IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 14 March 2023, accepted 26 April 2023, date of publication 1 May 2023, date of current version 24 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3271889

RESEARCH ARTICLE

Fulcrum Rateless Multicast Distributed Coding Design

YAIR RIVERA JULIO [1], **ISMAEL GUTIÉRREZ GARCÍA** [2], **AND JOSÉ MÁRQUEZ DÍAZ** [1]

[1]Department of Computer Science and Engineering, Universidad del Norte, Barranquilla 01800, Colombia
[2]Department of Mathematics, Universidad del Norte, Barranquilla 01800, Colombia

Corresponding author: Yair Rivera Julio (yairr@uninorte.edu.co)

This work was supported by the Advance in Doctoral Research (The design of next-generation distributed codes and a balance in the performance of multicast networks).

ABSTRACT Establishing reliable and efficient multicast services in heterogeneous networks under an erasure channel is one of the current priorities in coding theory, particularly in Network Coding (NC) over data networks. Additionally, the increasing number of clients with mobile devices with high processing capabilities and the prevalence of non-delay tolerant traffic have led to the demand for non-feedback multicast schemes regarding distributed resource management. Current communication platforms lack gradual and dynamic coding control based on the type of data being transmitted. This paper proposes a reliable transmission scheme based on a concatenation of systematic coding and Random Linear Network Coding (RLNC) called Fulcrum coding. This hybrid rateless coding scheme with distributed characteristics allows the implementation of an adaptive resource management system to increase the decoding probability during data reception. The method ultimately results in a higher network throughput and shorter transmission times (RTT) per packet by implementing efficient Forward Error Correction (FEC).

INDEX TERMS Fulcrum coding, random linear network coding (RLNC), network coding (NC), multicast, rateless coding, coding theory, round-trip time (RTT), forward error correction (FEC), heterogeneous networks, erasure channel.

I. INTRODUCTION

The excessive demand for services implies high throughput of digital data traffic in multicast mobile wireless networks. These services increasingly require data frame coding to ensure efficient delivery in hostile transmission environments [1]. The wireless channel is a stochastic medium in which many phenomena directly affect the transmission quality, including poor link quality, interference, noise, signal fading, and power supply. These factors directly affect the life cycle, reliability, and quality of services offered by mobile networks [2], [3]. Owing to this problem, a variety of solutions have been implemented based on protocols residing in various layers of the Open Systems Interconnection (OSI) model, some in the application layer and others in the transport layer, to ensure the delivery of data, in compliance with some Quality of Service (QoS) policies [4]. However, these protocols have some shortcomings, mainly because they do not differentiate between failures due to network congestion, consequences of bottlenecks, or modifications of the wireless medium, drawbacks that do not consider the type of data coding when sending high-performance streaming [5]. Thus, advances based on code theory have been designed, especially in modern NC coding schemes and in derived codes such as the Fulcrum Code, which aims to break the traditional stored and forward (SF) schemes and establish a new transmission unit format or protocol data unit (PDU) for data coding at each point of transmission [6], [7], [8]. The development of this next-generation code has been extensively implemented in digital data transfer, as it enables mapping of data segments to an n-dimensional vector space over a finite field, such as $GF(2^n)$, with $n \geq 1$ and supported with linear codes to implement forward error correction FEC strategies, including the general class of erasure codes [9].

The associate editor coordinating the review of this manuscript and approving it for publication was Jintao Wang.

Implementing these techniques in binary erasure channels is efficient in recovering valid information when it is affected by errors or losses in any transmission medium [10]. The inherent nature of digital communication channels implies unreliable links between each terminal, resulting in a wide range of variants in different network connections. These processes affect the system performance because of the increased randomness of the medium caused by signal fading, broadcast blockages, and congestion due to transmission on best-effort channels [11].

II. RELATED TECHNIQUES

Currently, complements to RLNC coding have been developed to create data redundancy while encompassing gradual decoding of the system, Fulcrum coding being an example of this.

Fulcrum coding is a concatenation of two independent coding systems: incoming inner code and outgoing outer code [12], [13], [14]. One part is in charge of adding redundancy, and the other has the option of adding additional complexity to increase the probability of generating new degrees of freedom at the decoding time. This coordinated system takes advantage of nodes with extra processing capabilities to develop higher-order decoding, such as using a degree extension of the binary corpus $GF\ (2^n)$, $n > 2$ [15], [16].

Fulcrum coding establishes a mapping of native packet segments to coded symbols through a finite field $GF(2)$, and a respective extension $GF\ (2^n)$, $n > 2$ in the same coded vector called an expansion. For an initial condition, the outer code coding generates an expansion of r new encoded vectors. This expansion results from the linear combination of the total of a generation or g segments of coded packets through matrix C, whose entries are made up of linear coefficients generated in a finite field. There is a direct relationship between the coding and extremes of the transmission in terms of processing from the degrees of freedom defined in r and the coding extent to which the linear coefficients are generated. In (1), each entry of the new coded vector is systematically represented. This vector was added to the native packet vector P to form U [17].

$$U_i = \sum_{j=1}^{g} c_{ij} p_j, i = 1, 2, \ldots, r, \tag{1}$$

where $c_{ij} \epsilon GF\ (2^n)$, $n > 2$, the new coded vector $\boldsymbol{P}^* = [\boldsymbol{PU}]$ will be of length $g + r$. Packet expansion guarantees a high probability of receiving linearly independent packets and increases the robustness against packet losses in an erasure channel. However, this expansion requires high energy consumption owing to the complexity of the processing symbols of binary extensions [18]. It is necessary to carry out the inner code coding, a process based on RLNC coding over $GF\ (2)$ to complement the concatenation, which leads to a decrease in complexity at the time of decoding because binary coefficients are implemented. Finally, each entry of the encoded vector is of dimension $g + r$, as shown in (2).

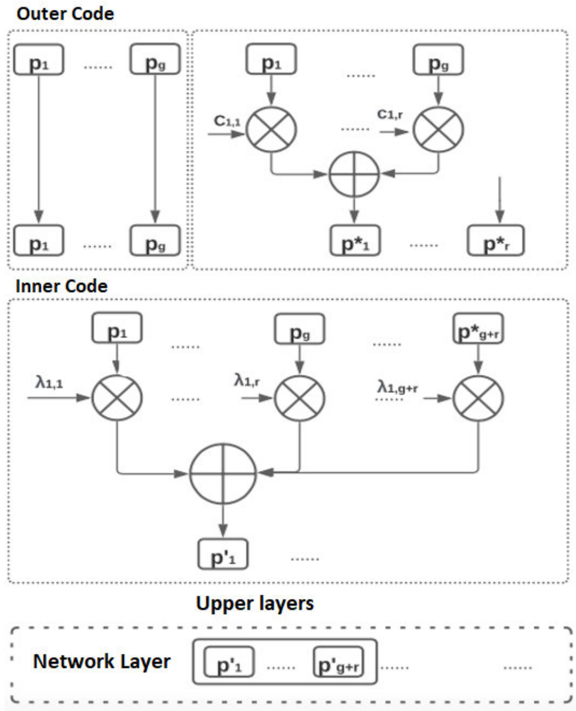

FIGURE 1. Fulcrum coding in a Cross-Layer Network Coding (CLNC) process controlled by an Application Programming Interface (API). The coding is composed of a direct sum of linear spaces and a multiplication of linear coefficients. A scheme is defined on layer 3 of the communication model and controlled with parameters that define the performance of the communication. [20].

$$p_i' = \sum_{j=1}^{g+r} \lambda_{ij} p_{ij}^*, i = 1, 2, \ldots, g + r \tag{2}$$

It should be observed that each coefficient $\{\lambda_{i,j}, c_{r,j}\}$ generated in the system is uniformly distributed in (3).

$$\Pr\{i = t\} = \begin{cases} \sigma_o, & t = 0 \\ \frac{1-\sigma_o}{q-1}, & t \in GF(2^n), n \geq 1 \end{cases} \tag{3}$$

where $0 \leq \sigma_o \leq 1$ and $\sigma_o = 1/q$. There have been advances in generating coefficients with other distributions, such as solitons [19]. Each data flow of the system is encoded through XOR ($\oplus$) and AND ($\otimes$) operators, defined in a vector subspace, and controlled at the application layer level (AL-Fulcrum, AL Application-Layer Fulcrum). It also includes a cross-layer process that relies on the transport layer for data segmentation, as Fig. 1.

However, within the same generation, the system can select the number of packet segments that enter the coding process, which we call the coding density. For example, for binary coding and the generation of ten packet segments, the system can generate five zeros (0) as linear coefficients to comply with a density of 0.5 [26]. On the other hand, each coded packet is introduced into the network with the possibility of being forwarded by other intermediate nodes or establishing new coding or recoding, which means defining new linear combinations in the information path, as illustrated in (4).

$$p_i'' = \sum_{j=1}^{g+r} z_{ij} p'_{ij},\ i = 1, 2, \ldots, g+r \tag{4}$$

Recoding determines the same operations as coding, meaning that new random coefficients are generated to operate on the uncoded information. It can be said that the recoding coefficients update the coding coefficients. With new linear combinations, the linearly dependent vectors at each point of the network are eliminated, thus generating a higher probability of obtaining an innovative vector or coded vector that increases the rank of the decoding matrix. Additionally, this new population of packets from recoding decreases and maintains a constant probability of packet loss ϵ over an erasure channel. That is, for an R-node transmission with recoding, the forwarding probability is characterized by (5).

$$Min\left[(1-\epsilon_1), (1-\epsilon_2), \ldots\ldots, (1-\epsilon_R)\right] \tag{5}$$

In contrast, under normal conditions, without recoding, the probability of sending will be $\prod_{i=1}^{R}(1-\epsilon_i)^i$; as a consequence, recoding will be a potential mechanism to minimize the error in the system, an error that is caused by channel fading and the effect of erasure on the information. This effect can be characterized by absorbing the error through random linear coefficients $\tau \in GF(q^n), n \geq 1$, according to (6).

$$p_{ei}'' = \sum_{j=1}^{g+r} z_{ij} p'_{ij} + \sum_{j=1}^{g+r} \tau \epsilon_{ij},\ i = 1, 2, \ldots, g+r \tag{6}$$

Nevertheless, it is important to note that at each receiver or forwarding node, it is possible to consider a vector subspace composed of encoded vectors or packets that arrive at the memory buffer of the receiver. This vector space is in a state of transition because new information constantly arrives. The decoder algorithm captures a set of encoded vectors and separates their linear coefficients to determine whether they are innovative. Through this procedure, it is also possible to detect linear dependence, which is usually the product of an error in the transmitter channel that creates a blurring effect in the linear system. Continuing with this line of analysis, when the receiver obtains a full rank matrix, each encoded vector is seen as a linear combination $\boldsymbol{P}'' = \boldsymbol{C} \times \boldsymbol{P}$, [21]; for that reason, to find the native packet segments, it is necessary to find $\boldsymbol{C}^{-1}$, [22].

The decoding process is performed using stepwise Gaussian elimination [23], [24]. As each coded packet allows storage of its linear coefficients, decoding can be performed only with the information of each packet [25]. NC attempts to suppress the disadvantages of current protocols, such as TCP, which is characterized by a Stateless Protocol, meaning that it does not preserve the states of previous request sessions. Although it improves the scalability of packet networks, this feature decreases performance by increasing the redundant data sent by specific retransmission requests. Instead, NC obtains complementary information from packets that have not been decoded to avoid retransmission of requirements, which makes it possible to decode each packet in the system, including at any point in the transmission, regardless of the session of each stream, because each packet contains the necessary information from the encoder field and its respective linear coefficients.

III. PROPOSED FULCRUM TUNING ADJUSTMENT

The design aims to mitigate media errors and erasure problems by gradually increasing coding complexity. This goal was achieved by controlling the density w in the coding matrix. In addition, because Fulcrum is rateless code, it is possible to generate a potentially unlimited sequence of coded packets from a set of native packets at the source such that these packets can ideally be recovered from any subset of coded packets of size equal to or slightly larger than the number of coded packets at the source.

Rateless codes generate a redundant amount δ of packets up to a set point in the communication, which indicates that they have the advantage of compensating for system losses caused by transmission errors and slowing down the injection of coded packets until all native packets are recovered at the endpoint of the communication. Based on the above, one way to achieve a gradual increase in communication as a function of coding density is to obtain the receiver degrees of freedom i to reach w, according to (7), [27].

$$w(i) = (g+u)\, min\left\{\frac{1}{2}, 1 - \sqrt[g+u-i]{1 - \frac{g+r}{g+r+\delta}}\right\} \tag{7}$$

The density of the inner code is required to be $w < 0.5$ because the linear coefficients are defined in $GF(2)$; therefore, their uniform distribution is a randomly generated average value. Owing to feedback from the receiver, it is possible to obtain the current rank i of the matrix in the decoder. Consequently, it is feasible to obtain the probability of generating innovative vectors with minimum complexity at the origin [28], [29] according to (8).

$$P_{inno}(i, g+r) \geq 1 - \left[1 - \frac{w}{(g+r)}\right]^{g+r-i},\quad i = 0, 1, \ldots, g-1 \tag{8}$$

where $0 \leq u \leq r$; that is, because r is the expansion of the encoded vector, this can also be gradual and controlled through u. In this case, $r = u$.

A. CODING REGIONS AND FEEDBACK

Because of delays in the network, it is almost impossible to obtain exact information from the decoder in real time; the solution to this problem is to define coding regions. To approximate the matrix rank at the decoder, the scheme represents a certain number of feedbacks according to the fulfillment of the $k = g+r$ degrees of freedom at the decoder. Consequently, this feedback allows the source to regularly update the coding density at the origin according to (9).

$$i(k) = \left\lfloor [k] \cdot \frac{2^k - 1}{2^k} \right\rfloor, k \in \left[1, 2, .., Log_2(k) + 1\right] \quad (9)$$

Each time the degrees of freedom corresponding to a specific area were completed, feedback was sent to adjust the matrix density during the coding process. However, if no feedback is obtained in response, the range at the decoder is assumed to grow gradually until a density of 50% is reached for binary coding. In addition, an overestimation of the receiver range must be calculated because of packet losses caused by transmission. The latter largely depends on the degree of congestion in the network, type of wireless transmission per channel, and processing level of the end devices [30].

B. PROPOSED MATHEMATICAL MODEL

According to the coding process and header fields of a coded packet, and as proposed by Muriel and Sundarararajan [9], linear coefficients can be stored in the packet for decoding in conjunction with the respective binary extensions, which can produce more degrees of freedom at the time of decoding.

First, one starts from the probability of obtaining k linearly independent vectors generated by inner code coding, represented by packets encoded in $GF(2)$, a k-dimensional vector subspace, which can generate 2^k different encoded vectors. In addition, the zero vector is explicitly excluded from the extraction urn in a real implementation; hence, the probability in (10).

$$\frac{2^k - 1}{2^k - 1} = 1 \quad (10)$$

For the second option of an innovative vector, it is necessary to discard any scalar multiple of the received vector. Therefore, the probability is defined in (11).

$$\frac{2^k - 2}{2^k - 1} \quad (11)$$

Subsequently, as innovative vectors arrive, they are accumulated and discounted from the ambient space $GF\left(2^k\right)$, and because each arrival of a vector is taken as an independent event, the probability of obtaining k innovative vectors is defined in (12) [31].

$$P_{gk} = \prod_{i=1}^{k} \left[\frac{2^k - 2^{i-1}}{2^k - 1}\right] \quad (12)$$

Thus far, the coding for a low-complexity Fulcrum inner code transmission that includes only binary coefficients has been characterized, in this case, for devices with low processing capacity. However, to increase the degrees of freedom in the receiver supported by an expansion of binary compatible codes, a concatenation of codes was developed that takes advantage of those devices with processing properties to exploit more complex transmissions. Therefore, it is feasible to create Fulcrum outer code decoding, where the increased coded vector resulting from the expansion increases the probability of generating innovative vectors at the beginning of the transmission. This model is specified in (13).

$$\begin{aligned} P_{ogk}(g, r) &= \left[\frac{2^k - 2^{i-1}}{2^k - 1}\right] \cdots \left[\frac{2^k - 2^{g-1}}{2^k - 1}\right] \\ &\quad \cdot \left[\frac{q^k - q^g}{q^k - 1}\right] \cdots \left[\frac{q^k - q^{g+r-1}}{q^k - 1}\right] \\ &= \prod_{i=1}^{g} \left[\frac{2^k - 2^{i-1}}{2^k - 1}\right] \cdot \prod_{j=g+1}^{g+r} \left[\frac{q^k - q^{j-1}}{q^k - 1}\right] \end{aligned} \quad (13)$$

It is essential to differentiate the probabilities of generating innovative vectors according to the system's complexity because at the moment of initiating communication through the session layer, it is essential to exchange the necessary parameters to determine the coding most in line with the properties of the codes. It is also necessary to map packet segments according to the Maximum Transmission Unit (MTU) requirements and streaming implemented through the transport protocol. It should be noted that rateless coding is generated because, owing to the stochastic effects of the data channel, the packet is subjected to transmission under the erasure effect. Consequently, as many generations as necessary must be produced to complete total decoding at the receiver. For this, we start by modifying the model generated by Trullols–Cruces [32], which considers the generation of extra packet segments in rateless coding as mutually exclusive events. In (14), we examine the modified Fulcrum Rateless $N = k + 1$ model [33].

$$\begin{aligned} P_{ogk}(g, r, N) &= P_{gk}(g, r) \cdot \sum_{i_1=1}^{g} \frac{2^k - 2^{i_1-1}}{2^k - 1} \\ &\quad \cdot \prod_{j_1=g+1}^{g+r} \left[\frac{q^k - q^{j_1-1}}{q^k - 1}\right] \end{aligned} \quad (14)$$

The modified model for a Fulcrum transmission is more robust and flexible. Adding this additional information vector would compensate for any additional linear combinations needed to decode each generation. There is a way to generate as many extra $N = k+\delta$ packets as required until the rank of each decoding matrix is obtained, as stated in (15).

$$\begin{aligned} &P_{ogk}(g, r, N) \\ &= P_{gk}(g, r) \sum_{i_1=1}^{g} \frac{2^k - 2^{i_1-1}}{2^k - 1} \prod_{j_1=g+1}^{g+r} \left[\frac{q^k - q^{j_1-1}}{q^k - 1}\right] \\ &\quad \cdot \sum_{i_2=i_1}^{g} \frac{2^k - 2^{i_1-1}}{2^k - 1} \prod_{j_2=g+1}^{g+r} \left[\frac{q^k - q^{j_2-1}}{q^k - 1}\right] \\ &\quad \cdots\cdot \sum_{i_{(\delta)}=i_{(\delta-1)}}^{g} \frac{2^k - 2^{i_{(\delta)}-1}}{2^k - 1} \prod_{j_{(\delta)}=g+1}^{g+r} \left[\frac{q^k - q^{j_{(\delta)}-1}}{q^k - 1}\right] \end{aligned} \quad (15)$$

The advantage of the rateless coding property is its efficiency in implementing it in unidirectional multicast communications because, owing to the nature of FEC code correction, based on linear codes, it is possible to reduce the effect of erasure correcting code (ECC) erasure, which ultimately translates into increased decoding probability. In addition, this effect consistently reduces system signaling, as it only generates the necessary feedback to inform the receiver that the limit of a coding region has been reached.

For this type of distributed coding mechanism, there is no need for additional coordination between the communication points to create missing packets in the transmission. These packets are originated by linear combinations of random coefficients under a rateless transmission type and are sensitive to controlled by complementary factors such as sliding window control [38].

C. PACKET COMPENSATION FOR ERASURE EFFECT AND DECODING PROBABILITY

It is crucial to determine whether data transmission occurs on a channel with an erasure effect ϵ_h, where h is the transmitting channel. In addition, the transmitting source system must generate δ extra coded packets, the necessary ones, until the receiver obtains the entire generation of the g native packet segments. Therefore, the probabilistic model that characterizes decoding according to these conditions, where a gradual adjustment of the system by the receiver is considered, is defined through a binomial probability model that considers the transmission of a total of $N = k + \delta$, as stated in (16) [32], [34].

$$F_h(\delta) = \sum_{m=k}^{N} \left(\frac{\mathrm{k}+\delta}{k}\right) \epsilon_h^{N-m} (1-\epsilon_h)^m \cdot P_{ogk}(g, r, N) \cdot P_{inno}(i, g+r) \tag{16}$$

Implementing many links l, where each link contains its erasure level $\epsilon_h(0 \leq h \leq l)$, is feasible when considering a channel model with memoryless erasure effect. It is essential to consider this aspect because many current Multiple-input Multiple-output (MIMO) transmission technologies are based on their efficiency in implementing multiple frequency-hopping spread-spectrum (FHSS) transmission channels. Therefore, the probability of reception increases as the number of links increases, when copies of the same coded vector are sent. This new generalized model for a total of m native packet segments is defined in (17).

$$P_{m_i} = \sum_{\varrho=1}^{l} \prod_{h=1}^{m} F_h(\mathrm{N}_\varrho = \delta) \tag{17}$$

By sending multiple signals through different transmitter channels, the receiver can reconstruct information by selecting the maximum decoding probability. The following section describes the algorithm for sending a single generation and communication link, as well as decoding delay $\bar{d}$ as a performance metric in the system. For this, it is necessary to understand that data transmission is established over a channel with time slots, and for each slot, a coded data packet or a signaling packet is sent. To understand this delay metric based on finite fields, the lower and upper bounds of $\bar{d}$ must be considered. This was proved by I. Chatzigeorgiou and A. Tassi in corollary 1 in [35], which we state below:

Consider a transmitter that employs RLNC over $GF(q)$ on L source packets, and generates coded packets. Suppose that a potentially infinite number of coded packets can be transmitted over a broadcast channel to multiple receivers. In this case, the average decoding delay $\bar{d}$ incurred by each receiver is bounded as follows:

$$\frac{L}{1-\epsilon} \leq \bar{d} < \frac{1}{1-\epsilon}\left[L + \frac{q\left(1-q^{-L}\right)}{(q-1)^2}\right] \tag{18}$$

Therefore, the minimum delay under an instantaneous suppression and feedback effect is defined as $\bar{d} + \bar{D}$, where $\bar{D}$ is the packet delay in the network, that is, the time it takes from the source to destination under point-to-point transmission.

IV. PROPOSED FULCRUM TUNING CONTROLLER

The objective is to offer a protocol that guarantees a reliable transmission over the transport layer. Data segmentation is performed according to the system frame size and with the minimum signaling load to guide the connection. A solution that exploits the advantages offered by the linear combinations of the inner code coding, which aims to increase a robust system against packet loss caused by a channel with an erasure effect, as well as the advantages of outer code coding, which is necessary to exploit distributed coding in a network with devices with high processing capacities. In the end, all of the above translates into creating a distributed and flexible code capable of establishing a balance between complexity and energy efficiency in the system because, although the complexity of the system increases, the transmission time is reduced owing to the FEC capabilities. Equally important, the coding scheme allows for a decrease in the transmission complexity by tuning each generation dispatch with respect to the receiver degrees of freedom. Under this design, it is necessary that, at the beginning of each data transmission, a session is defined, where an exchange of essential parameters is performed to establish the number of feedbacks required for the encoder algorithm, as well as the number of generations and expansion level in the data vector to be encoded. See Fig. 2.

Through this feedback process, the density of the coding W matrix can be adjusted. Matrix calculates the number of coefficients required to generate innovative packets with minimum complexity. The scheme suggests increasing the coding density as the receiver(s) accumulates more linearly independent packets of generation. Jointly, δ extra coded vectors are generated, thanks to the fact that the error rate of the transmission channel is known; therefore, it is possible to compensate for packet loss caused by system erasures. Consequently, it is possible to achieve any fraction of throughput in exchange for an additional delay fraction resulting from packet forwarding, owing to the lack of degrees of freedom in decoding.

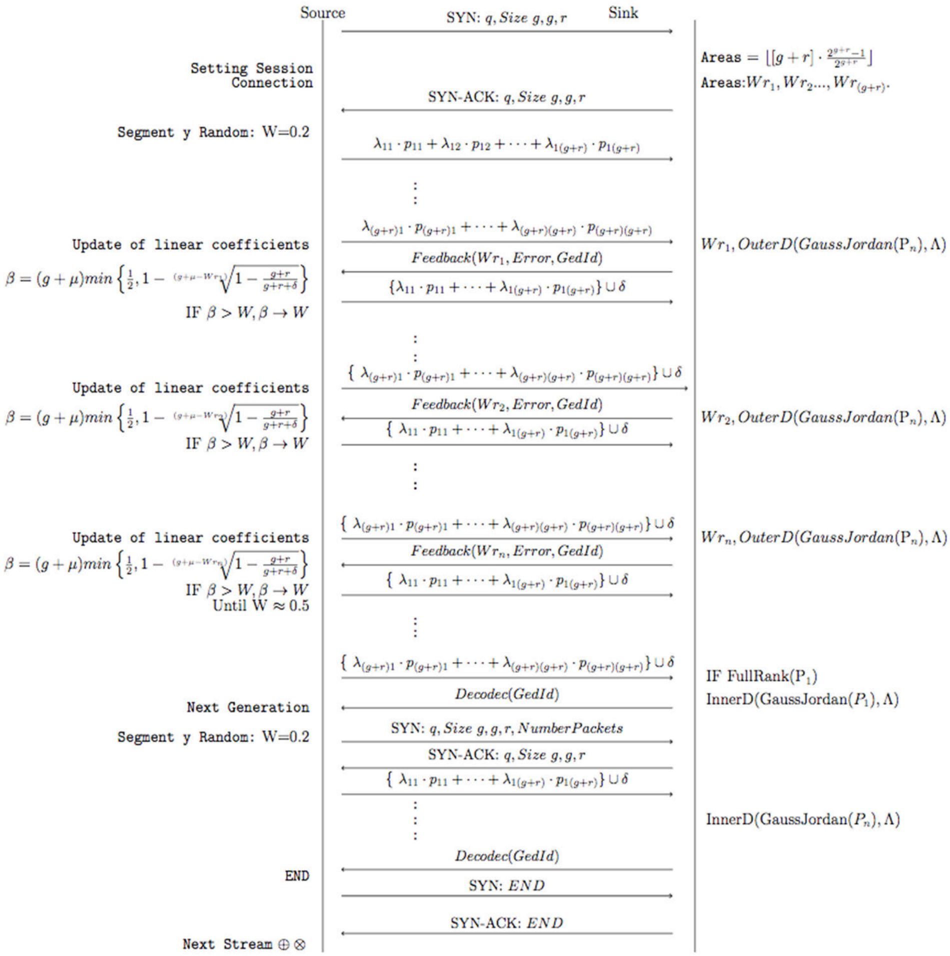

FIGURE 2. **Fulcrum Rateless Adapter Protocol signaling diagram.**

A. CODING ALGORITHMS WITH ADAPTIVE TUNING AND ADJUSTMENT

The process comprises the first stage of expansion or systematic coding, called outer, and a second stage defined by RLNC coding, called the inner code. See Algorithm 1.

where m is the total number of generations to be encoded. Consequently, the following process is a gradual control of the transmission with the help of specific parameters of the receiver: the current range of the Wr decoder matrix to control the coding density, percentage of packet loss error, and identification of decoded GedId generations to generate only linear combinations of segments that have not yet been decoded. Therefore, it is important to have functions that allow the random selection of packet segments that have not been decoded, and the generation of linear coefficients to control the coding density. See Algorithm 2.

At the end of each generation transmission, the density of the matrix reaches its maximum value of $W = 0.5$, meaning that only 50% of the packet segments are sent. Above this value, congestion occurs in the network; hence, the density of the matrix has an upper limit.

B. ADAPTIVE DECODING ALGORITHM

The decoding process, as defined, involves a stepwise Gauss–Jordan reduction of a rank k matrix. However, the system can obtain linear coefficients of a binary extension generated in the outer code coding, giving rise to extra encoded data to complete the linear system for reduction. This last event shortens the decoding steps because the linear system goes from having rank $g+r$ to rank g, with coefficients absorbed by binary extensions. See Algorithm 3.

In the decoder algorithm, we identified that the system only executes the feedback when the decoding matrix has reached the range number or when the limit of the areas to be agreed upon from the start of the session has been completed.

The extension property in Fulcrum coding leads to the intuitive assumption that the decoder algorithm outperforms the standard RLNC scheme in terms of computational complexity because working with binary extensions exploits the computational capabilities of the receiving system to reduce decoding times. It is feasible to increase the gradual complexity in decoding to obtain additional states because the table of symbols to work with increases.

Algorithm 1 Coding - Outer Code

Require: Generation $\boldsymbol{g} > 0$,Expansion $r > 0$
Native Packets $P_i = \{p_1, p_2, p_3, \ldots, p_g\}$
Ensure: Coding Packets $P_i^* = \{p_1, p_2, \ldots, p_g, p_1^*, \ldots, p_r^*\}$
$\boldsymbol{k} \leftarrow 1$
while $\boldsymbol{k} < \boldsymbol{m}$ **do**
$\boldsymbol{j}$ <- 1
for $\boldsymbol{j} < \boldsymbol{r}$ **do**
$\boldsymbol{i} \leftarrow 1$
$\mathcal{U}_i \leftarrow 0$
$J \leftarrow \emptyset$
while $\boldsymbol{i} < \boldsymbol{g}$ **do**
$\boldsymbol{j}$ <— Random Segment Packets {1,2,3, **..,*g*}** — ***GedId**J***
$\boldsymbol{c}$ <— Random coefficients $\backslash GF(2^n)$,n > 1
$J \leftarrow J \cup \{j\}$
$\mathcal{U}_i \leftarrow \mathcal{U}_j \oplus (c_{ij} \cdot \mathcal{P}_i)$
$i \leftarrow i + 1$
end while
$\mathcal{P}^* \longleftarrow \mathcal{P}$ U $\mathcal{U}_j$
end for
$k \leftarrow k + 1$
end while

Algorithm 2 Coding - Inner Code

Require: N Generations $m > 0$, Generation $g > 0$, Expansion $r > 0$
Segment Packets $\mathcal{P}_i^* = \{p_1, p_2, \ldots, p_g, p_1^*, \ldots, p_r^*\}$.
Ensure: Coding Packets $\mathcal{P}'_i = \{q_1, q_2, q_3, \ldots, q_{g+r}\}$.
$GedIr \leftarrow \emptyset$
$i \leftarrow 1$
while $i \leq m$ **do**
$Wt \leftarrow Wt = (g + \mu) \left\{ \sqrt[9+\mu]{1 - \frac{g+r}{g+r+\delta}} \right\}$
$\mathcal{Q}_i \leftarrow \overrightarrow{0}$
$\delta \leftarrow \emptyset$
$\lambda 1 \leftarrow \emptyset$
$Ar \leftarrow$ SendArea (g)
while True **do**
if (*Received* ()) **then**
[*Wr Error* $\leftarrow$ Received ()
if FullRank(Wr) then
$i \leftarrow i + 1$
Break
end if
$\delta \leftarrow$ Rateless (*Error*)
$\lambda 1 \leftarrow (n + r) \cdot Min \left((Densi(Wr, \delta, g, r)), \frac{1}{2} \right)$
end if
if $(\lambda 1 > Wt)$ **then**
$Wt \leftarrow \lambda 1$
end if
$\lambda_i \leftarrow$ Random coefficients $\backslash GF(2) y Wt = \lambda 1$
$\mathcal{P}_i^* \leftarrow$ Random Segment Packets $\{1, 2, 3, \ldots, g + r\} - GedId$
$\mathcal{Q}_i \leftarrow \mathcal{Q}_i \oplus (\lambda_i \cdot \mathcal{P}_i^*)$
$\mathcal{P}_i' \leftarrow \mathcal{Q}_i \cup \delta$
Send$'$ $(\mathcal{P}'_i)$
end while
end while

It is important to note that the only nodes that can encode are the origins because recoding at intermediate points disables the tuning owing to the lack of density control in the coding; therefore, there is no control over the complexity of the communication.

Algorithm 3 Adaptative Decoding

Require:
Require: N Generations $m > 0$, Generation p_j':, λ_j coding coefficient, Expansion $r > 0$, g: Generation size, Error : Error Channel, Com : Complex device
High, Low
Ensure: $\mathcal{P}$:Native Packet
$j \leftarrow 1$
while $j \leq m$ **do**
$i \leftarrow 1$
$Ar \leftarrow ObtainArea()$
while i $\leq g + r$**do**
p_j': $\leftarrow$ new incoming coded packet
λ_j':$\leftarrow$ new incoming coding vector
P_i' :$\leftarrow 0$
if *Depnendency* (P_i', Λ) *then*
continue
end if
if $|P_i'| \geq n + r$ **then**
$P_i' \leftarrow InnerD\left(Gauss\ Jordan\ \left(P_{ij}'\right), \Lambda\right)$
else
if $Com = Alta$**then**
$P_i' \leftarrow OuterD\left(GaussJordan\left(P_{ij}'\right), \Lambda\right)$
else
continue
end if
end if
if $P_i' \neq 0$ **then**
$P_{ij} \leftarrow P_{ij} \cup \{P_i'\}$
$[WrGedId] \leftarrow issuccessfulAre\ (P_{ij})$
$Error \leftarrow ObtainError()$
if FullRank $(A\ r)$ **then**
Decode(GedIdr)
$i \leftarrow i + 1$
break
else if CheckArea(Wr,Ar) **then**
$Feedback(Wr, Error)$
end if
end if
$i \leftarrow i + 1$
$Wr \leftarrow 0$
$\Lambda \leftarrow 0$
end while
$j \leftarrow j + 1$
end while

V. ANALYSIS OF FRAMEWORK FULCRUM PERFORMANCE WITH TUNING CONTROLLER

A simulation with point-to-point transmission from the source node to the destination node was set up for the design experiments. The data come from a random sampling scheme with 30 (thirty) samples per simulation state, guaranteeing the normalization of the data with a confidence interval corresponding to 95 %. Table 1 summarizes the experimental parameters, performance metrics, factors, and levels for the different scenarios [36], [37].

TABLE 1. Parameters and levels.

Parameter	Value
Inner code $\mathbb{F}_q$	*GF (2).*
Outer code $\mathbb{F}_q$	*GF (2^4), GF (2^5), GF (6^6), y GF (6^7).*
Rateless	Not Rateless, Rateless k+1, Rateless k+2
RLNC	*GF (2), GF (2^4), GF (2^5), GF (2^6) y GF (2^7)*
Generation (g)	18
Expansion (r)	5
Packet frame	1518 bytes.
Channel erasure ϵ %	[10 20 30 40 50 60 70 80 90].
Operative system	Linux-Ubuntu
Programming language	Python
Coding library	API-KODO
Number of random simples	10000
Processor and RAM	Icore 5 y 8 GB

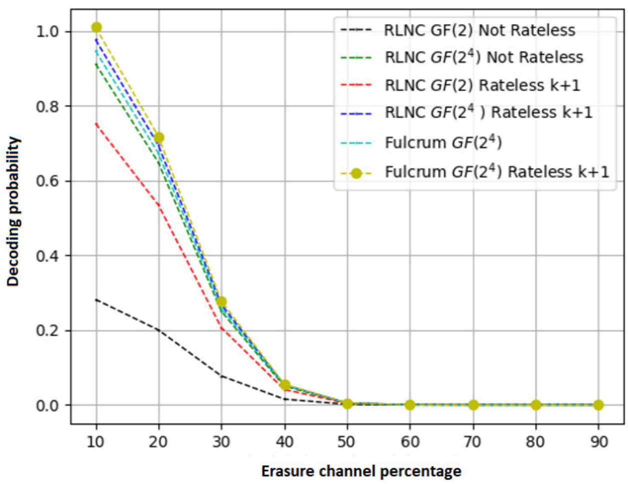

FIGURE 3. **The probability of decoding with different levels of erasure is simulated through a binomial probability.**

The maximum allowed frame size is set for each packet, with a header overhead of 8 bytes for the link layer (LLC), Logical Link Control, and overhead due to the storage of linear codes. Care should be taken when selecting the block size or g-generation, because choosing a large block can result in a significant latency overhead, even more than what would be induced by retransmission. Conversely, selecting a small block may be insufficient to protect against packet loss.

The decoding probability at the receiving node was analyzed to evaluate the performance of the proposed scheme. Data traffic and a gradual opening of the coding window are determined because the transmission must be tuned, which generates several coding coefficients proportional to the degrees of freedom in the decoding matrix with a maximum of 50% of the number of generations to be coded. It should be noted that the experiment uses a generation g of 18 segments of native packets and an r of five segments of packets encoded in a binary extension. The decoding probability was examined to determine the efficiency of data reception, as shown in Fig. 3.

The Fulcrum rateless scheme determines the maximum decoding probability, where additional traffic with the same

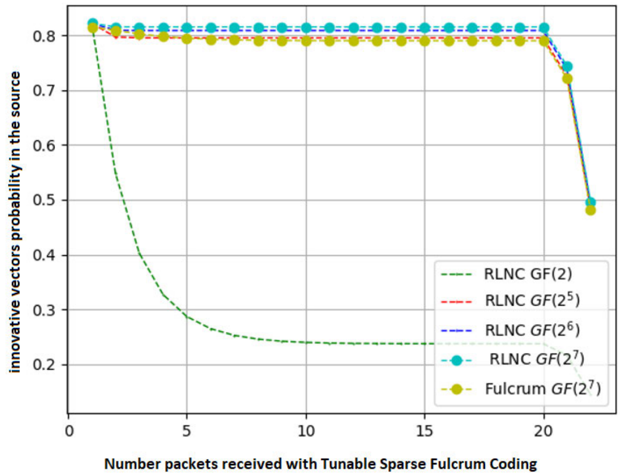

FIGURE 4. **Fulcrum Adapter Tuning Protocol with the transmission of $g + r = 23$ segments and $i = 3$ feedback corresponding to the generation of 3 areas, with degrees of freedom of 12, 18, and 21.**

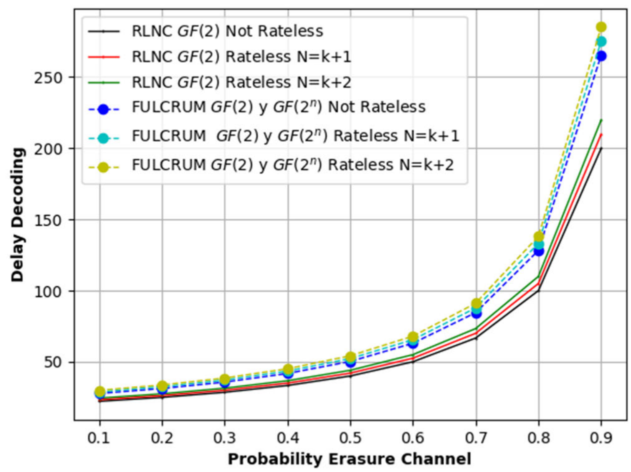

FIGURE 5. **Rateless Decoding delay with different erasure levels simulated through a binomial probability, with (g=18 and r=5).**

characteristics as encoded traffic is delivered. This scheme shortens the transmission time. However, in a tuned transmission, it is observed that Fulcrum coding maintains the maximum probability of generating innovative vectors, as shown in Fig. 4.

Fig. 4 shows that the difference between the decoding probabilities of Fulcrum and RLNC over $\boldsymbol{GF}\left(\mathbf{2^7}\right)$ is negligible, except for RLNC coding over $\boldsymbol{GF}(2)$. The probability of generating innovative vectors decays abruptly from the beginning of transmission. It is also evident that through Fulcrum coding, it is possible to create innovative vectors with lower complexity than RLNC coding, where all its inputs are encoded in the same binary extension. The delay in each decoding is shown in Fig. 5.

Fulcrum coding uses the flexibility of components. Indeed, it is realized by combining two encodings: the first, called inner coding, is performed on the binary field $\boldsymbol{GF}(2)$, while the other, called outer coding, is performed using an extension of the binary field, say $\boldsymbol{GF}\left(\mathbf{2^n}\right)$. This process allows us to

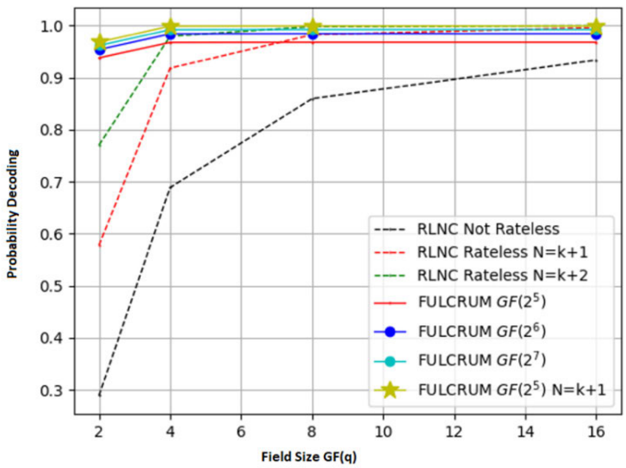

FIGURE 6. Decoding delay with different coding field sizes at the source, with (g=18 and r=5).

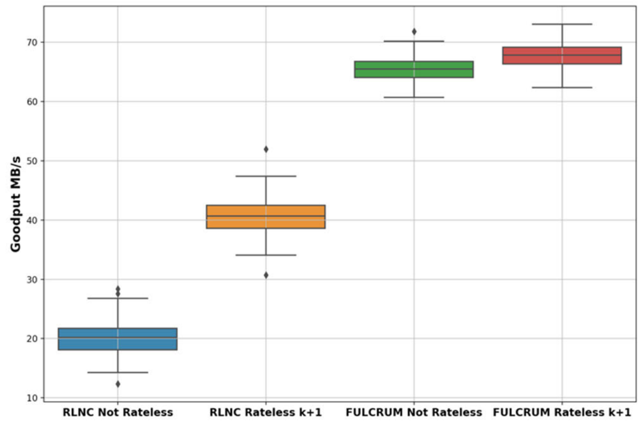

FIGURE 7. Transmission generated under a 100MB/s channel, with a 70% erasure probability.

take advantage of the mapping between vector spaces to take part of a generation encoded in a binary extension and reduce it in another vector subspace to complete the full decoding of the data. It implies the generation of innovative vectors through linear combinations. While this mapping between vector spaces allows for reducing the complexity of the system, on the other hand, it requires an increase in computational power. Significant research now promises to mitigate these effects [39].

Although a higher complexity implies a longer decoding delay, this is compensated for by the suppression of redundant messages owing to the linear dependency caused by lower base coding. Likewise, FEC-type coding facilitates the elimination of signaling traffic owing to error and transmission control, as well as the additional traffic caused by packet requirements. Similarly, as shown in Fig. 6, the field size at the coding time influences the decoding probability.

The capabilities of Fulcrum coding are highlighted, which defines a maximum probability with the coding of (18/23) 78% in each of its binary-based $GF(2)$ inputs and the other 22% under a respective extension, which is used in high-level processing devices. In this case, the binary inputs are mapped to extensions to achieve decoding with fewer steps, owing to the reduction in dimensions.

Finally, A traffic encoded with Fulcrum code Rateless directly increases the overall resilience of the system, thus generating an efficiently obtained Goodput at the receiver, as shown in Fig. 7.

While generating data in $GF(2)$ RLNC encoders require less complexity, it also leads to the generation of linearly dependent vectors or redundant information in the system, which affects a decrease in Goodput at the final receiver of the data.

A. THEORETICAL ANALYSIS

The model consideration is a Hybrid Automatic Repeat request (HARQ) scheme that implements Automatic Repeat reQuest (ARQ) methods at the source of the transmission for error control, as well as at the final destination (FEC) of the system. This controlled architecture from the application layer through, an API allows receiver feedback, not by packets received, but by the fulfilment of degrees of freedom at the decoder. Specifically, the system triggers feedback for each respective and acknowledged generation of packet segments at the receiver. For this reason, distributed coding reduces the number of necessary retransmissions, partly caused by system errors and the extra signaling generated by the control packets. Instead of retransmitting the necessary messages at the receiver, these HARQ systems based on Fulcrum codes successively probe the receivers to determine how to create innovative vectors with minimum complexity by tuning while generating additional traffic to compensate for the loss caused by stochastic processes caused by the transmitter channel. Therefore, the proposed architecture aims to define a valid model for multicast streaming, generated with minimum delay requirements for its propagation, whose transmission efficiency depends on the processing performance of the receiving device.

It is important to note that the only nodes that can encode are the origins because recoding at intermediate points disables the tuning owing to the lack of density control in the coding; therefore, there is no control over the complexity of the communication.

VI. CONCLUSION

The proposed protocol is a mechanism that improves the reliability of packet transmission over other protocols that do not provide guaranteed delivery, such as the UDP protocol, especially in multicast networks and/or in applications that require low-delay tolerance. The architecture focuses on the resiliency generated by Fulcrum coding in favor of transmission with the help of linear combinations to recover strictly determined symbols due to erasures. Owing to this linear combination of packet segments in a random manner, any information is recoverable at its destination with the help of packets that have not been decoded, so there is technically considered to be no loss of information. These properties are complemented by a type of rateless code transmission, within which the sender slows down the sending of coded packets until each generation obtains a full-range matrix at the final destination, that is, a mass sending of packets is performed, conditioned by each system feedback, which reduces the system signaling, and consequently, the RTT times.

The compatibility of Fulcrum coding with higher layer services allows for a gradual deployment of the communication, which leads to the exploitation of a comprehensive scenario of heterogeneous configurations and hybrid topologies. A distributed coding architecture that takes advantage of devices with heterogeneous performance in the network was implemented to fix a balance between performance and complexity at the ends of the communication as a consequence of generating a hybrid HARQ requirement scheme that ensures connection through the recognition of degrees of freedom in the final path of the information. While developing a multicast in the network reduces the end-to-end delay overhead, additional benefits were found if exceptional capabilities and specific coding algorithms were implemented at various layers of the protocol stack, which decreased system signaling.

This study provides in-depth evidence of several aspects of distributed and flexible coding for delay-insensitive streaming data transmission. While most previous work on coding focuses on point-to-point communications, this scheme targets the exploitation of heterogeneous devices in the network to reduce the retransmissions generated by system errors. The design is motivated by numerous applications in which a gateway server with access to sufficient computing power is defined as an agent for distributed data processing and complemented by the endpoints of the communication.

REFERENCES

[1] E. Magli, M. Wang, P. Frossard, and A. Markopoulou, "Network coding meets multimedia: A review," *IEEE Trans. Multimedia*, vol. 15, no. 5, pp. 1195–1212, Aug. 2013, doi: 10.1109/TMM.2013.2241415.

[2] K. J. Ma, R. Bartos, S. Bhatia, and R. Nair, "Mobile video delivery with HTTP," *IEEE Commun. Mag.*, vol. 49, no. 4, pp. 166–175, Apr. 2011, doi: 10.1109/MCOM.2011.5741161.

[3] J. Nightingale, J. M. A. Calero, Q. Wang, and P. Salva-Garcia, "5G-QoE: QoE modelling for ultra-HD video streaming in 5G networks," *IEEE Trans. Broadcast.*, vol. 64, no. 2, pp. 621–634, Jun. 2018, doi: 10.1109/TBC.2018.2816786.

[4] G. Carlucci, L. De Cicco, and S. Mascolo, "HTTP over UDP: An experimental investigation of QUIC," in *Proc. 30th Annu. ACM Symp. Appl. Comput.*, Apr. 2015, pp. 609–614, doi: 10.1145/2695664.2695706.

[5] G. Oman and C. N. Curtis, "Problems and solutions," *College Math. J.*, vol. 49, no. 5, pp. 371–377, Oct. 2018, doi: 10.1080/07468342.2018.1531656.

[6] R. J. Yair Enrique, G. Ismael, and M. Jose, "Fulcrum coding performance on fog computing architecture," *IEEE Latin Amer. Trans.*, vol. 18, no. 11, pp. 1966–1974, Nov. 2020, doi: 10.1109/TLA.2020.9398638.

[7] V. Nguyen, J. A. Cabrera, G. T. Nguyen, F. Gabriel, C. Lehmann, S. Mudriievskyi, and F. H. P. Fitzek, "Adaptive decoding for fulcrum codes," in *Proc. IEEE 9th Annu. Inf. Technol., Electron. Mobile Commun. Conf. (IEMCON)*, Nov. 2018, pp. 133–139, doi: 10.1109/IEMCON.2018.8614865.

[8] Q. Huang, J. Li, Y. Zhu, and D. Wu, "Trickle irrigation: Congestion relief for communication with network coding," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8695–8708, Sep. 2018, doi: 10.1109/TVT.2018.2844350.

[9] J. K. Sundararajan, D. Shah, M. Medard, S. Jakubczak, M. Mitzenmacher, and J. Barros, "Network coding meets TCP: Theory and implementation," *Proc. IEEE*, vol. 99, no. 3, pp. 490–512, Mar. 2011, doi: 10.1109/JPROC.2010.2093850.

[10] C. W. Sorensen, D. E. Lucani, and M. Medard, "On network coded filesystem shim: Over-the-top multipath multi-source made easy," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–10, doi: 10.1109/ICC.2017.7996675.

[11] M. Videbæk, F. H. Paul, M. V. Pedersen, J. Heide, and F. Fitzek, "Kodo: An open and research oriented network coding library," in *Proc. Int. Conf. Res. Netw.*, vol. 6827, 2011, pp. 145–152.

[12] D. E. Lucani, M. V. Pedersen, D. Ruano, C. W. Sørensen, F. H. P. Fitzek, J. Heide, O. Geil, V. Nguyen, and M. Reisslein, "Fulcrum: Flexible network coding for heterogeneous devices," *IEEE Access*, vol. 6, pp. 77890–77910, 2018, doi: 10.1109/ACCESS.2018.2884408.

[13] S. Feizi, D. E. Lucani, C. W. Sorensen, A. Makhdoumi, and M. Medard, "Tunable sparse network coding for multicast networks," in *Proc. Int. Symp. Netw. Coding*, Jun. 2014, pp. 1–6, doi: 10.1109/NETCOD.2014.6892129.

[14] V. Nguyen, J. A. Cabrera, G. T. Nguyen, D. You, and F. H. P. Fitzek, "Versatile network codes: Energy consumption in heterogeneous IoT devices," *IEEE Access*, vol. 8, pp. 168219–168228, 2020, doi: 10.1109/ACCESS.2020.3023639.

[15] A. Markopoulou, "I^2NC: Intra- and inter-session network coding for unicast flows in wireless networks," in *Proc. IEEE INFOCOM*, Apr. 2011, pp. 1–15.

[16] D. E. Lucani, M. V. Pedersen, D. Ruano, C. W. Sorensen, F. H. P. Fitzek, J. Heide, O. Geil, V. Nguyen, and M. Reisslein, "Correction to 'Fulcrum: Flexible network coding for heterogeneous devices,'" *IEEE Access*, vol. 9, pp. 108–199, 2021, doi: 10.1109/ACCESS.2021.3101355.

[17] D. E. Lucani, M. V. Pedersen, J. Heide, and F. H. P. Fitzek, "Coping with the upcoming heterogeneity in 5G communications and storage using fulcrum network codes," in *Proc. 11th Int. Symp. Wireless Commun. Syst. (ISWCS)*, Aug. 2014, pp. 997–1001, doi: 10.1109/ISWCS.2014.6933499.

[18] G. Raja, R. Jayaraman, R. Arul, and K. Kottursamy, "Network coding with buffer scheme in multicast for broadbad wireless network," *Int. J. Electron. Commun. Eng.*, vol. 10, no. 6, pp. 1022–1027, 2016.

[19] K.-K. Yen, Y.-C. Liao, C.-L. Chen, and H.-C. Chang, "Modified robust soliton distribution (MRSD) with improved ripple size for LT codes," *IEEE Commun. Lett.*, vol. 17, no. 5, pp. 976–979, May 2013, doi: 10.1109/LCOMM.2013.040913.130136.

[20] M. S. Al-Abiad, M. Z. Hassan, and M. J. Hossain, "Cross-layer network codes for content delivery in cache-enabled D2D networks," 2021, *arXiv:2101.07291*.

[21] *Simulation of Random Linear Network Coding in Ad-Hoc Networks*, May 2012.

[22] Y. O. Soon, E.-K. Lee, and M. Gerla, "Adaptive forwarding rate control for network coding in tactical MANETs," in *Proc. Mil. Commun. Conf.*, Oct. 2010, pp. 1381–1386, doi: 10.1109/MILCOM.2010.5680142.

[23] S. Kompella and S. Chilukuri, "Fast and reliable code dissemination in WSNs," *Comput. Sci.*, vol. 1, pp. 3252–3255, Jan. 2019.

[24] T. Ho and H. Viswanathan, "Dynamic algorithms for multicast with intra-session network coding," *IEEE Trans. Inf. Theory*, vol. 55, no. 2, pp. 797–815, Feb. 2009, doi: 10.1109/TIT.2008.2009809.

[25] P. Pahlevani, M. Hundebøll, M. V. Pedersen, D. Lucani, H. Charaf, F. H. P. Fitzek, H. Bagheri, and M. Katz, "Novel concepts for device-to-device communication using network coding," *IEEE Commun. Mag.*, vol. 52, no. 4, pp. 32–39, Apr. 2014, doi: 10.1109/MCOM.2014.6807944.

[26] Y. Li, W. Y. Chan, and S. D. Blostein, "On design and efficient decoding of sparse random linear network codes," *IEEE Access*, vol. 5, pp. 17031–17044, 2017, doi: 10.1109/ACCESS.2017.2741972.

[27] P. Garrido, D. E. Lucani, and R. Aguero, "How to tune sparse network coding over wireless links," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2017, pp. 1–6, doi: 10.1109/WCNC.2017.7925561.

[28] P. Garrido, D. E. Lucani, and R. Agüero, "Markov chain model for the decoding probability of sparse network coding," *IEEE Trans. Commun.*, vol. 65, no. 4, pp. 1675–1685, Apr. 2017.

[29] W. Chen, F. Lu, and Y. Dong, "The decoding success probability of sparse random linear network coding for multicast," 2020, *arXiv:2010.05555*.

[30] A. Cohen, D. Malak, V. B. Bracha, and M. Medard, "Adaptive causal network coding with feedback," *IEEE Trans. Commun.*, vol. 68, no. 7, pp. 4325–4341, Jul. 2020, doi: 10.1109/TCOMM.2020.2989827.

[31] M. V. Pedersen, D. E. Lucani, F. H. P. Fitzek, C. W. Sørensen, and A. S. Badr, ''Network coding designs suited for the real world: What works, what doesn't, what's promising,'' in *Proc. IEEE Inf. Theory Workshop (ITW)*, Sep. 2013, pp. 1–5, doi: 10.1109/ITW.2013.6691231.

[32] B. Roux, ''The art of dissecting the function of a potassium channel,'' *Neuron*, vol. 47, no. 6, pp. 777–778, Sep. 2005, doi: 10.1016/j.neuron.2005.08.031.

[33] O. Trullols-Cruces, J. M. Barcelo-Ordinas, and M. Fiore, ''Exact decoding probability under random linear network coding,'' *IEEE Commun. Lett.*, vol. 15, no. 1, pp. 67–69, Jan. 2011, doi: 10.1109/LCOMM.2010.110310.101480.

[34] D. Vukobratovic, A. Tassi, S. Delic, and C. Khirallah, ''Random linear network coding for 5G mobile video delivery,'' *Information*, vol. 9, no. 4, pp. 1–20, 2018, doi: 10.3390/info9040072.

[35] I. Chatzigeorgiou and A. Tassi, ''Decoding delay performance of random linear network coding for broadcast,'' *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 7050–7060, Aug. 2017, doi: 10.1109/TVT.2017.2670178.

[36] J. H. M. Medard, K. Fouli, K. F. Steinwurf Aps, and S. Shi. (2020). *Random Linear Network Coding (RLNC)-Based Symbol Representation Draft-Heide-NWCRG-RLNC-02*. [Online]. Available: http://weekly.cnbnews.com/news/article.html?no=124000

[37] S. Aps, ''Kodo-ns3-examples documentation release master,'' Tech. Rep., 2015.

[38] E. Tasdemir, V. Nguyen, G. T. Nguyen, F. H. P. Fitzek, and M. Reisslein, ''FSW: Fulcrum sliding window coding for low-latency communication,'' *IEEE Access*, vol. 10, pp. 54276–54290, 2022, doi: 10.1109/ACCESS.2022.3175815.

[39] V. Nguyen, E. Tasdemir, G. T. Nguyen, D. E. Lucani, and F. H. P. Fitzek, ''Tunable expansion packets for fulcrum codes,'' in *Proc. 25th Eur. Wireless Conf.*, Aarhus, Denmark, 2019, pp. 1–7.

YAIR RIVERA JULIO received the M.Sc. degree in telematics and telecommunications in engineering from Universidad Del Norte, Barranquilla, Colombia, and the Ph.D. degree in computer science from Universidad del Norte. His research focuses on wireless network coding with multicasting. His current research interests include finite soluble fields and their applications in digital data transfer, network sensors, and the IoT.

ISMAEL GUTIÉRREZ GARCÍA received the Dr. rer'. nat. degree in mathematics from Johannes Gutenberg University Mainz, Germany. His current research interests include finite groups, graphs, combinatorics, and discrete mathematics and their applications.

JOSÉ MÁRQUEZ DÍAZ received the B.Sc. degree in computer science from Universidad del Norte, Barranquilla, Colombia, the M.Sc. degree in computer science from ITESM-México under agreement with Universidad Autónoma de Bucaramanga, and the Ph.D. degree in computer science from Universidad del Norte. His primary research interests include the quality of service (QoS) in computer networks, ad hoc networks, and network coding in multicasting.