "Increasing the amount of collected data using network coding and continuous movement of mobile sinks in wireless sensor networks." IET Networks 9.3 (2020): 128-138.

[11] Israa Al-Barazanchi, Aparna Murthy, Ahmad Abdul Qadir Al Rababah, Ghadeer Khader, Haider Rasheed Abdulshaheed, Hafiz Tayyab Rauf, Elika Daghighi, Yitong Niu. "Blockchain Technology - Based Solutions for IOT Security" IJCSM : Iraqi Journal for Computer Science and Mathematics, vol. 3, no. 1, Jan. 2022.

[12] Hussein, Adil Yousef, Paolo Falcarin, and Ahmed T. Sadiq. "Enhancement performance of random forest algorithm via one hot encoding for IoT IDS." Periodicals of Engineering and Natural Sciences 9.3 (2021): 579-591.

[13] Ahmad AlRababah "Assurance Quality and Efficiency in Corporate Information Systems", IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.4, April 2019.

[14] Shafique, Arslan. "A noise-tolerant cryptosystem based on the decomposition of bit-planes and the analysis of chaotic gauss iterated map." Neural Computing and Applications 34.19 (2022): 16805-16828.

[15] Antoniou, Andreas. Digital filters: analysis, design, and signal processing applications. McGraw-Hill Education. Published 2018

[16] Ahmad AlRababah, Ahmad Alzahrani. "Software Maintenance Model through the Development Distinct Stages", IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.2, February 2019.

[17] Chen, Ziyun, et al. "A noise rate estimation method for image classification with label noise." Journal of Physics: Conference Series. Vol. 2433. No. 1. IOP Publishing, 2023.

[18] Ahmad AlRababah, Bandar Ali Alghamdi. "Information Protection Method in Distributed Computer Networks Based on Routing Algorithms" IJCSNS International Journal of Computer Science and Network Security, VOL.19 No.2, February 2019.

[19] Savva, Giannis, Konstantinos Manousakis, and Georgios Ellinas. "Network coding-based routing and spectrum allocation in elastic optical networks for enhanced physical layer security." Photonic Network Communications 40.3 (2020): 160-174.

[20] Li, Yumei, Futai Zhang, and Xin Liu. "Secure data delivery with identity-based linearly homomorphic network coding signature scheme in IoT." IEEE Transactions on Services Computing 15.4 (2020): 2202-2212.

[21] Ahmad AlRababah. "Data Flows Management and Control in Computer Networks", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 9, No. 11, 2018.

[22] Aziz, Haris, et al. "A Noise-Tolerant Audio Encryption Framework Designed by the Application of S8 Symmetric Group and Chaotic Systems." Mathematical Problems in Engineering 2021.1 (2021): 5554707.

[23] Yang, Xudong, et al. "A coarse to accurate noise-tolerant positioning evaluation for mobile target based on modified genetic algorithm." Ad Hoc Networks 143 (2023): 103123.