entropy

MDPI

Article

Scalable Network Coding for Heterogeneous Devices over Embedded Fields

Hanqi Tang [1], Ruobin Zheng [2], Zongpeng Li [3], Keping Long [1] and Qifu Sun [1,*]

[1] Department of Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
[2] Network Technology Lab, Huawei Technologies Co., Ltd., Shenzhen 518000, China
[3] Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084, China
* Correspondence: qfsun@ustb.edu.cn

Abstract: In complex network environments, there always exist heterogeneous devices with different computational powers. In this work, we propose a novel scalable random linear network coding (RLNC) framework based on embedded fields, so as to endow heterogeneous receivers with different decoding capabilities. In this framework, the source linearly combines the original packets over embedded fields based on a precoding matrix and then encodes the precoded packets over GF(2) before transmission to the network. After justifying the arithmetic compatibility over different finite fields in the encoding process, we derive a sufficient and necessary condition for decodability over different fields. Moreover, we theoretically study the construction of an optimal precoding matrix in terms of decodability. The numerical analysis in classical wireless broadcast networks illustrates that the proposed scalable RLNC not only guarantees a better decoding compatibility over different fields compared with classical RLNC over a single field, but also outperforms Fulcrum RLNC in terms of a better decoding performance over GF(2). Moreover, we take the sparsity of the received binary coding vector into consideration, and demonstrate that for a large enough batch size, this sparsity does not affect the completion delay performance much in a wireless broadcast network.

Keywords: random linear network coding (RLNC); wireless broadcast network; scalable network coding

Citation: Tang, H.; Zheng, R.; Li, Z.; Long, K.; Sun, Q. Scalable Network Coding for Heterogeneous Devices over Embedded Fields. *Entropy* **2022**, *24*, 1510. https://doi.org/10.3390/e24111510

Academic Editors: Shenghao Yang and Kenneth Shum

Received: 29 September 2022
Accepted: 17 October 2022
Published: 22 October 2022

1. Introduction

In a communication network, linear network coding (LNC) advocates intermediate nodes to linearly combine received messages before transmission, so as to improve various network performances, such as increasing network throughput, reliability, and reducing transmission delay. Random linear network coding (RLNC) provides a distributed and asymptotically optimal approach for linear coding with coefficients randomly selected from a base field [1]. It shows the potential to improve the performance of unreliable or topologically unknown networks such as D2D networks [2], ad hoc networks [3], and wireless broadcast networks [4–7].

One of the reasons that hinder the large-scale practical applications of RLNC is the compatibility issue of different computational overheads. In complex network environments, there exist heterogeneous devices with different computational powers. Specifically, sources and certain receivers usually have ample computational powers while a large number of intermediate nodes and other receivers are computationally constrained such as the data collectors in ad hoc networks or low-cost devices in the Internet of Things paradigm [8]. It turns out that the coding compatibility among heterogeneous devices with different computational powers has to be considered in RLNC design.

This paper proposes a novel framework for scalable RLNC design based on embedded fields. The adjective *scalable* means that the finite fields chosen in the encoding process are not limited to a single base field but a set of *embedded fields* which consists of a large finite

field and all its subfields. The encoding process at the source consists of two stages. In stage 1, based on a precoding matrix, all original packets are linearly combined over different finite fields to form precoded packets. In stage 2, the final packets to be transmitted are formed by randomly combining the precoded packets over GF(2). The heterogeneous receivers can recover the original packets over different fields under different computational constraints.

It is worthwhile to remark that prior to this work, there have been studies [9–14] that have taken different fields into account in the course of RLNC design. On one hand, the so-called Telescopic codes [9–11] and Revolving codes [12] considered different fields aiming at reducing the decoding complexity. However, they assume that all receivers have the same decoding capability, that is, they need support the arithmetic over the largest defined finite field. On the other hand, a flexible RLNC scheme called Fulcrum [13,14] makes use of GF(2) and its extension field GF(2^8) for code design and it supports receivers to decode over both fields. Actually, Fulcrum can be regarded as a special instance in our proposed framework, while the decoding rule over GF(2) considered therein is weaker than the one proposed in this paper. In addition, there is limited discussion on the construction of an optimal encoding matrix in Fulcrum.

The main contributions of this paper are summarized as follows.

- We mathematically justify how to make the arithmetic over different finite fields compatible.
- We derive a necessary and sufficient condition for decodability at a receiver over different finite fields. In particular, the proposed decoding rule over GF(2) is stronger than the one proposed in Fulcrum.
- We theoretically study the construction of an optimal precoding matrix in terms of the decodability performance.
- By numerical analysis in classical wireless broadcast networks, we demonstrate that the proposed scalable RLNC not only guarantees a better decoding compatibility over different fields compared with classical RLNC over a single field, but also provides a better decoding performance over GF(2) in terms of smaller average completion delay compared with Fulcrum.
- In numerical analysis, we also take the sparsity of the received binary coding vector into consideration, and demonstrate that for a large enough batch size, this sparsity does not affect the completion delay performance much in a wireless broadcast network.

This paper is structured as follows. Section 2 reviews the mathematical fundamentals of embedded fields. Section 3 first presents the general principles of the proposed scalable RLNC framework and then formulates the encoding and decoding process. Section 4 investigates the design of an optimal precoding matrix. Section 5 numerically analyzes the proposed scalable RLNC and compares its performance with classical RLNC over a single finite field as well as Fulcrum. Moreover, we take the sparsity into consideration and illustrate the influence on its performance. Conclusion is given in Section 6.

2. Mathematical Fundamentals

In our proposed scalable RLNC framework, different receivers will be able to recover the original packets over different finite fields, upon their different computational powers. In order to make the arithmetic over different finite fields compatible, we need the concept of embedded fields, which will be briefly reviewed in this section. One may refer to [15] for a detailed introduction on finite fields.

Recall that a finite field GF(2^{d_1}) is a subfield of GF(2^{d_2}) if and only if $d_1|d_2$. Thus, GF(2^{d_1}), GF(2^{d_2}), ..., GF(2^{d_D}) are said to form embedded fields $\mathcal{F}$ if $d_1 < d_2 < \ldots < d_D$ and $d_1|d_2|\ldots|d_D$. For arbitrary GF(2^{d_i}) and GF(2^{d_j}) in $\mathcal{F}$ with $i < j$, as GF(2^{d_j}) can be regarded as GF$\left((2^{d_i})^{d_j/d_i}\right)$, it can be expressed not only as a d_{d_j}-dimensional vector space over GF(2), but also as a d_j/d_i-dimensional vector space over GF(2^{d_i}) at the same time.

Example 1. *Assume that* $d_1 = 1$, $d_2 = 2$, $d_3 = 4$. *The field GF*(2^4) *can be expressed as a four-dimensional vector space over GF*(2) *as well as a two-dimensional vector space over GF*(2^2). *Let* α *be a root of the irreducible polynomial* $x^2 + x + 1$ *over GF*(2) *so that GF*(2^2) $= \{0, 1, \alpha, \alpha^2\}$. *The polynomial* $g(x) = x^4 + x + 1$ *is irreducible over GF*(2) *but reducible over GF*(2^2) *and can be factorized as* $g(x) = (x^2 + x + \alpha)(x^2 + x + \alpha^2)$. *Let* β *be a root of the irreducible polynomial* $f(x) = x^2 + x + \alpha$ *over GF*(2^2) *and* β *a root of* $f(x)$, *so that* $g(\beta) = \beta^4 + \beta + 1 = 0$ *as well. Then, every element in* GF(2^4) *can be expressed as* $a_0 + a_1\beta + a_2\beta^2 + a_3\beta^3$ *with* $a_j \in \{0, 1\}$. *Moreover,* $\alpha = \beta^2 + \beta = \beta^5$, *so that* GF($2^2$) $= \{0, \beta^0, \beta^5, \beta^{10}\}$. *Based on this, every element in GF*(2^4) *can also be uniquely expressed as* $b_0 + b_1\beta$, $b_0, b_1 \in$ GF(2^2), *which is summarized in Figure 1. In Figure 1, the integers* 0 *to* 15 *are the decimal representation of the binary* 4*-tuple* (a_3, a_2, a_1, a_0), *e.g.*, 13 *refers to* $1 + \beta^2 + \beta^3$, *which can be expressed* $1 + \alpha\beta$.

b_1 \ b_0	0	1	α	α^2
0	0	1	6	7
1	2	3	4	5
α	12	13	10	11
α^2	14	15	8	9

Figure 1. Every element $a_0 + a_1\beta + a_2\beta^2 + a_3\beta^3, a_j \in \{0, 1\}$ in GF(2^4) has a unique expression in the form of $b_0 + b_1\beta$, $b_0, b_1 \in \{0, 1, \alpha, \alpha^2\} =$ GF(4), where $\alpha^2 + \alpha + 1 = \beta^2 + \beta + \alpha = \beta^4 + \beta + 1 = 0$. The integers 0 to 15 represent the decimal expression of the binary 4-tuple (a_3, a_2, a_1, a_0).

3. Framework Description

3.1. General Principles

In this paper, we focus on the construction of a general scalable RLNC framework over embedded fields, so we attempt to alleviate the influence of specific models of networks. In the course of framework description, we merely classify the nodes in a network into three types: a unique source node, intermediate nodes and receiver nodes. Assume that the source has the highest computational power, so that it can generate coded packets over embedded fields. The intermediate nodes in the network just recode the received data packets over GF(2), so as to fully reduce the overall computational complexities in the network. The heterogeneous receivers have different decoding capabilities. Under its own computational constraint, every receiver can judge whether sufficient coded packets have been received for decoding. More importantly, even though a receiver may not have sufficient computational power to deal with the arithmetic in a larger field over which some received packets are coded, it can still fully utilize these packets instead of directly throwing away in the process of decoding. For instance, assume that two received packets $\mathbf{w}_1$ and $\mathbf{w}_2$ are respectively equal to $\mathbf{p}_1 + \mathbf{p}_2 + \alpha\mathbf{p}_3$ and $\mathbf{p}_2 + \alpha\mathbf{p}_3$, where $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are original packets generated by the source node and α is an element not equal to 0 and 1 in the field GF(2^{d_D}). For the receiver under the strongest field constraint GF(2), the original packet $\mathbf{p}_1$ can be recovered by $\mathbf{w}_1 + \mathbf{w}_2$ instead of directly throwing $\mathbf{w}_1$, $\mathbf{w}_2$ away. Consequently, the proposed scalable RLNC framework not only ensures the decoding capabilities of heterogeneous network devices but also fully reduces the required number of received packets for decoding.

3.2. Encoding and Recoding

In every batch, the source s has n original packets $\mathbf{p}_i, 1 \leq i \leq n$, each of which is an M-dimensional column vector over GF(2), to be transmitted to receivers. Without loss of generality, assume M is divisible by 2^{2^D}, which can be achieved by padding dummy bits into every packet. With increasing D, the double exponentially increasing packet length M may cause the practical issue of an excessive padding overhead. Such an issue can be effectively solved based on the methods proposed in [16,17].

The encoding process at s has two stages. First, based on $\mathbf{p}_i, 1 \leq i \leq n$, for each $1 \leq d \leq D$, extra r_d precoded packets are generated based on coding coefficients selected

from GF(2^{2^d}). In this process, every original packet $\mathbf{p}_i$ is regarded as a vector of $m_d = M/2^d$ symbols, each of which consists of 2^d bits and represents an element in GF(2^{2^d}). The multiplication of $\mathbf{p}_i$ by a coefficient in GF(2^{2^d}) is thus realized by symbol-wise multiplication. Note that when $d_1 < d_2$, the coefficients in GF($2^{2^{d_1}}$) also appear in GF($2^{2^{d_2}}$), but the coding arithmetic changes. The mathematical fundamentals in the previous section guarantee the coding compatibility which will be illustrated in the next example.

Example 2. *Assume* $M = 4$, $n = 2$, $d_1 = 1$ *and* $d_2 = 2$. *Based on two original packets* $\mathbf{p}_1 = [1\,0\,0\,0]^{\mathrm{T}}$ *and* $\mathbf{p}_2 = [1\,1\,0\,1]^{\mathrm{T}}$, *a precoded packet is to be generated over* $\mathrm{GF}(4) = \{0, 1, \alpha, \alpha^2\}$ *by the linear combination* $\alpha\mathbf{p}_1 + \alpha^2\mathbf{p}_2$. *First regard* $\mathbf{p}_1$ *and* $\mathbf{p}_2$ *as vectors of* 2 *symbols over GF*(2^2), *that is,* $\mathbf{p}_1 = \begin{bmatrix}\alpha\\0\end{bmatrix}$ *and* $\mathbf{p}_2 = \begin{bmatrix}\alpha+1\\1\end{bmatrix} = \begin{bmatrix}\alpha^2\\1\end{bmatrix}$. *Then,*

$$\alpha\mathbf{p}_1 + \alpha^2\mathbf{p}_2 = \begin{bmatrix}\alpha^2\\0\end{bmatrix} + \begin{bmatrix}\alpha\\\alpha^2\end{bmatrix} = \begin{bmatrix}1\\\alpha+1\end{bmatrix} = [0\,1\,1\,1]^{\mathrm{T}} \tag{1}$$

According to Figure 1, in GF(2^4), $\alpha = \beta^2 + \beta = \beta^5$ *and* $\alpha^2 = \beta^2 + \beta + 1 = \beta^{10}$. *As every element in GF*(2^4) = *GF*(4^2) *can be uniquely expressed as* $b_0 + b_1\beta$, $b_0, b_1 \in \mathrm{GF}(4)$, *every four-dimensional vector* $[a_3\ a_2\ a_1\ a_0]^{\mathrm{T}}$ *over GF*(2) *as the following element in GF*(16)

$$[a_3\ a_2\ a_1\ a_0]^{\mathrm{T}} = a_3\beta^6 + a_2\beta + a_1\beta^5 + a_0.$$

Based on this rule, $\mathbf{p}_1 = \beta^6$ *and* $\mathbf{p}_2 = \beta^6 + \beta + 1$. *Consequently,* $\beta^5\mathbf{p}_1 + \beta^{10}\mathbf{p}_2 = \beta + \beta^{10} = \beta + \beta^5 + 1$, *which is* $[1\ \alpha+1]^{\mathrm{T}}$ *over GF*(4) *and* $[0\,1\,1\,1]^{\mathrm{T}}$ *over GF*(2), *same as* (1) *obtained by the GF*(4) *arithmetic.*

After stage 1, there are a total of $N = n + r_1 + r_2 + \ldots + r_D$ precoded packets, the first n of which are just the original packets. Let $\mathbf{G} = \begin{bmatrix}\mathbf{I}_n\ \mathbf{A}_1\ \ldots\ \mathbf{A}_D\end{bmatrix}$ denote the $n \times N$ precoding matrix for the N precoded packets, where $\mathbf{I}_n$ refers to the $n \times n$ identity matrix and $\mathbf{A}_d$ is a coefficient matrix defined over GF(2^{2^d}).

In stage 2, every coded packet $\mathbf{c}$ the source finally sends out is a random GF(2)-linear combination of the N precoded packets, that is,

$$\mathbf{c} = [\mathbf{p}_1\ \mathbf{p}_2\ \cdots\ \mathbf{p}_n]\mathbf{G}\mathbf{h},$$

for some randomly generated N-dimensional column vector $\mathbf{h}$ over GF(2), which is referred to as the *coding vector* for packet $\mathbf{c}$. For a *systematic* scheme, the first n coded packets $\mathbf{c}_1, \ldots, \mathbf{c}_n$ transmitted by the source are just n original packets, that is, the coding vector for $\mathbf{c}_j$ is just an N-dimensional unit vector with the j^{th} position equal to 1. Every coded packet will affix its coding vector to its header. In contrast, the information of precoding matrix $\mathbf{G}$ can either be affixed to the header of every packet or presettled to be known at every receiver.

At an intermediate node, the coded packets it transmits are GF(2)-linear combinations of its received packets. Specifically, if an intermediate node receives coded packets $\mathbf{c}_1, \ldots, \mathbf{c}_l$ with respective coding vectors $\mathbf{h}_1, \ldots, \mathbf{h}_l$, then it will recode them to generate a new coded packet $\mathbf{c}'$ to be transmitted as

$$\mathbf{c}' = a_1\mathbf{c}_1 + \ldots + a_l\mathbf{c}_l,$$

where $a_1, \ldots, a_l$ are random binary coefficients. The concomitant coding vector for $\mathbf{c}'$ is $a_1\mathbf{h}_1 + \ldots + a_l\mathbf{h}_l$.

It is worthwhile to note that prior to this work, a flexible RLNC scheme called Fulcrum has been investigated in [13,14]. Fulcrum can be regarded as a special instance in our proposed framework with the setting $D = 3$ and $r_1 = r_2 = 0$.

3.3. Decoding

Define a linear map $\varphi : \mathrm{GF}(2)^N \to \mathrm{GF}(2^{2^D})^n$ by

$$\varphi(\mathbf{v}) = \mathbf{G}\mathbf{v}.$$

for every column vector $\mathbf{v} \in \mathrm{GF}(2)^N$. The notation φ also applies to a set $\mathcal{V}$ of vectors: $\varphi(\mathcal{V}) = \{\varphi(\mathbf{v}) : \mathbf{v} \in \mathcal{V}\}$.

Moreover, let $\mathcal{U}_d$, $0 \le d \le D$, denote the vector subspace of $\mathrm{GF}(2)^N$ spanned by unit vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{\sum_{d'=0}^{d} r_{d'}}$ where a unit vector $\mathbf{u}_j$ refers to an N-dimensional vector with the only nonzero entry at position j.

For a receiver t, assume d_t is the largest field for computation, and m packets have been received. Let $\mathbf{H}$ denote the $N \times m$ matrix over GF(2) obtained by columnwise juxtaposition of the coding vectors of the m received packets, and $\mathcal{H}$ the column space (over GF(2)) of $\mathbf{H}$.

In order to recover original packets under the field constraint $\mathrm{GF}(2^{2^{d_t}})$, we need make use of coding packets with coding vectors in $\mathcal{U}_{d_t} \cap \mathcal{H}$ rather than in $\mathcal{H}$. This is because the lower $\sum_{d'>d_t} r_{d'}$ entries in every coding vector corresponds to the original precoded packets generated by the source over a larger field than $\mathrm{GF}(2^{2^{d_t}})$. We next characterize the following necessary and sufficient condition for decodability at t up to field constraint $\mathrm{GF}(2^{2^{d_t}})$.

Theorem 1. *Based on the m received packets, the original n source packets can be recovered at t if and only if*

$$\dim(\varphi(\mathcal{U}_{d_t} \cap \mathcal{H})) = n. \tag{2}$$

Proof. First assume (2) holds. Then, there must exist n vectors, denoted by $\mathbf{v}_1, \ldots, \mathbf{v}_n$ in $\mathcal{U}_{d_t} \cap \mathcal{H}$ such that

$$\dim(\varphi(\{\mathbf{v}_1, \ldots, \mathbf{v}_n\})) = n. \tag{3}$$

Consequently, there exists an $m \times n$ matrix $\mathbf{K}$ over GF(2) such that $[\mathbf{v}_1 \ \ldots \ \mathbf{v}_n] = \mathbf{HK}$, and (3) implies the full rank n of $\mathbf{GHK}$. As the last $\sum_{d'>d_t} r_{d'}$ rows in $\mathbf{HK}$ are all zero, the elements in $\mathbf{GHK}$ belong to $\mathrm{GF}(2^{2^{d_t}})$, and hence there exists an $n \times n$ matrix $\mathbf{D}$ over $\mathrm{GF}(2^{2^{d_t}})$ subject to $\mathbf{GHKD} = \mathbf{I}_n$, that is, the original packets can be recovered at t.

Next assume that the original n packets can be recovered at t. Then, there exists an $m \times n$ matrix $\mathbf{D}$ over $\mathrm{GF}(2^{2^{d_t}})$ such that $\mathbf{GHD} = \mathbf{I}_n$. Further, $\mathbf{D}$ can be written as $\mathbf{D}_1\mathbf{D}_2$, where $\mathbf{D}_1$, $\mathbf{D}_2$ are over $\mathrm{GF}(2^{2^{d_t}})$ and of respective size $m \times n$ and $n \times n$. Thus, $\mathbf{GHD}_1$ is a matrix over $\mathrm{GF}(2^{2^{d_t}})$ of full rank n. Recall that none of the elements in the last $\sum_{d'>d_t} r_{d'}$ columns in $\mathbf{G}$ is in $\mathrm{GF}(2^{2^{d_t}})$. Thus, every element in $\mathbf{GHD}_1$ belonging to $\mathrm{GF}(2^{2^{d_t}})$ implies that the last $\sum_{d'>d_t} r_{d'}$ rows in $\mathbf{HD}_1$ are all zero. Moreover, as $\mathbf{H}$ is defined over GF(2), we can further deduce that $\mathbf{D}_1$ can be written as $\mathbf{D}_1'\mathbf{D}_1''$ for an $m \times n$ matrix $\mathbf{D}_1'$ over GF(2) and an $n \times n$ matrix $\mathbf{D}_1''$ over $\mathrm{GF}(2^{2^{d_t}})$, such that the last $\sum_{d'>d_t} r_{d'}$ rows in $\mathbf{HD}_1'$ are all zero too, that is, the columns in $\mathbf{HD}_1'$ belong to $\mathcal{U}_{d_t} \cap \mathcal{H}$. In addition, the full rank of $\mathbf{GHD}_1$ implies the full rank of $\mathbf{GHD}_1'$. Equation (2) is thus proved to hold. □

Based on the above theorem, we can further characterize the following equivalent condition for decodability at a receiver from the perspective of matrix rank. For $0 \le d \le D$, denote by $\mathbf{H}_{d_t}$ the $\sum_{d'>d_t} r_{d'} \times m$ submatrix of $\mathbf{H}$ obtained by restricting $\mathbf{H}$ to the last $\sum_{d'>d_t} r_{d'}$ rows.

Corollary 1. *Based on the m received packets, the original n source packets can be recovered at t if and only if*

$$\mathrm{rank}(\mathbf{G}(\mathbf{HK}_{d_t})) = n, \tag{4}$$

where $\mathbf{K}_{d_t}$ is an $m \times (m - \mathrm{rank}(\mathbf{H}_{d_t}))$ matrix whose columns constitute a basis for the kernel of the column space of $\mathbf{H}_{d_t}$ such that $\mathbf{H}_{d_t}\mathbf{K}_{d_t} = \mathbf{0}$.

Note that the column space of $\mathbf{HK}_{d_t}$ are exactly the subspace $\mathcal{U}_{d_t} \cap \mathcal{H}$ in (2), and all entries in the last $\sum_{d'>d_t} r_{d'}$ rows of $\mathbf{HK}_{d_t}$ are zero, so the computation of (4) only involve arithmetic over GF($2^{2^{d_t}}$). Moreover, in order to check (4), it suffices to select rank($\mathbf{HK}_{d_t}$) linearly independent column vectors in $\mathbf{HK}_{d_t}$, juxtapose them into a matrix $\mathbf{H}'$, and check whether rank($\mathbf{GH}'$) $= n$. With the number m of received packets at t increasing, the matrix $\mathbf{K}_{d_t}$ and $\mathbf{H}'$ can be established in the following iterative way.

Algorithm 1. *Denote by $\mathbf{h}^m$ the N-dimensional coding vector over GF(2) for the m^{th} received packet at receiver t. Without loss of generality, assume that there is at least one non-zero entry in $\mathbf{h}^m$. Let $\mathbf{h}^m_{d_t}$ denote the vector restricted from $\mathbf{h}^m$ to the last $\sum_{d'>d_t} r_{d'}$ entries. The next procedure efficiently produces desired $\mathbf{K}_{d_t}$ and $\mathbf{H}'$.*

Initialization. Let $\mathbf{K}_{d_t}$, $\mathbf{H}'$, $\mathbf{B}$ and $\mathbf{B}_{d_t}$ be empty matrices. They are to consist of a m rows, N rows, N rows and $\sum_{d'>d_t} r_{d'}$ rows respectively.

Iteration. Consider the case that the m^{th} packet with coding vector $\mathbf{h}^m$ is just received, and assume receiver t has dealt with the former $m-1$ coding vectors $\mathbf{h}^j, 1 \le j < m$. Perform either of the following two depending on $\mathbf{h}^m_{d_t}$.

- *If $\mathbf{h}^m_{d_t}$ is a zero vector, then update $\mathbf{K}_{d_t}$ as*

$$\mathbf{K}_{d_t} = \begin{bmatrix} \mathbf{K}_{d_t} & \mathbf{0} \\ 0 \dots 0 & 1 \end{bmatrix}, \tag{5}$$

  and respectively append a zero column vector to $\mathbf{B}$ and to $\mathbf{B}_{d_t}$ on the right. Further check whether $\mathbf{h}^m$ is a GF(2)-linear combination of columns in $\mathbf{H}'$. If so, keep $\mathbf{H}'$ unchanged. Otherwise, update $\mathbf{H}'$ as $[\mathbf{H}'\ \mathbf{h}^m]$. The iteration for the current value of m completes.
- *If $\mathbf{h}^m_{d_t}$ is not a zero vector, check whether it is a GF(2)-linear combination of columns in $\mathbf{B}_{d_t}$. If no, respectively update $\mathbf{B}$, $\mathbf{B}_{d_t}$ and $\mathbf{K}_{d_t}$ as*

$$\mathbf{B} = [\mathbf{B}\ \mathbf{h}^m], \mathbf{B}_{d_t} = [\mathbf{B}_{d_t}\ \mathbf{h}^m_{d_t}], \mathbf{K}_{d_t} = \begin{bmatrix} \mathbf{K}_{d_t} \\ 0 \dots 0 \end{bmatrix}, \tag{6}$$

  and the iteration for the current value of m completes. Otherwise, perform the following steps. First compute an $(m-1)$-dimensional vector $\mathbf{k}$ subject to $\mathbf{B}_{d_t}\mathbf{k} = \mathbf{h}^m_{d_t}$, and then update $\mathbf{K}_{d_t}$ as

$$\mathbf{K}_{d_t} = \begin{bmatrix} \mathbf{K}_{d_t} & \mathbf{k} \\ 0 \dots 0 & 1 \end{bmatrix}. \tag{7}$$

  Further compute a new vector $\mathbf{v} = \mathbf{Bk} + \mathbf{h}^m$, and respectively append a zero column vector to $\mathbf{B}$ and to $\mathbf{B}_{d_t}$ on the right. Check whether $\mathbf{v}$ is a GF(2)-linear combination of columns in $\mathbf{H}'$. If so, keep $\mathbf{H}'$ unchanged. Otherwise, update $\mathbf{H}'$ as $[\mathbf{H}'\ \mathbf{v}]$. The iteration for the current value of m completes.

Note that after the above procedure, the sum of the number of nonzero columns in $\mathbf{B}_{d_t}$ and the number of columns in $\mathbf{K}_{d_t}$ is m. The nonzero columns of $\mathbf{B}_{d_t}$ keep to form a basis of the column space of $\mathbf{H}_{d_t} = [\mathbf{h}^1_{d_t} \dots \mathbf{h}^m_{d_t}]$. The columns of $\mathbf{K}_{d_t}$ keep to form a basis of the null space spanned by columns of $\mathbf{H}_{d_t}$. The columns in $\mathbf{H}'$ keep to be a basis of the column space of $\mathbf{HK}_{d_t}$, where $\mathbf{H} = [\mathbf{h}^1 \dots \mathbf{h}^m]$.

Example 3. *Assume that $D = 2$, $n = r_0 = 3$, and $r_1 = r_2 = 1$. The 3×5 precoding matrix $\mathbf{G}$ is designed as*

$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 & \alpha & \beta \\ 0 & 1 & 0 & \alpha^2 & \beta \\ 0 & 0 & 1 & 1 & \beta \end{bmatrix}$$

where β is a primitive element in GF(2^4) and $\alpha = \beta^5$, which can be regarded as a primitive element of GF(2^2) $\subset$ GF(2^4).

Assume that at a receiver t, GF(2^2) is the largest field for computation, and 4 packets have been received with the columnwise juxtaposition of the respective coding vectors prescribed by

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

As $\mathbf{H}_{d_t} = [1\ 1\ 1\ 1]$ herein, the aforementioned iterative approach can yield the following $\mathbf{K}_{d_t}$ and concomitant $\mathbf{H}'$:

$$\mathbf{K}_{d_t} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \mathbf{H}' = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},$$

where the columns of $\mathbf{H}'$ form a basis for the subspace $\mathcal{U}_{d_t} \cap \mathcal{H}$. Consequently, $\mathbf{GH}' = \begin{bmatrix} 1 & 1 & \alpha \\ 1 & 0 & 1+\alpha^2 \\ 0 & 1 & 0 \end{bmatrix}$. Since $1 + \alpha + \alpha^2 = 0$ in GF(2^2), $\mathrm{rank}(\mathbf{GH}') = 2$, that is, (4) does not hold. Therefore, the receiver requires to receive more packets before decoding all original packets.

Assume $\mathbf{h}^5 = [1\ 0\ 0\ 1\ 1]^{\mathrm{T}}$ is the coding vector for the 5^{th} received packet. Then, the matrix $\mathbf{K}_{d_t}$ is dynamically updated to

$$\mathbf{K}_{d_t} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

but there is no change for $\mathbf{H}'$, because $\mathbf{H} \cdot [1\ 0\ 0\ 0\ 1]^{\mathrm{T}}$ belongs to the column space of $\mathbf{H}'$.

Assume $\mathbf{h}^6 = [0\ 0\ 1\ 0\ 0]^{\mathrm{T}}$ is the coding vector for the 6^{th} received packet. First, dynamically update $\mathbf{K}_{d_t}$ to

$$\mathbf{K}_{d_t} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

Then, as $\mathbf{h}^6 = [0\ 0\ 1\ 0\ 0]^{\mathrm{T}}$ does not belong to the column space of $\mathbf{H}'$, update $\mathbf{H}'$ as $[\mathbf{H}'\ \mathbf{h}^6]$:

$$\mathbf{H}' = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$

Consequently, $\mathbf{GH}' = \begin{bmatrix} 1 & 1 & \alpha & 0 \\ 1 & 0 & 1+\alpha^2 & 0 \\ 0 & 1 & 0 & 1 \end{bmatrix}$, and it has full rank 3, so the receiver can recover the source packets. Actually, in this case, the source packets can be recovered by merely GF(2)-based operations.

In two special cases that $d_t = D$ and $d_t = 0$, *i.e.*, receiver t has the highest and the lowest computational power respectively, (4) degenerates to a more concise form.

Corollary 2. *When $d_t = D$, (4) is equivalent to*

$$\mathrm{rank}(\mathbf{GH}) = n. \tag{8}$$

When $d_t = 0$, (4) is equivalent to

$$\text{rank}(\mathbf{H}) - \text{rank}(\mathbf{H}_{d_t}) = n. \tag{9}$$

Recall that Fulcrum [13,14] can be regarded as a special RLNC scheme of our framework. One may notice that in Fulcrum, the decoding rule over GF(2) at a receiver is

$$\text{rank}(\mathbf{H}) = N, \tag{10}$$

which is sufficient but not necessary. In contrast, (9) is both necessary and sufficient. As to be seen in Section 5, there is an observable performance gain when (9) is adopted as the decoding rule instead of (10). Moreover, our proposed scalable RLNC is more flexible than Fulcrum, because the receivers with intermediate computational power can fully utilize its decoding capability to decode over intermediate fields (rather than only over GF(2)), so that the number of required coded packets can be reduced.

3.4. Decoding Complexity Analysis

In this subsection, we briefly analyze the computational complexity of the proposed scalable RLNC scheme at receiver t with the field constraint GF(2^{d_t}). We assume that after a sufficiently large recoding process over GF(2), the last r positions in every received binary column vector $\mathbf{h}$, which corresponds to the r precoded packets generated over the larger fields than GF(2), are nonzero. According to Corollary 1, when enough coded packets have been received such that the condition

$$\text{rank}(\mathbf{G}(\mathbf{H}\mathbf{K}_{d_t})) = n$$

is satisfied, receiver t can recover all original packets by linear combining n coded packets over GF(2^{d_t}). Accordingly, it requires at most $n^2 M/d_t$ multiplications and $n(n-1)M/d_t$ additions over GF(2^{d_t}) in the decoding process. Following the same consideration in [4,18,19], we assume that it respectively takes d_t and $2d_t^2$ binary operations to realize addition and multiplication between two elements in GF(2^{d_t}). Consequently, the total number of required binary operations can be characterized as $\mathcal{O}(Mnd_t)$ to recover every M-bit original packet.

Herein, we did not consider the complexity to compute the inverse matrix of $\mathbf{G}\mathbf{H}\mathbf{K}_{d_t}$ because in practice the packet length M is much larger than n, and this convention has also been adopted in [4,19] for computational complexity analysis.

4. Optimal Construction of Precoding Matrix G

Based on the analysis in the previous section, we are motivated to carefully design such a precoding matrix $\mathbf{G}$ that the full rank of $\mathbf{H}$ is equivalent to the full rank of $\mathbf{GH}$, which can optimize the decodability performance for fixed parameters n and N. To achieve this goal, for the precoding matrix $\mathbf{G}$, we first introduce the following condition that is stronger than the conventional maximal distance separable (MDS) property.

Definition 1. *An $n \times N$ matrix $\mathbf{G}$ over GF(2^{2^D}) is said to be* MDS under GF(2)-mapping *if for any full-rank $N \times n$ matrix $\mathbf{H}$ over GF(2),* $\text{rank}(\mathbf{GH}) = n$.

Recall that if $\mathbf{G}$ satisfies the conventional MDS property, all n columns in it are linearly independent. Obviously, the conventional MDS property is a prerequisite for the proposed MDS property under GF(2)-mapping. However, Example 3 demonstrates an MDS matrix $\mathbf{G}$ that is not MDS under GF(2)-mapping. To the best of our knowledge, except for a brief attempt in [13], there is no prior literature involving the construction of a matrix satisfying the MDS property under GF(2)-mapping. We next characterize an equivalent condition on

the MDS property under GF(2)-mapping, so as to facilitate the explicit construction. Given an $n \times N$ matrix $\mathbf{G}$, let $\mathcal{C}$ denote the set of row vectors generated by $\mathbf{G}$:

$$\mathcal{C} = \{\mathbf{mG} : \mathbf{m} \in \mathrm{GF}(2^{2^D})^n\}. \tag{11}$$

For every $\mathbf{c} \in \mathcal{C}$, let $\mathcal{N}_\mathbf{c}$ denote its null space in $\mathrm{GF}(2^{2^D})^N$.

Theorem 2. *An $n \times N$ matrix $\mathbf{G}$ is MDS under GF(2)-mapping if and only if*

$$\dim(\mathcal{N}_\mathbf{c} \cap \mathrm{GF}(2)^N) < n, \ \forall\, \mathbf{c} \in \mathcal{C} \backslash \{\mathbf{0}\} \tag{12}$$

Proof. We prove the theorem in a contrapositive argument. Assume that there exists a nonzero $\mathbf{c} \in \mathcal{C}$ such that $\dim(\mathcal{N}_\mathbf{c} \cap \mathrm{GF}(2)^N) \geq n$, and let $\mathbf{m}$ be a row vector over $\mathrm{GF}(2^{2^D})$ satisfying $\mathbf{c} = \mathbf{mG}$. Then, we can select n linearly independent column vectors $\mathbf{h}_1, \ldots, \mathbf{h}_n$ over GF(2) from $\mathcal{N}_\mathbf{c}$. Write $\mathbf{H} = [\mathbf{h}_1 \ldots \mathbf{h}_n]$. Thus, $\mathbf{mGH} = \mathbf{cH} = \mathbf{0}$, so that $\mathbf{GH}$ is not full rank n, *i.e.*, $\mathbf{G}$ is not MDS under GF(2)-mapping.

Assume that $\mathbf{G}$ is not MDS under GF(2)-mapping, and let $\mathbf{H}$ be a full rank $N \times n$ matrix over GF(2) subject to $\mathrm{rank}(\mathbf{GH}) < n$. Then, there exists an n-dimensional row vector $\mathbf{m}$ such that $\mathbf{mGH} = \mathbf{0}$. Write $\mathbf{c} = \mathbf{mG}$, so that $\mathbf{cH} = \mathbf{0}$. Since $\mathbf{H}$ is full rank n, there are at least n linearly independent vectors (which are the columns of $\mathbf{H}$) belonging to $\mathcal{N}_\mathbf{c}$, *i.e.*, $\dim(\mathcal{N}_\mathbf{c} \cap \mathrm{GF}(2)^N) \geq n$. □

For $\mathbf{c} \in \mathcal{C}$, let $\eta(\mathbf{c})$ denote the number of elements in $\mathbf{c}$ belonging to $\mathrm{GF}(2^{2^D}) \backslash \{0, 1\}$, and define an indicator δ which is set to 1 if $\mathbf{c}$ consists of an element equal to 1 and set to 0 otherwise. The following is a useful corollary of Theorem 2.

Corollary 3. *If an $n \times N$ matrix $\mathbf{G}$ is MDS under GF(2)-mapping, then the followings hold*

$$\eta(\mathbf{c}) + \delta > N - n, \ \forall\, \mathbf{c} \in \mathcal{C} \backslash \{\mathbf{0}\}. \tag{13}$$

$$\mathcal{C} \cap \mathrm{GF}(2)^N = \{\mathbf{0}\}. \tag{14}$$

Proof. Assume there is a nonzero $\mathbf{c} \in \mathcal{C}$ with $\eta(\mathbf{c}) + \delta \leq N - n$, *i.e.*, $N - \eta(\mathbf{c}) \geq n + \delta$. Define a new vector $\mathbf{c}'$ by restricting to its components belonging to GF(2), so that the dimension of $\mathbf{c}'$ is $N - \eta(\mathbf{c})$. Thus, the dimension of the null space of $\mathbf{c}'$ in $\mathrm{GF}(2)^{N-\eta(\mathbf{c})}$ is $N - \eta(\mathbf{c}) - \delta$, which is no smaller than n. Correspondingly, $\dim(\mathcal{N}_\mathbf{c} \cap \mathrm{GF}(2)^N) \geq n$, a contradiction to the MDS property under GF(2)-mapping for $\mathbf{G}$ according to (12).

If there is a nonzero $\mathbf{c} \in \mathcal{C}$ belonging to $\mathrm{GF}(2)^N$, then $\eta(\mathbf{c}) = 0$ so that (13) cannot hold as $N > n$, and thus $\mathbf{G}$ cannot be MDS under GF(2)-mapping. □

Conditions (13) and (14) are insufficient for the MDS property under GF(2)-mapping. The key reason is the possibility of the following

$$\sum\nolimits_{<j>} \alpha_j \in \mathrm{GF}(2), \ \alpha_j \in \mathrm{GF}(2^{2^D}) \backslash \{0, 1\}. \tag{15}$$

For this reason, we should pay more attention in the matrix design to avoid the involvement of those elements in (15). The special case $N = n + 1$ is easier to manipulate.

Proposition 1. *When $N = n + 1$, an $n \times N$ matrix $\mathbf{G}$ is MDS under GF(2)-mapping if and only if (14) holds.*

Proof. The necessity has been shown in Corollary 3. To prove sufficiency, assume (14) holds for $\mathcal{C}$ defined in (11) based on $\mathbf{G}$. Let $\mathbf{c}$ be an arbitrary vector in $\mathcal{C}$. As (14) holds, $\eta(\mathbf{c}) > 0$. In the case $\eta(\mathbf{c}) = 1$, there must be at least one element in $\mathbf{c}$ equal to 1, because otherwise we can find another vector in $\mathcal{C}$ with all elements in GF(2), a contradiction to (14). Thus, $\dim(\mathcal{N}_\mathbf{c} \cap \mathrm{GF}(2)^N) < n$ for this case. Consider the case $\eta(\mathbf{c}) \geq 2$. Without loss of

generality, write $\mathbf{c} = [c_1 \ \dots \ c_{\eta(\mathbf{c})} \ 0 \ \dots \ 0]$ with $c_j \neq 0$. We can assume c_j not all identical, because otherwise we can again find another vector in $\mathcal{C}$ with all elements in GF(2), a contradiction to (14). Moreover, for arbitrary two elements $a, b \in \mathrm{GF}(2^{2^D})$, $a + b = 0$ if and only if $a = b$. Hence, there are at most $\eta(\mathbf{c}) - 2$ linearly independent vectors in $\mathrm{GF}(2)^{\eta(\mathbf{c})}$ that are in the null space of $\mathbf{c}$, which further implies $\dim(\mathcal{N}_\mathbf{c} \cap \mathrm{GF}(2)^N) < n$. We have proved (12) and thus the considered $\mathbf{G}$ is MDS under GF(2)-mapping. □

Corollary 4. *When $N = n + 1$, there exists a systematic $n \times N$ matrix $\mathbf{G} = [\mathbf{I}_n \ \mathbf{A}_D]$ over GF(2^{2^D}) that is MDS under GF(2)-mapping if and only if $n < 2^D$.*

Proof. Assume $n < 2^D$. Define an n-dimensional column vector $\mathbf{a} = [\alpha, \alpha^2, \dots, \alpha^n]^\mathrm{T}$, where α is a primitive element of $\mathrm{GF}(2^{2^D})$. In this way, all elements in $\mathbf{a}$ are distinct and every GF(2)-combination $\sum_{1 \leq j \leq n} a_j \alpha^j$ among them does not belong to GF(2). By Proposition 1, $[\mathbf{I}_n \ \mathbf{a}]$ is an MDS matrix under GF(2)-mapping. When $n \geq 2^D$, let $\mathbf{a} = [\alpha_1, \dots, \alpha_n]^\mathrm{T}$ be an arbitrary n-dimensional vector in $\mathrm{GF}(2^{2^D})$. In order to make $[\mathbf{I}_n \ \mathbf{a}]$ MDS under GF(2)-mapping, according to (14) in Corollary 3, there is not any element α_j belonging to GF(2). If there is a basis, say $\{\alpha_1, \dots, \alpha_{2^D}\}$ of $\mathrm{GF}(2^{2^D})$ in $\mathbf{a}$, then 1 can be written as a GF(2)-linear combination of the basis, so that (14) does not hold. If there is not a basis of $\mathrm{GF}(2^{2^D})$ in $\mathbf{a}$, then there exists an n-dimensional nonzero row vector $\mathbf{v}$ over GF(2) subject to $\mathbf{va} = 0$, so that (14) does not hold either. Thus, it is impossible for $[\mathbf{I}_n \ \mathbf{a}]$ to be MDS under GF(2)-mapping. □

Based on the above corollary, the required field size is exponentially larger than N in the construction of an $n \times N$ systematic MDS matrix under GF(2)-mapping. This implies that it is infeasible to construct such a practical precoding matrix $\mathbf{G}$ for large N. For this reason, it is alternative to choose to randomly generate $\mathbf{G}$, which may cause a near-optimal decodability behavior as illustrated in the next example.

Example 4. *Define the following vectors $\mathbf{a}_1 = [\alpha \ \alpha^2 \ \alpha^3 \ \dots \ \alpha^7]^\mathrm{T}$ and $\mathbf{a}_2 = [\alpha^2 \ \alpha^4 \ \alpha^6 \ \dots \ \alpha^{14}]^\mathrm{T}$ over GF(2^8) in which α is a primitive element. It can be checked that both matrices $[\mathbf{I}_7 \ \mathbf{a}_1]$ and $[\mathbf{I}_7 \ \mathbf{a}_2]$ are MDS under GF(2)-mapping. Although the 7×9 matrix $\mathbf{G} = [\mathbf{I}_7 \ \mathbf{a}_1 \ \mathbf{a}_2]$ is not MDS under GF(2)-mapping, among 42435 7-dimensional subspaces of $\mathrm{GF}(2)^9$, there are only 127 instances to break the desired MDS property, that is, every basis for each of the instances forms a 9×7 matrix $\mathbf{H}$ with $\mathrm{rank}(\mathbf{GH}) < 7$.*

5. Numerical Analysis

In this section, we numerically analyze the performance of applying the proposed systematic scalable RLNC scheme to a wireless broadcast network, which is a classical model to demonstrate the advantage of RLNC [4–7]. The number n of original packets in a batch is varied from $n = 6$ to 24. In every timeslot, the source broadcasts one packet to all receivers. The memoryless and independent packet loss probability for every receiver is $p_e = 0.2$, that is, in every timeslot, every receiver can successfully receive a packet with probability $1 - p_e$. We consider the scheme with parameters $D = 2, r = 2$ where $r_1 = r_2 = 1$. In the $n \times N$ precoding matrix $\mathbf{G} = [\mathbf{I}_n \ \mathbf{A}_1 \ \mathbf{A}_2]$, the entries in $\mathbf{A}_1$ and $\mathbf{A}_2$ are randomly selected from $\mathrm{GF}(2^2)$ and $\mathrm{GF}(2^4)$, respectively. In the numerical analysis of scalable RLNC, the single source s has n original packets to be broadcast to a total of 30 receivers with different decoding capabilities. Specifically, the 30 receivers fall into 3 different groups and the 10 receivers in every group has the same decoding capability, and can decode based on the decoding rule (4) over GF(2), $\mathrm{GF}(2^2)$ and $\mathrm{GF}(2^4)$, respectively. In the first n timeslots, the source broadcasts n original packets, whose coding vectors are $(n + r)$-dimensional unit vectors, to all receivers. Starting from timeslot $n + 1$, the source broadcasts coded packets, each of which is generated based on a random N-dimensional column vector $\mathbf{h}$ over GF(2), till all the receivers can recover the n original packets. Herein, for every parameter setting and every considered RLNC scheme, we conduct 1200 independent rounds of simulation which result in 95% confidence intervals.

Figure 2 depicts the average group completion delay per packet for the 3 groups of receivers, respectively labeled as "Scalable-GF(2^x)", $x \in \{1, 2, 4\}$ of the considered scalable RLNC scheme. The group completion delay means the number of extra coded packets the source broadcasts till all the 10 receivers in the group can recover n original packets. For a better comparison, the figure also depicts the average group completion delay per packet, labeled as "RLNC-GF(2^x)", for a group of 10 receivers of three *different* classical systematic RLNC schemes over different fields GF(2^x), $x \in \{1, 2, 4\}$. Recall that in the classical systematic RLNC scheme over GF(2^x), the source first broadcasts n original packets and then randomly coded packets with n-dimensional coding vectors over GF(2^x). One may observe from Figure 2 that for the case of GF(2^4), the average completion delay of scalable RLNC is almost same as the classical RLNC. Over other smaller fields, even though scalable RLNC yields higher average completion delay than classical RLNC, it simultaneously guarantees the decoding compatibility at heterogeneous receivers, which cannot be endowed by classical RLNC schemes. For instance, assume that the source adopts classical RLNC over GF(2^2) to generate coded packets. On one hand, the group of receivers with decoding capability constrained to GF(2) will fail to recover the original packets. On the other hand, the groups of receivers with decoding capability over GF(2^4) cannot fully utilize their higher computational power so that the average completion delay cannot be further reduced compared with decoding over GF(2^2). As a result, the performance loss for the cases of smaller fields in our proposed scalable RLNC compared to classical RLNC is the cost of decoding compatibility over different fields.

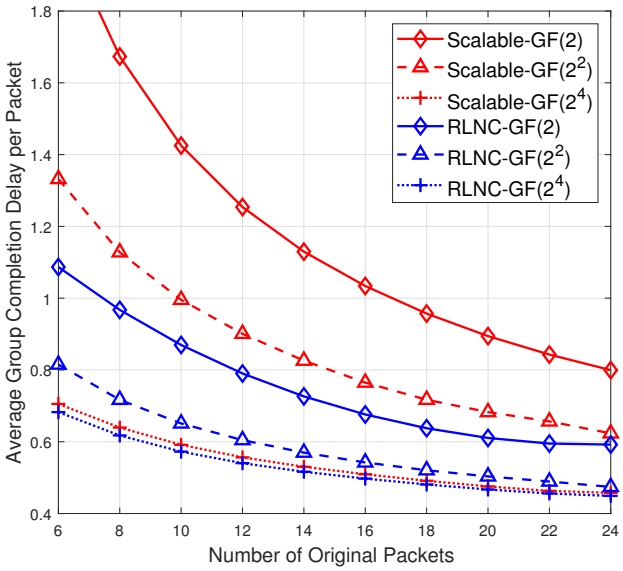

Figure 2. The average group completion delay per packet for the receivers of different systematic RLNC schemes in a wireless broadcast network with $r_1 = r_2 = 1$ and packet loss probability $p_e = 0.2$.

For the considered systematic scalable RLNC scheme, recall that for decoding over GF(2) in the proposed scalable RLNC, Equation (9) obtained in Sec. III is a necessary and sufficient rule while Equation (10), originally adopted in [13,14] for Fulcrum decoding, is a non-necessary rule. Figure 3 compares the average group completion delay per packet for 10 receivers as well as the average completion delay per packet at a single receiver when the receivers adopt different decoding rules (9) and (10) over GF(2). For the average completion delay at a single receiver, a noticeable performance gain can be observed. In particular, when the number of original packets is less than 10, the average completion delay at a single receiver is reduced by more than 20% based on the decoding rule (9) instead of (10). For the average group completion delay, the performance gain by adopting (9) instead of (10) becomes less obvious because it is offset by the increasing number of receivers in a group. Compared with Fulcrum, which only supports decoding over the smallest field GF(2) or

the largest field GF(2^{2^D}), in addition to the performance gain illustrated in Figure 3, our proposed scalable RLNC is more flexible. This is because the receivers with intermediate computational power can fully utilize its decoding capability to decode over intermediate fields (rather than only over GF(2)), so that the average completion delay can be reduced.

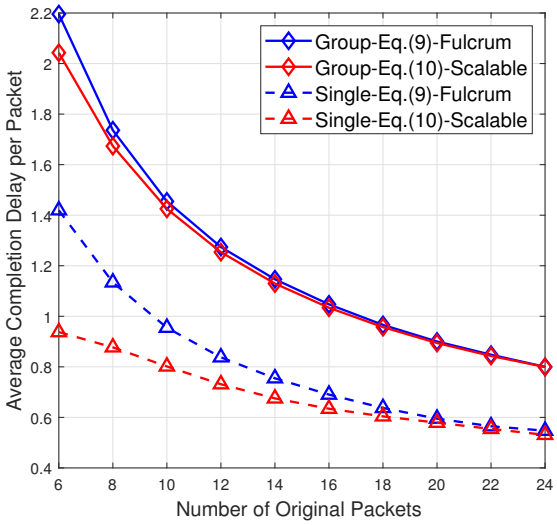

Figure 3. The average group completion delay per packet for 10 receivers as well as the average completion delay per packet at a single receiver when the receivers adopt different decoding rules (9) and (10) over GF(2).

In the remaining part of this section, we shall analyze the performance of our scalable RLNC scheme by adjusting the *sparsity* $0 < P_h < 1$ of $\mathbf{h}$, which controls to the probability for every component in $\mathbf{h}$ to be one. Specifically, for every packet to be transmitted by the source, the expected number of precoded packets to form it is $P_h(n+r)$. In previous analysis of this section, P_h is set to $1/2$. We next consider a more *sparse* $\mathbf{h}$ with $P_h \leq 1/2$.

According to the work in [20], given that there are $(i-1)$ $(n+r)$-dimensional linearly independent binary vectors with sparsity P_h, the probability that a new randomly generated $(n+r)$-dimensional binary coding vector $\mathbf{h}^i$ with sparsity P_h is linearly independent with them is lower bounded by

$$1-(1-P_h)^{n+r-i}. \tag{16}$$

This bound indicates that except for the case i close to $(n+r)$, the lower bound keeps very close to 1. Further, at the end of Sec. IV, we have illustrated that a random $\mathbf{G}$ will bring a near-optimal decodability behavior, that is, the full rank of $\mathbf{H}$ will lead to the full rank of $\mathbf{GH}$ with high probability. As a result, although our proposed scalable RLNC scheme with two-stage encoding process is different from the conventional sparse RLNC described in [20], we are motivated to bring the *sparsity* into our proposed scheme and attempt to meet a balance between completion delay and decoding complexity. The work in [14] has taken the sparsity into consideration in their performance analysis of Fulcrum, which is a special instance of our proposed scalable RLNC scheme.

In simulation, besides the consideration of sparsity P_h, we also extend the value range of n from $[6,24]$ to $[8,64]$ and set $r_1 = r_2 = 2$. All other parameter settings are same as those in Figure 2. The 3 solid curves in Figure 4 illustrate the average group completion delay per packet for the 3 groups of 10 receivers under different field constraints GF(2), GF(2^2) and GF(2^4) for scalable RLNC with sparsity $P_h = 1/2$. The 3 dotted curves in Figure 4 illustrate the completion delay performance under different field constraints GF(2), GF(2^2) and GF(2^4) for scalable RLNC with $P_h = 1/4$. It is interesting to observe that with the batch size n increasing, under the same decoding constraint (*i.e.*, two curves in the same color), the completion delay performance for the case $P_h = 1/4$ will converge to the case $P_h = 1/2$. This result indicates that the lower bound in (16) is rather loose when i is close to $n+r$,

and moreover, for a large enough batch size n, a more sparse vector $\mathbf{h}$ will not affect the completion delay performance much in a wireless broadcast network.

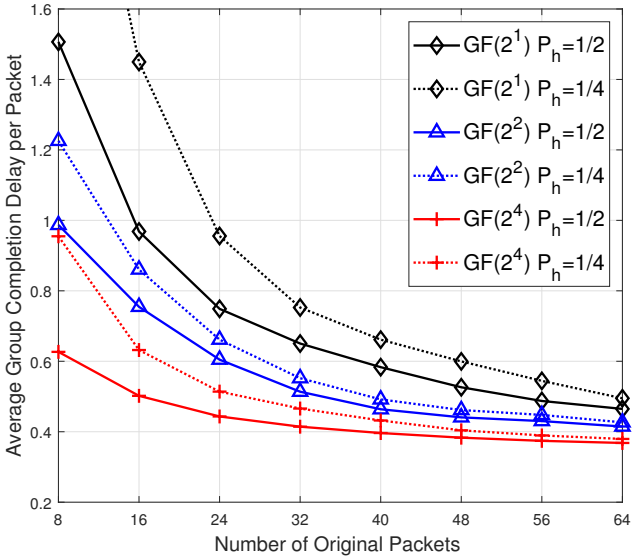

Figure 4. The average group completion delay per packet for scalable RLNC with different sparsity P_h.

6. Conclusions

In this work, the proposed scalable RLNC framework based on embedded fields aims at endowing heterogeneous receivers with different decoding capabilities in complex network environments. In this framework, we derive a general decodability condition by the arithmetic compatibility of embedded fields. Moreover, we theoretically study the specific construction of an optimal precoding matrix $\mathbf{G}$ and illustrate the rationality of the near-optimal behavior of a randomly generated $\mathbf{G}$.

In numerical analysis, we demonstrate that the proposed scalable RLNC not only guarantees a better decoding compatibility compared with classical RLNC, but also provides a better decoding performance over GF(2) in terms of smaller average completion delay compared with Fulcrum. In addition, the numerical analysis also demonstrates that for a large enough batch size, the sparsity of the vector $\mathbf{h}$ does not affect the completion delay performance much. As a potential future work, the theoretical insight behind this observation deserves a further investigation so as to facilitate the design of a scalable RLNC scheme with a better tradeoff between decoding complexity and completion delay.

Last, the present scalable RLNC framework assumes block-based coding. It would also be interesting to make use of the embedded fields structure to generalize the design of sliding window-based random linear coding schemes such as the ones studied in [21–23].

Author Contributions: R.Z. and Q.S. conceived and designed the mathematical model. H.T. designed the whole coding framework and wrote the paper with the help of Q.S., K.L. and Z.L. All authors were involved in problem formulation, data analysis and editing of this paper. All authors have agreed to the published version of the manuscript.

Funding: This work was partially supported by the National Natural Science Foundation of China under Grant 62101028 and 62271044, and by China Postdoctoral Science Foundation under Grant 2021TQ0031, and by Huawei TC20211126644 and by China Telecom 20222910016.

Acknowledgments: This paper was partly presented in [24] at the IEEE/CIC International Conference on Communications in China (ICCC) 2021.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Ho, T.; Médard, M.; Koetter, R.; Karger, D.R.; Effros, M.; Shi, J.; Leong, B. A random linear network coding approach to multicast network. *IEEE Trans. Inf. Theory* **2006**, *52*, 4413–4430. [CrossRef]
2. Huang, J.; Gharavi, H.; Yan, H.; Xing, C.C. Network coding in relay-based device-to-device communications. *IEEE Netw.* **2017**, *31*, 102–107. [CrossRef] [PubMed]
3. Asterjadhi, A.; Fasolo, E.; Rossi, M.; Widmer, J.; Zorzi, M. Toward network coding-based protocols for data broadcasting in wireless ad hoc networks. *IEEE Trans. Wirel. Commun.* **2010**, *9*, 662–673. [CrossRef]
4. Su, R.; Sun Q.; Zhang Z. Delay-complexity trade-off of random linear network coding in wireless broadcast. *IEEE Trans. Commun.* **2020**, *68*, 5606–5618. [CrossRef]
5. Eryilmaz, A.; Ozdaglar, A.; Médard, M.; Ahmed, E. On the delay and throughput gains of coding in unreliable networks. *IEEE Trans. Inf. Theory* **2008**, *54*, 5511–5524. [CrossRef]
6. Swapna, B.T.; Eryilmaz, A.; Shroff, N.B. Throughput-delay analysis of random linear network coding for wireless broadcasting. *IEEE Trans. Inf. Theory* **2013**, *59*, 6328–6341. [CrossRef]
7. Zhu, H.; Ouahada, K. Investigating random linear coding from a pricing perspective. *Entropy* **2018**, *20*, 548. [CrossRef] [PubMed]
8. Wunderlich, S.; Cabrera, J.A.; Fitzek, F.H.; Reisslein, M. Network coding in heterogeneous multicore IoT nodes with DAG scheduling of parallel matrix block operations. *IEEE Internet Things J.* **2017**, *4*, 917–933. [CrossRef]
9. Heide, J.; Lucani, D.E. Composite extension finite fields for low overhead Network Coding: Telescopic codes. In Proceedings of the 2015 IEEE International Conference on Communications (ICC), London, UK, 8–12 June 2015.
10. Marcano, N.J.H.; Heide, J.; Lucani, D.E.; Fitzek, F.H. On the overhead of telescopic codes in network coded cooperation. In Proceedings of the 2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall), Boston, MA, USA, 6–9 September 2015.
11. Heide, J. Composite extension finite fields for distributed storage erasure coding. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 22–27 May 2016.
12. Yazdani, V.; Lucani, D. Revolving codes: Overhead and computational complexity analysis. *IEEE Commu. Lett.* **2021**, *25*, 374–378. [CrossRef]
13. Lucani, D.E.; Pedersen, M.V.; Ruano, D.; Sørensen, C.W.; Fitzek, F.H.; Heide, J.; Geil, O.; Nguyen, V.; Reisslein, M. Fulcrum: Flexible network coding for heterogeneous devices. *IEEE Access* **2018**, *6*, 77890–77910. [CrossRef]
14. Nguyen, V.; Tasdemir, E.; Nguyen, G.T.; Lucani, D.E.; Fitzek, F.H.; Reisslein, M. DSEP Fulcrum: Dynamic sparsity and expansion packets for fulcrum network coding. *IEEE Access* **2020**, *8*, 78239–78314. [CrossRef]
15. Lidl, R.; Niederreiter, H. *Finite Fields*, 3rd ed.; Cambridge University Press: Cambridge, UK, 1997.
16. Schutz, B.; Aschenbruck, N. Packet-preserving network coding schemes for padding overhead reduction. In Proceedings of the 2019 IEEE 44th Conference on Local Computer Networks (LCN), Osnabrueck, Germany, 14–17 October 2019.
17. Taghouti, M.; Lucani, D.E.; Cabrera, J.A.; Reisslein, M.; Pedersen, M.V.; Fitzek, F.H. Reduction of padding overhead for RLNC media distribution with variable size packets. *IEEE Trans. Broadcast.* **2019**, *65*, 558–576. [CrossRef]
18. Tang, H.; Sun, Q.T.; Li, Z.; Yang, X.; Long, K. Circular-shift linear network coding. *IEEE Trans. Inf. Theory* **2019**, *65*, 65–80. [CrossRef]
19. Hou, H.; Shum, K.W.; Chen, M.; Li, H. BASIC codes: Low-complexity regenerating codes for distributed storage systems. *IEEE Trans. Inf. Theory* **2016**, *62*, 3053–3069. [CrossRef]
20. Feizi, S.; Lucani, D.E.; Sørensen, C.W.; Makhdoumi, A.; Médard, M. Tunable Sparse Network Coding for Multicast Networks. In Proceedings of the 2014 IEEE International Symposium on Network Coding (NetCod), Aalborg Oest, Denmark, 27–28 June 2014; pp. 1–6.
21. Karetsi, F.; Papapetrou, E. Lightweight network-coded ARQ: An approach for ultra-reliable low latency communication. *Comput. Commun.* **2022**, *185*, 118–129. [CrossRef]
22. Ma, S.; Liu, X.; Yan, Y.; Zhang, B.; Zheng, J. Sliding-window based batch forwarding using intra-flow random linear network coding. In Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC), Limassol, Cyprus, 15–19 June 2020.
23. Tasdemir, E.; Nguyen, V.; Nguyen, G.T.; Fitzek, F.H.; Reisslein, M. FSW: Fulcrum sliding window coding for low-latency communication. *IEEE Access* **2022**, *10*, 54276–54290. [CrossRef]
24. Tang, H.; Zheng, R.; Li, Z.; Sun, Q.T. Scalable Network Coding over Embedded Fields. In Proceedings of the 2021 IEEE/CIC International Conference on Communications in China (ICCC), Xiamen, China, 28–30 July 2021; pp. 641–646.