Chapter 6: Modified Earliest Decoding for RLNC

Contents

In this chapter we present a practical network configuration in order to obtain generator matrices of a lower triangular structure at receiver nodes in an RLNC network. This lower triangular structure enables the implementation of earliest decoding which offers low decoding delay. Further, we present a decoding method adapted from earliest decoding that offers better coding efficiency in this practical network configuration.

6.1 Introduction

One of the disadvantages of using Gaussian elimination (GE) for decoding in RLNC networks is high computational complexity and delay, which can outweigh the practical advantages that RLNC offers, especially for large blocks of information [**8**], [**10**], [**31**]. Thus, the effectiveness of RLNC in a practical setting is dependent on the complexity and decoding delay of the decoding algorithm that is implemented [**40**].

In [**8**] a decoding method was proposed to improve the decoding delay in an RLNC network. Earliest decoding (ED) is a more cost-effective decoding method as it decodes the source data in smaller subsections when innovative packets are received. Moreover, it employs GE on a subset of the linearly independent received packets when they have sufficient rank. The decoding delay of earliest decoding is approximately constant and independent of the number of source symbols in a generation [**8**], [**39**], [**49**].

The efficiency of ED is, however, dependent on the specific lower triangular structure of the encoded packets obtained by a receiver node. In the literature is which ED has been presented and implemented [**8**], [**39**], it was merely stated that the lower triangular form of the received packets is due to network causality. To the best of our knowledge, no coding method or network configuration has been presented on how this lower triangular structure was obtained.

Our first contribution in this chapter is to present a practical network configuration to obtain the lower triangular structure of encoded packets, so that ED can be successfully implemented. We show how to obtain the lower triangular structure of the generator matrix without altering the encoding algorithm at intermediate network nodes. We illustrate the way in which the causality of the random network can render packets with a lower triangular structure if the source transmits packets in a defined manner. The structure of the received lower triangular generator matrix is studied and a mathematical model developed. This mathematical model will be used in Chapter 6 for calculations on decoding delay and complexity.

Through the construction and study of this network configuration, we recognised this network configuration as an ideal platform for further improvements in decoding. The second contribution in this chapter is the presentation and evaluation of a decoding method called Modified Earliest Decoding (MED). MED is based on ED, but has been adapted to implement the iterative decoding principal of belief propagation (BP). In a network environment where packets are roughly lower triangular, and as compared to ED, MED can improve the decoding delay.

Our aim in this chapter is to construct and implement low complexity decoding at receiver nodes where all the nodes in the network perform RLNC. This means that all the packets received from intermediate nodes are randomly encoded.

6.2 Network model

We consider a network scenario that can be represented by a directed acyclic graph $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$, as presented in Chapter 2. As previously discussed, the data at the source node $s \in \mathcal{V}$ contains n source symbols $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n] \in \mathbb{F}_q$. These symbols are transmitted over the outgoing edges, randomly encoded over $\mathbb{F}_2$ by the intermediate nodes and collected by the receiver nodes $Z = \{z_1, \dots, z_{|z|}\}, Z \subset \mathcal{V}$. The received encoded packets $y(e)$ can be written as linear combinations of the source symbols

$$y(e) = \sum_{i=1}^{n} g_i(e)\, \boldsymbol{x}_i. \tag{6.1}$$

The global encoding vector $\boldsymbol{g}(e) = [g_1(e), g_2(e), \dots, g_n(e)]$ included in each packet $y(e)$ forms a row vector of the generator matrix $\boldsymbol{G}$ at the receiver node where

$$\boldsymbol{G} \times \boldsymbol{X}^T = \boldsymbol{Y}^T. \tag{6.2}$$

In this network scenario, the finite field over which coding is performed is $\mathbb{F}_2$, therefore $\boldsymbol{G}$ can be considered a binary matrix.

As discussed in Chapter 2, it has traditionally been assumed that coding is performed over a large finite field which would render encoded packets linearly independent from one another with high probability [**5**]. However, coding and decoding over large field sizes can further increase the computational complexity of coding schemes [**61**], [**62**]. It was shown by [**8**], [**60**] that sufficient innovative packets can be obtained with high probability by receiver nodes if coding is performed randomly, independently and over a sufficiently large finite field relative to the size of the network, $|\mathcal{V}|$. Work done in [**61**] and [**63**] proved that that when coding in an RLNC network is performed over finite field $\mathbb{F}_2$, a receiver node requires approximately $n + 2$ encoded packets in order to obtain generation matrix $\boldsymbol{G}$ of full rank. This evaluation is also performed in Chapter 4.

In this network configuration we assume that the random linear encoding at network nodes is performed over $\mathbb{F}_2$, as only a small excess of non-innovative packets would be received. We assume that when a packet with a non-innovative global encoding vector is received, the packet is immediately discarded by the receiver node and not added to the generator matrix $\boldsymbol{G}$.

6.2.1 Gaussian elimination

A solution for (6.2) exists when $\boldsymbol{G}$ contains global coding vectors of n linearly independent packets. With high probability, when N is slightly larger than n, the encoding vectors stored in $\boldsymbol{G}$ are linearly independent [**5**], [**60**]. In a traditional network environment, where all the coefficients of the coding vectors are selected at random, $\boldsymbol{G}$ can be viewed as a random matrix. An example of such a

matrix is depicted in Figure 6.1 for $n = 100$, where each dot represents a global encoding vector coefficient equal to 1.

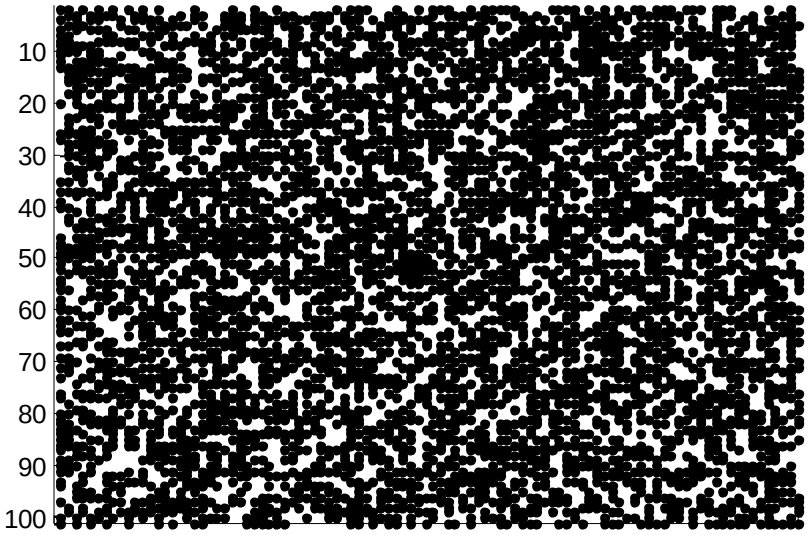


Figure 6.1: Generator matrix received from randomly encoded packets

GE is a well-defined method for solving a system of randomly encoded linear equations of the type shown in Figure 6.1. With the implementation of GE, however, a receiver node would be unable to decode the source packets until an entire block of encoded packets has been received [**7**], as GE can only be performed when $\boldsymbol{G}$ is of full rank n.

The GE method consists of two steps: *forward elimination* and *backward substitution*. The algorithm of GE is presented in Algorithms 6.1 and 6.2 [**80**], [**81**]. Note that, for the sake of simplicity, in the algorithms $\boldsymbol{y}(e_i)$ is written as $\boldsymbol{y}_i$.

Algorithm 6.1: Gaussian elimination: forward elimination

```
for i = 1 → (n − 1) do
    for j = i + 1 to n do
        m_ji = G_ji / G_ii
        for k = j + 1 → n do
            G_kj ← G_jk − m_ji G_ik
        end for
    end for
end for
```

Accordingly, the forward elimination step of GE systematically eliminates unknowns from equations until each equation contains only a single unknown, thus forming a strict lower triangular matrix.

Algorithm 6.2: Gaussian elimination: backward substitution

```
for i = 1 → n do
    x_i ← y_i
    for j = i + 1 → n do
        y_j ← y_j − G_ji * x_i
    end for
end for
```

The backward substitution step solves the equations one by one through the substitution of decoded symbols in the linear equations. Note that these algorithms only show the decoding steps performed on the generator matrix $\boldsymbol{G}$.

These two steps require $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$ operations respectively, which has the disadvantage of high decoding complexity as well as a decoding delay proportional to the number of source symbols [**25**]. In a situation where there is a small number of source messages, GE is an efficient decoding method, but it decreases in efficiency as n becomes larger [**8**], [**39**].

6.2.2 Earliest decoding

Earliest decoding (ED) is a method developed to decrease the decoding delay and the number of arithmetic opertations of GE [**8**] in an RLNC environment. In contrast to traditional GE, where z has to wait for $\boldsymbol{G}$ to be full rank before decoding can commence, ED performs decoding as the packets are collected by a receiver node. This method entails the use of GE on linearly independent packets of sufficient rank as soon as they are collected by a receiver node. The decoder determines the number of undecoded source symbols at the node (number of unknowns), as well as the number of linearly independent coding vectors (number of linear equations). When a receiver node has collected $1 \leq i \leq n$ linearly independent packets containing information on i undecoded source symbols, decoding of the i source symbols is possible through matrix inversion. When source symbols are decoded, they are linearly combined with all the new incoming packets in order to eliminate the decoded symbol from the fresh packets. This means that the decoding of a subset of source symbols, $\boldsymbol{X_{sub}} \subseteq \boldsymbol{X}$, can take place through the inversion of a sub matrix $\boldsymbol{G_{sub}} \subseteq \boldsymbol{G}$ while innovative packets are still being received and the decoding matrix $\boldsymbol{G}$ is incomplete [**8**], [**39**], [**49**].

ED was developed for a practical network scenario where the received packets have a lower triangular structure [**8**]. The decoding delay of ED is approximately constant and is independent of the number of transmitted source symbols, [**8**], [**39**], [**49**]. With the reduction in decoding delay, the resource consumption at receiver nodes is reduced through the use of ED instead of GE. Although ED has a decoding delay independent of the number of source symbols, complex matrix inversion is still employed by the decoder. The process of ED can be explained by means of Example 6.1.

Example 6.1:

Assume that the sink node of a RLNC network receives the following encoded packets: $\boldsymbol{Y} = [y(e_1), y(e_2), \dots, y(e_5)]$. These packets are linear combinations of source symbols from a single generation where $n = 5$, global encoding vectors of which are depicted in the received matrix in Figure 6.2 (a). The sub matrix inversion of the last two packets is illustrated in Figure 6.2 (b).

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \\ \boldsymbol{x_4} \\ \boldsymbol{x_5} \end{pmatrix} = \begin{pmatrix} y(e_1) \\ y(e_2) \\ y(e_3) \\ y(e_4) \\ y(e_5) \end{pmatrix} \qquad \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \boldsymbol{x_4} \\ \boldsymbol{x_5} \end{pmatrix} = \begin{pmatrix} y(e_4) \\ y(e_5) \end{pmatrix}$$

Figure 6.2: (a) Example of earliest decoding **(b) Decoded sub matrix**

With the reception of $y(e_1)$, the receiver is able to decode source symbol $\boldsymbol{x}_1$ as it is the only source symbol present in the packet. Source symbols $\boldsymbol{x}_2$ and $\boldsymbol{x}_3$ can be decoded after the reception of packets $y(e_2)$ and $y(e_3)$ respectively. The receiver node is presented with two undecoded source symbols in $y(e_4)$ and therefore no decoding is possible. Only once $y(e_5)$ is received, is the decoder able to decode $\boldsymbol{x}_4$ and $\boldsymbol{x}_5$ through the use of matrix inversion.

It can therefore be seen that the decoding delay of ED remains in many cases independent of n [**8**], [**39**]. From the simulation in [**8**] it was shown that the algorithmic decoding delay of ED is often only in the order of a few source packets, much smaller than n. Although ED still employs matrix inversion, the blocks that are decoded through GE are small and in most cases independent of n.

6.3 Practical network configuration for improved decoding

In Chapter 2 we discussed the practical solutions proposed in [**8**] for the challenges of realistic network scenarios, which included ED as discussed in Section 6.2.2. In order to implement ED successfully, however, the network environment must be constructed in such a way that the sink nodes receive packets where the global coding vectors resemble a lower triangular matrix. This means that in a lossless scenario the ith packet $\boldsymbol{y}_i$ received by a receiver node tends to be a linear combination of the first i transmitted source packets $\{\boldsymbol{x}_p\}_{p=1}^{i}$, so that $\boldsymbol{G}$ is likely to be lower triangular [**39**]. This lower triangular structure of the received packets allows a receiver node to decode a selection of source symbols $\boldsymbol{X_{sub}} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_i], i \leq n$ with high probability after the collection of i encoded packets [**8**].

This lower triangular form of the generator matrix, as received at the receiver nodes, is what enables the receiver nodes of the network to perform ED. To the best of our knowledge the manner in which to obtain a lower triangular generator matrix at the receiver nodes has not been discussed in the literature. It is merely stated that the lower triangular form of the packets is as result of the causality of the network.

There are many methods, based on network coding, where the form of $\boldsymbol{G}$ can be controlled for easier decoding; however, most of these methods require centralised control mechanisms or complex encoding algorithms at the intermediate nodes [**45**], [**82**]. We presented such an encoding

method in Chapter 5 where a subset of intermediate nodes was required to encode packets that approximate the RS distribution in order to enable BP at receiver nodes.

In the following section we discuss the network configuration we developed in order to obtain generator matrices at sink nodes that resemble a lower triangle in order to successfully implement earliest decoding. All the intermediate network nodes implement RLNC and the structure of the generator matrix is obtained through specified encoding at the source node of the network.

6.3.1 Network causality

A causal network can be seen as a directed acyclic graph where the data present at each non-source node is a function of the data available at its parent nodes [**83**]. As stated in Chapter 2, we assume that all edges of the non-cyclic network $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$ have unit capacity, but may have multiple edges between nodes in order to model different edge capacities, and that information flows over the edges in zero time. As a result of the causality of such a network, it can be assumed with high probability that the source symbols contained in the first encoded packets transmitted over the network would be the first to be received at the sink nodes.

From this we can deduce that a lower triangular structure of $\boldsymbol{G}$ can be obtained at the receiver nodes if source symbols are transmitted sequentially over the network. This means that at transmission opportunity $t_s = 1$, the source would transmit an encoded packet containing symbol $\boldsymbol{x}_1$, and at transmission opportunity $t_s = 2$ an encoded packet containing symbol $\boldsymbol{x}_2$, and so on. This would lead to a sink node sequentially receiving innovative packets possibly containing only a single new source symbol.

We cannot, however, transmit only native packets (source symbols) sequentially into the network, as a source symbol can be deleted completely from the network if just a single native packet is lost before it is linearly combined with other packets in the network. We aim to overcome this problem by encoding each packet at the source to include a single new source symbol. For example, the ith transmitted source packet would include a random linear combination of the first $(i-1)$ source symbols, as well as the ith source symbol, where $1 \leq i \leq n$. This would eliminate the danger of a source symbol being deleted with a single erasure as it would be randomly included in packets transmitted later. Thus, the source effectively transmits encoded packets that resemble a strict lower triangular matrix, where the ith transmitted packet contains a linear combination of the first i source symbols. Due to the transmission of the source symbols in a sequential manner, the causality of the network allows for the source symbols that were transmitted first to have a high probability of arriving first at the receiver nodes. This would allow a receiver node to collect encoded packets that form a non-strict lower triangular generator matrix.

6.3.2 Network configuration

In this section we present a network configuration where the specific encoding of source symbols at the intermediate network node would cause the receiver nodes to obtain packets with global encoding vectors that form a non-strict lower triangular generator matrix.

Consider a network scenario that can be represented by a directed acyclic graph $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$, as presented in Chapter 2. In this instance only a single generation of size n is considered, but can easily be extended to multiple generations. The data at the source node $s \in \mathcal{V}$ contains n source symbols $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_n] \in \mathbb{F}_q$.

We want to transmit the source symbols in a sequential fashion, as described in Section 6.3.1. As the source is presented with a transmission opportunity, an encoded packet $y(e)$ would be constructed from a linear combination of the source symbols where

$$\begin{aligned} y(e) &= \beta_1(e)\boldsymbol{x}_1, \beta_2(e)\boldsymbol{x}_2, \dots, \beta_n(e)\boldsymbol{x}_n \\ y(e) &= \sum_{i=1}^{n} \beta_i(e)\boldsymbol{x}_i. \end{aligned} \tag{6.3}$$

where $\boldsymbol{\beta}(e) = [\beta_1(e), \beta_2(e), \dots \beta_i(e)]$ is generated from finite field $\mathbb{F}_2$ and forms the local encoding vector of packet $y(e)$ at the source node. These coefficients of $\boldsymbol{\beta}(e)$ are not, however, randomly chosen, but selected on the basis of the transmission time step t_s. When $t_s = i$, the local encoding coefficients of the ith encoded packet $y(e)$ will be constructed as follows:

$$\beta_j(e) = \begin{cases} \{1,0\} & \text{if } j < i \\ 1 & \text{if } j = i \\ 0 & \text{if } j > i \end{cases} \tag{6.4}$$

The random selection of local encoding coefficients $\beta_j(e), j < i$ is to ensure that each source symbol is included in other encoded packets so that it will not be lost due to a packet erasure. The inclusion of the source symbol when $j = i$ is to ensure that each encoded packet contains only one new source symbol, as no symbols are further included when $j > i$. Thus, the source effectively transmits encoded packets that resemble a strict lower triangular matrix, where the ith transmitted packet contains a linear combination of the first i source symbols.

When the source node is presented with a transmission opportunity, these encoded packets are multicast over the outgoing edges of the source node and RLNC is performed on the packets received at the intermediate nodes. As the first packets transmitted over the network contained the first source symbols, the causality of the network allows the source symbols that entered the network first to have a high probability of arriving first at the receiver nodes.

The encoded packets $y(e)$ collected by the receiver nodes $Z = \{z_1, \dots, z_{|z|}\}, Z \subset \mathcal{V}$ can be written as a linear combination of the source symbols

$$y(e) = \sum_{i=1}^{n} g_i(e)\, \boldsymbol{x}_i \tag{6.5}$$

where $\boldsymbol{g}(e) = [g_1(e), g_2(e), \dots, g_n(e)]$ is the global encoding vector over $\mathbb{F}_2$ of each packet $y(e)$.

Owing to causality of the network, as discussed in Section 6.3.1, we can assume that the receiver nodes receive encoded packets in a sequential manner in terms of which the packets first transmitted by the source node over the network will be the first packets to be included in the encoded packets $\boldsymbol{Y}$ collected by the receiver nodes. Thus the global encoding vectors of the collected packets $\boldsymbol{Y}$ at each receiver node form with high probability the coding vectors of a non-strict lower triangular generator matrix $\boldsymbol{G}$, where

$$\boldsymbol{G} \times \boldsymbol{X}^T = \boldsymbol{Y}^T. \tag{6.7}$$

As the finite field over which coding is performed is $\mathbb{F}_2$, $\boldsymbol{G}$ can be considered a binary matrix.

This network configuration allows each receiver to obtain a non-strict lower triangular $\boldsymbol{G}$ matrix with high probability. The lower triangular structure of $\boldsymbol{G}$ is not guaranteed, as the structure is dependent on the connectivity and structure of the network. All intermediate nodes encode packets randomly and, in a decentralised network, the length of max-flow paths may vary. These factors may influence the sequential structure of the packets, resulting in source symbols transmitted in early packets arriving later than symbols transmitted after them. Therefore it is possible for the sink to collect an encoded packet containing new source symbols $\boldsymbol{x}_{i+1}$ before collecting a packet containing $\boldsymbol{x}_i$. This means that the matrix may contain non-zero elements on the second, third or above, diagonals.

An example of such a non-strict lower triangular $\boldsymbol{G}$ matrix obtained from an RLNC network, where the source symbols are sequentially encoded and transmitted as discussed in this section, is depicted in Figure 6.3 (a) for $n = 100$. The first 11 received packets are enlarged and shown in Figure 6.3 (b) to illustrate the occurrence of packets that are received out of order owing to network factors such the varying lengths of max-flow paths.

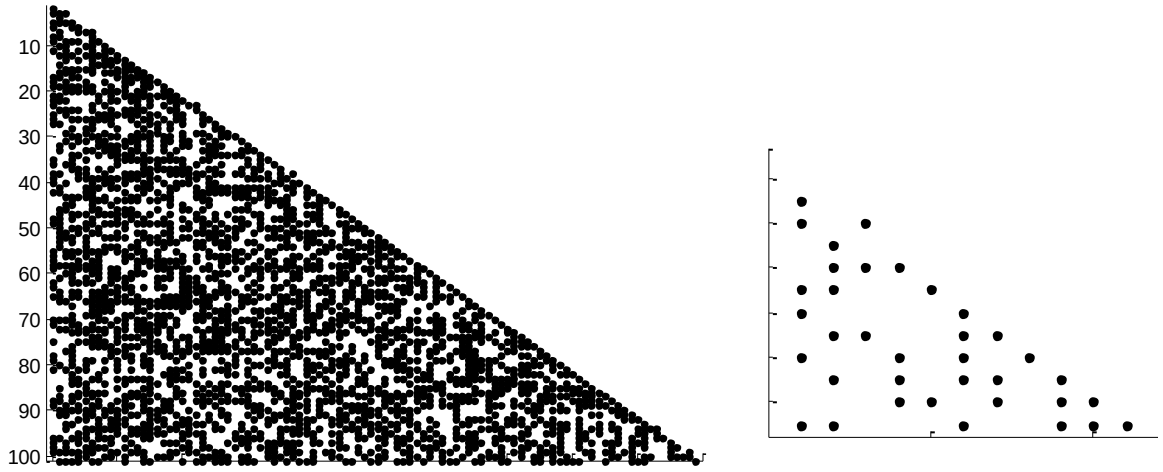


Figure 6.3: (a) Random non-strict lower triangular generator matrix **(b) Enlargement of 1st 11 collected packets of generator matrix**

It can be seen that this network configuration allows us to obtain a $\boldsymbol{G}$ matrix that resembles the form of the $\boldsymbol{G}$ matrix described in [**8**] so that ED can be successfully implemented. In the following two sections we evaluate the network configuration developed and determine the probability of obtaining a non-strict lower triangular generator matrix at the receiver nodes of an RLNC network.

6.4 Evaluation of generator matrices

In our network scenario described in Section 6.3, the finite field over which coding is performed is $\mathbb{F}_2$ which allows all the elements of matrix $\boldsymbol{G}$ to be either equal to 1 or 0. $\boldsymbol{G}$ can be considered as a binary matrix where the entries $\{g_{ij}\}$ of $\boldsymbol{G}$ are distributed according to a Bernoulli distribution where

$$\{g_{ij}\} = \begin{cases} 1 & \text{with probability } P_{ij} \\ 0 & \text{with probability } (1 - P_{ij}) \end{cases}. \tag{6.8}$$

However, owing to the fact that the generator matrix $\boldsymbol{G}$ has a lower triangular structure, we can assume that the probabilities P_j of the occurrence of a non-zero element in a specific diagonal are equal, as shown in Figure 6.4.

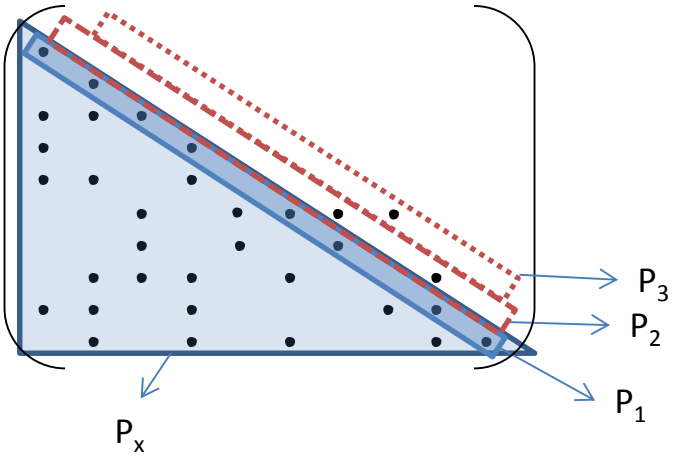


Figure 6.4: Lower triangular matrix with diagonal probabilities

Therefore, not all the entries have unique distributions, but entries on the same diagonal can form a collection with equal probabilities for the occurrence of a non-zero element. We represent P_1 as the probability of receiving a 1 on the main diagonal of the matrix. This translates to the ith received packet containing source symbol $\boldsymbol{x}_i$. P_2 is the probability of receiving a 1 on the second diagonal of the matrix, thus receiving source symbol $\boldsymbol{x}_{i+1}$ in the ith packet. As a result of the fact that the source node randomly includes source symbols $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_{i-1}$ for the ith packet, we consider the probability for the occurrence of a non-zero element below the diagonal as equal, with value P_x.

Therefore, we can simplify (6.8) and state probabilities for all diagonals where

$$\{g_{ij}\} = \begin{cases} 1 & \text{with probability } P_1 \\ 0 & \text{with probability } (1 - P_1) \end{cases}, \forall\, i = j \tag{6.9}$$

is the probability for non-zero elements on the main diagonal;

$$\{g_{ij}\} = \begin{cases} 1 & \text{with probability } P_2 \\ 0 & \text{with probability } (1 - P_2) \end{cases}, \forall\, i = (j - 1) \tag{6.10}$$

the probability for non-zero elements on the second diagonal, and so forth:

$$\{g_{ij}\} = \begin{cases} 1 & \text{with probability } P_m \\ 0 & \text{with probability } (1 - P_m) \end{cases}, \forall\, i = (j - m + 1). \tag{6.11}$$

It is possible to derive a mathematical model to describe the structure of $\boldsymbol{G}$. This mathematical model describes the matrix structure that would allow us to determine the decoding complexity and delay associated with it.

6.4.1 Strict lower triangular $\boldsymbol{G}$ matrix

A strict lower triangular generator matrix is the ideal structure as each newly received packet contains only a single new source symbol. This would enable a receiver to decode a source symbol

with the reception of each new encoded packet, which can be seen as instantaneous decodability, as defined in Chapter 1. The generalised structure, as well as an example of a strict lower triangular matrix, is illustrated in Figure 6.5.

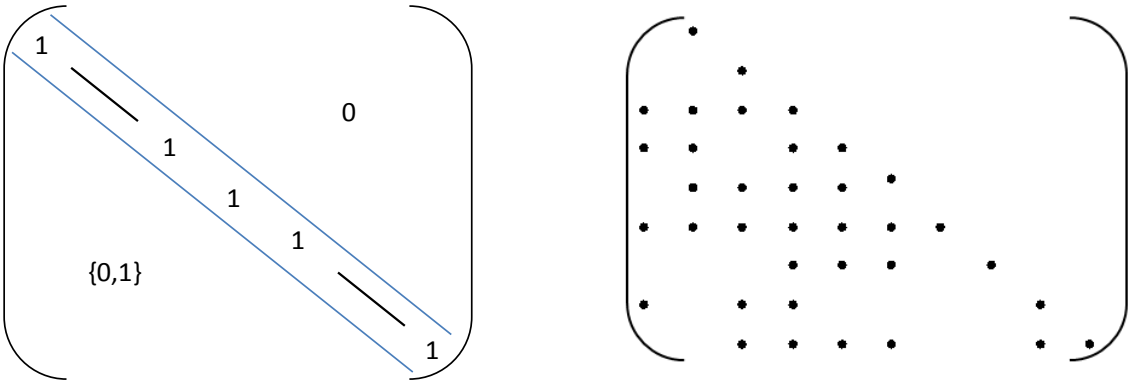


Figure 6.5: (a) Strict lower triangular matrix structure (b) Strict lower triangular matrix example

The probability that $\boldsymbol{G}$ is strictly lower triangular with non-zero entries on the diagonal and only zero entries above the diagonal is

$$\{g_{ij}\} = \begin{cases} 0 & \forall i < j \\ 1 & \forall i = j \\ \{0,1\} & \forall i > j \end{cases} \tag{6.12}$$

As we consider the probability for the occurrence of a non-zero element in a diagonal to be equal, a strict lower triangular matrix requires the following:

- all $n-1$ entries on the second diagonal must be 0, with probability $(1-P_2)^{n-1}$
- all $n-2$ entries on the third diagonal must be 0, with probability $(1-P_3)^{n-2}$
- all $n-j$ entries on the $(j+1)$th diagonal must be 0, with probability $(1-P_j)^{n-j}$
- n entries on the main diagonal must be 1, with probability ${P_1}^n$
- all entries below the main diagonal can be either 0 or 1, with probability 1.

Thus, the probability, P_{strict}, can be determined as:

$$\begin{aligned} P_{strict} &= {P_1}^n \cdot (1-P_2)^{n-1} \cdot (1-P_3)^{n-2} \dots (1-P_n) \\ P_{strict} &= {P_1}^n \cdot \prod_{j=2}^{n} (1-P_j)^{n-j+1} \end{aligned} \tag{6.13}$$

The values of P_{strict} for variable values of n and P_j are discussed in Chapter 7.

6.4.2 Non-strict lower triangular $\boldsymbol{G}$ matrix

Network factors like the variable lengths of max-flow paths in the network can influence the network structure where an encoded packet $y(e_i)$, containing source symbols $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i+1}$ is received before encoded packet $y(e_{i+1})$, which only contains a linear combination of source symbols $\{\boldsymbol{x}_1, \dots, \boldsymbol{x}_i\}$. This results in non-zero entries on second or third diagonals, which can be seen as a lower Hessenberg matrix with a small number of 1s on the second diagonal. The generalised structure of a non-strict lower triangular matrix and an example are shown in Figure6.6.

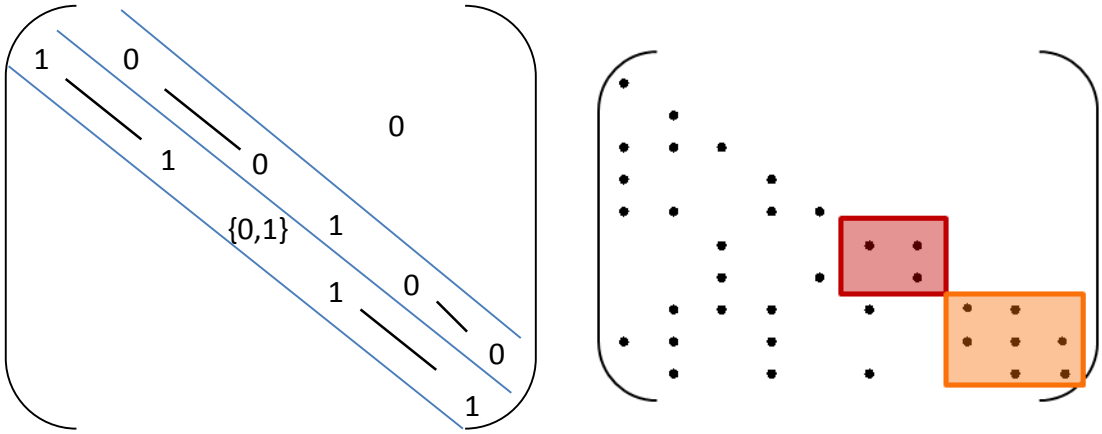


Figure 6.6: (a) Lower Hessenberg matrix with limited 1s in second diagonal **(b) Non-strict lower triangular matrix example**

The probability that $\boldsymbol{G}$ is a non-strict lower triangular matrix with non-zero entries on the second diagonal is equivalent to a Hessenberg matrix with 1s on the second diagonal where

$$\begin{aligned} g_{ij} &= 0, && \forall i < j+1 \\ g_{ij} &= \{0,1\}, && \forall i = j+1 \\ g_{ij} &= \begin{cases} \{0,1\} & \text{if } g_{ij+1} = 1 \\ 1 & \text{if } g_{ij+1} = 0 \end{cases} && \forall i = j \end{aligned} \tag{6.14}$$

Obtaining a Hessenberg matrix with a single non-zero entry on the second diagonal requires the following:

- $n-2$ entries on the second diagonal must be 0, with probability $(1-P_2)^{n-2}$
- 1 entry on the second diagonal must be 1, with probability P_2
- $n-1$ entries on the main diagonal must be 1, with probability ${P_1}^{n-1}$
- 1 entry on the main diagonal can be either 0 or 1, with probability 1.
- all entries on the diagonals above the second must be 0
- the non-zero entry on the second diagonal can be in any of the $(n-1)$ places

Thus, the probability, $P_{non-strict}$, can be determined as

$$P_{Hess\_1} = (n-1) \cdot {P_1}^{n-1} \cdot P_2 \cdot (1-P_2)^{n-2} \cdot (1-P_3)^{n-2} \dots (1-P_n)$$
$$P_{Hess\_1} = (n-1) \cdot {P_1}^{n-1} \cdot P_2 \cdot (1-P_2)^{n-2} \cdot \prod_{j=3}^{n} (1-P_j)^{n-j+1} \quad (6.15)$$

The same process can be followed to calculate the probabilities of φ non-zero entries on the second diagonal, where $1 \leq \varphi \leq (n-1)$. The requirements are the following:

- $n-\varphi-1$ entries on the second diagonal must be 0, with probability $(1-P_2)^{n-\varphi-1}$
- φ entries on the second diagonal must be 1, with probability ${P_2}^{\varphi}$
- $n-\varphi$ entries on the main diagonal must be 1, with probability ${P_1}^{n-\varphi}$
- φ entries on the main diagonal can be either 0 or 1, with probability 1.
- all entries on the diagonals above the second must be 0
- the φ non-zero entries can be placed in ${}_{n-1}C_{\varphi}$ ways on the second diagonal

Thus the probability of obtaining a Hessenberg matrix with $1 \leq \varphi \leq (n-1)$ non-zero entries on the second diagonal is

$$P_{Hess\_\varphi} = \binom{(n-1)}{\varphi} \cdot {P_1}^{n-\varphi} \cdot {P_2}^{\varphi} \cdot (1-P_2)^{n-\varphi-1} \cdot (1-P_3)^{n-2} \dots (1-P_n)$$
$$P_{Hess\_\varphi} = \binom{(n-1)}{\varphi} \cdot {P_1}^{n-\varphi} \cdot {P_2}^{\varphi} \cdot (1-P_2)^{n-\varphi-1} \cdot \prod_{j=3}^{n} (1-P_j)^{n-j+1} \quad (6.16)$$

From (6.16) we can calculate the probability of obtaining a non-strict lower triangular matrix, $P_{non-strict}$, with any number of non-zero entries on the second diagonal. To obtain $P_{non-strict}$ we sum the probabilities of P_{Hess_φ} for $\varphi = \{1,2,\dots,(n-1)\}$.

$$P_{non-strict} = \prod_{j=3}^{n} (1-P_j)^{n-j+1} \cdot \sum_{\varphi=1}^{n-1} {P_1}^{n-\varphi} \cdot {P_2}^{\varphi} \cdot (1-P_2)^{n-\varphi-1} \cdot \binom{n-1}{\varphi}. \quad (6.17)$$

The values of $P_{non-strict}$ for variable values of n and P_j are discussed in Chapter 7.

The non-zero entries on the second diagonal can be non-adjacent or adjacent. The position of the non-zero entries can influence the decoding delay of the decoding method used to decode $\boldsymbol{G}$.

Non-adjacent second diagonal 1s

A packet with a non-zero entry on the second diagonal has the potential to add two new source symbols to the receiver. This would present the decoder with two unknown source symbols in a single packet which would be undecodable by the receiver. If the following innovative packet received does not include a second diagonal entry, the decoder is presented with an additional linearly independent packet that does not contain a new source symbol. With the reception of this second packet, the decoder is able to decode the two source symbols.

An example of this can be seen indicated as a red block in Figure 6.6 (b), where the sixth received packet presents the decoder with two unknown source symbols that can only be decoded after the reception of the seventh innovative packet.

Adjacent second diagonal 1s

A single received packet with a non-zero entry on the second diagonal can add two new source symbols to the receiver. When the following received packet also has a non-zero entry on the second diagonal, that packet also adds a single new source symbol to the receiver. In this situation these packets cannot be decoded immediately, which leads to a larger block that must be decoded.

The presence of $(\varphi - 1)$ adjacent packets with second diagonal 1s forms a block which consists of $(\varphi - 1)$ linearly independent packets and φ undecoded source symbols. In order for this block to be decoded, another packet must be received after the last packet with a second diagonal 1, to add the additional linearly independent equation. This would form a $\varphi \times \varphi$ sub matrix that can be decoded. An example of this can be seen indicated as an orange block in Figure 6.6 (b), where the last three packets must be decoded by means of matrix inversion.

6.4.3 Matrix characterisation

From the above evaluation it can be seen that any $n \times n$ non-strict lower triangular matrix $\boldsymbol{G}$ can be categorised by two types of sub-matrix, those that do not contain entries on the second diagonal and those that do. We name these sub matrices α and β blocks respectively, where the packets in these blocks are called α and β packets.

The β blocks consist of all packets with adjacent 1s on the second diagonals plus the first packet following the last packet with a second diagonal entry. The α blocks contain all the packets with zero entries on the second diagonals except the ones included in the β blocks. An example of this characterisation can be seen in Figure 6.7.

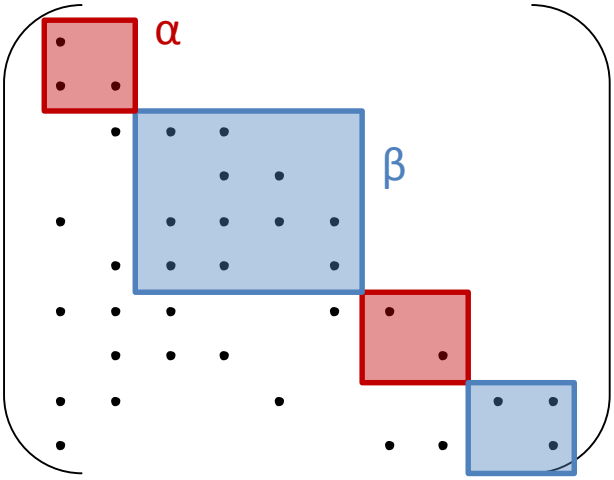


Figure 6.7: Characterisation of G matrix

In the case of a strict lower triangular matrix it is clear that the whole generation would be considered as part of a single α block. This characterisation regarding strict and non-strict lower triangular matrices assists in the calculation of decoding delay and decoding complexity for different decoding methods. These evaluations are performed in Chapter 7.

6.4.4 Linear independency

Various studies have been performed on the number of additional packets a receiver node must collect in order to obtain n linearly independent packets from a network that performs RLNC [**61**], [**69**]. In our network scenario, the finite field over which coding is performed is $\mathbb{F}_2$ which leads to more linearly dependent packets than when coding is performed over a large finite field. We assume that the receiver nodes can monitor the global encoding vectors of the packets to discard non-innovative packets and only populate $\boldsymbol{G}$ with innovative encoded packets. The calculations of probabilities in equations (6.13–6.17) do not take into account the probability of obtaining packets that are linearly dependent, that is, non-innovative, which are discarded.

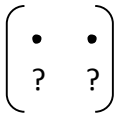
In this section we determine the probability that a packet received from the network for inclusion in lower triangular matrix $\boldsymbol{G}$ is linearly dependent. In a practical network scenario, this probability will equate to the probability of a receiver node discarding a received packet.

Through observation it can be seen that either an α or β packet with a 1 on its main or second diagonal, respectively, would be innovative; an α packet with a 0 on its main diagonal will be non-innovative; and the last packet of a β block may possibly be non-innovative.

The probability of obtaining a non-innovative packet in an α block is equal to the probability of a packet having zero entries on the main and second diagonals:

$$P_{NI\_\alpha} = (1 - P_1) \cdot (1 - P_2). \tag{6.18}$$

The probability of obtaining a non-innovative packet in a β block is higher, as the last packet in the block must be linearly independent without introducing a new source symbol. To obtain the probability of a non-innovative packet, we can assume that all the source symbols received before the β block have been successfully decoded. In a β block of size $\varphi = 2$, the first packet is always linearly independent, as it contains one or two undecoded source symbols. The possibilities of the first β packet are shown in Figure 6.8, where the dots indicate a non-zero entry.



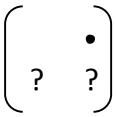


Figure 6.8: (a) First possible structure of β block (b) Second possible structure of β block

The second β packet must not add a third undecoded source symbol, but must be linearly independent of the first packet. Through the evaluation of the β block in Figure 6.8 (a), it can be seen that the second packet will be innovative if it is [1 0] or [0 1], with probabilities $P_x(1 - P_1)$ and $P_x P_1$, respectively. The same probabilities can be calculated for the β block in Figure 6.8 (b). Thus, the probability of obtaining an innovative packet in a β block of size $\varphi = 2$ equals

$$P_{NI\_\beta 2} = P_x. \tag{6.19}$$

where $P_x = (1 - P_x)$ is the probability of obtaining a 1 on the diagonals below the main diagonal. The same process is followed for β blocks of size $\varphi > 2$ and the probability determined for obtaining a non-innovative packet in a β block is

$$P_{NI\_\beta\varphi} = 2^{\varphi-2} \cdot {P_x}^{\varphi-1} \tag{6.20}$$

The method of calculating (6.19–6.20) is described in Appendix B.

These probabilities are determined to show that the possibility exists that non-innovative packets can be obtained by a receiver node in an RLNC network. In a practical network scenario, however, we assume that the receiver node obtains sufficient packets from the network to overcome the reception of non-innovative packets and subsequently discards these. Therefore, in our further analysis regarding the performance of decoding methods, we assume that the receiver can obtain sufficient innovative packets for successful decoding.

6.5 Establishing probabilities of non-zero matrix elements

The developed network configuration can enable us to calculate the probabilities of obtaining non-strict lower triangular generator matrices at the receiver nodes of an RLNC network. These algorithms are useful for evaluating the decoding performance of different decoding methods. Before we can determine how practical decoding algorithms would perform in an RLNC network environment, we must first determine the probabilities of obtaining non-zero elements on the diagonals of the generator matrices constructed at the receiver nodes.

To obtain probabilities that would accurately represent a practical RLNC network, the network configuration described in Section 6.3 is implemented in an RLNC network as described in Chapter 2. Through Monte-Carlo analysis we determine values of P_x, P_1, P_2 etc. for modelling the behaviour of a practical RLNC network.

6.5.1 Simulation setup

The network topology used for these simulations can be described by a network $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$ with a single source s, multiple receivers $Z = \{z_1, \dots, z_{|Z|}\}, Z \subset \mathcal{V}$ and intermediate nodes which implement RLNC. The number of nodes in the network is depicted as $R = |\mathcal{V}|$. We base our experimental setup on that of [**20**] where random geometric graphs (RGGs) were used to model a network implementing RLNC. We selected the network parameters to approximate the practical network considered in [**8**] in order to accommodate the practical considerations of RLNC networks considered in Chapter 2.

The network is modelled by a RGG formed by placing the R nodes uniformly at random on a unit square with a communication radius of l. An edge $e = (v, w) \in \mathcal{E}$ exists between two nodes $v, w \in \mathcal{V}$ when the Euclidean distance between v and w is $d(v, w) \leq l$. We assume a symmetric case where all the network nodes have equal transmission power and, thus, an identical connectivity radius l. The probability p that two nodes v, w are connected is bounded by

$$\frac{1}{4}(\pi l^2) \leq p \leq \pi l^2. \tag{6.21}$$

The lower bound is due to the fact that a node can be situated in one of the corners of the unit square. The upper bound is the direct consequence of the communication radius of a node [**77**].

Let $r(s, Z)$ be the achievable rate at which s can multicast the source packets reliably to the receivers. From the min-cut max-flow theorem, the value of min-cut(s, Z) is the upper bound on $r(s, Z)$ [**6**]. The expected value of min-cut(s, Z) can be approximated by

$$(R-1)p - \sqrt{(R-1)p(1-p)/\pi} \tag{6.22}$$

where the value of p is shown in (6.21) [**78**]. In order to ensure the successful transmission of n source packets from s to Z, the connectivity radius l and the number of network nodes R are chosen to accommodate the required min-cut value min-cut$(s, Z) \geq n$, for varying values of n. If a constructed RGG has a min-cut smaller than required, the graph is discarded and a new RGG is generated.

The data transmitted by the s to Z consists of approximately $hn = 10000$ source symbols in the finite field $\mathbb{F}_{2^8}$. The source symbols are divided into h generations of sizes varying from $n = 35$ to $n = 100$, where the ith generation contains packets $\{x_{(i-1)n+j}\}_{j=1}^{n}$ and where random linear

network coding at the intermediate nodes is performed over $\mathbb{F}_2$. The min-cut(s, Z) from source to receiver nodes must support the transmission of generations of sizes varying from $n = \{35,100\}$.

We followed the method of *independent replications* [**79**] in order to obtain results which are not affected by different network scenarios. For the simulations we generated 40 RGGs with different seeds. From each random graph we ran five instances with different sources and receivers which were randomly chosen. Finally, for each of the sub-instances we ran the simulation five times with different seeds. This equates to 1000 Monte-Carlo simulations.

6.5.2 Experimental methodology

Since we want to determine probabilities for obtaining non-zero elements on the diagonals of the generator matrices obtained by the receiver nodes, certain network parameters are chosen to remain constant throughout this simulation. These parameters may influence the probabilities obtained, but as we aim to obtain probabilities that model this network graph, it will remain constant throughout our simulations.

The following network parameters are specifically constrained for the purpose of the study:

1. *Network topology*. We assume that the nodes in the network simulation are not mobile nodes and remain in fixed positions throughout a single iteration. This is a justified assumption as the causality of the network influences the probabilities obtained.

2. *Continuous transmission*. We assume that the source node continuously multicasts encoded packets over the network until the generator matrix at each receiver node is of full rank. Assume n source symbols in a generation. After the source has transmitted the nth encoded packet over the network, the transmission counter starts again from $t_s = 1$. In this way, the probabilities of non-zero entries can be described accurately for a general network.

3. *Non-overlapping generations.* For this simulation we divided our source data into non-overlapping generations. In this network scenario, a complete generator matrix must be built from packets in a single generation to determine the probabilities of non-zero entries. Overlapping generations would influence the obtained probabilities.

4. *Error and erasure free network.* We consider a network environment where all transmissions are free of errors and erasures.

6.5.3 Simulation results

For each replication, the coding vectors of all the innovative packets received at each sink node were placed in a matrix $\boldsymbol{G}$ and evaluated. As we use diagonals to characterise the generator matrices, we determine the probability of non-zero elements for each diagonal in $\boldsymbol{G}$.

It was found that even when the value of n is changed, the diagonal probabilities remained approximately the same. This shows that the selection of l and R to sustain the required min-cut value for varying values of n has been done correctly. The probabilities of obtaining non-zero elements in a specific diagonal are shown in Figure 6.8 and summarised below.

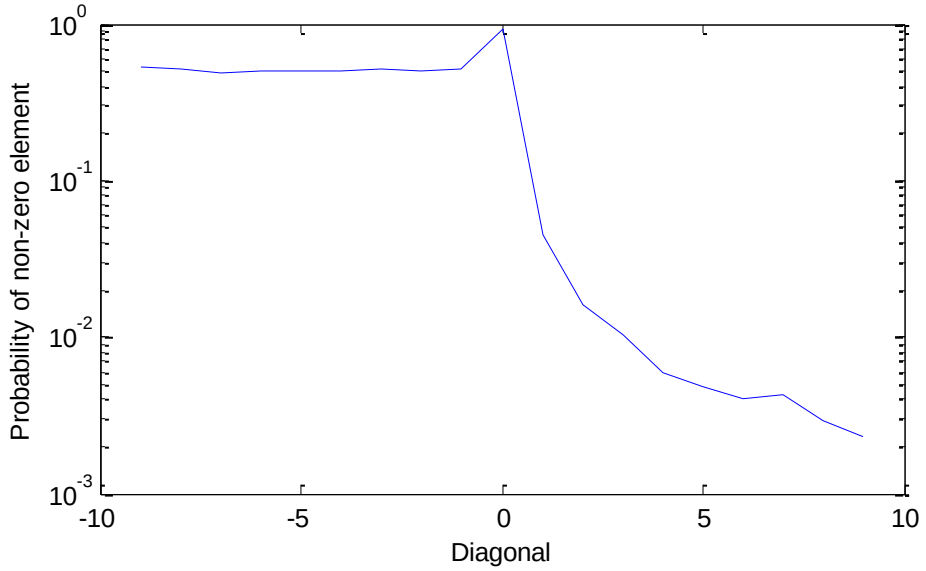


Figure 6.9: Probabilities of non-zero elements in G

The probabilities shown in Figure 6.9 show a relationship with the encoding of the local encoding coefficients for each packet at the source node shown in (6.4). There are, however, some differences. The probability of obtaining non-zero elements on the main diagonal, P_1, is not 1 and the possibility of obtaining non-zero entries above the main diagonal is not 0. In decentralised networks, the length of max-flow paths may vary, which can influence the sequential flow of packets through the network resulting in source symbols transmitted in early packets arriving later than symbols transmitted after them. Also, as coding is performed over $\mathbb{F}_2$, a source symbol in a packet can be eliminated when linearly combined with another packet that contains the same symbol. These factors can result in a packet having a non-zero entry on the second diagonal and, possibly, a zero entry on the main diagonal.

The probability of obtaining a non-zero element below the diagonal is approximately 0.5. This is expected as the local encoding coefficients below the main diagonal are selected randomly at the source node and continue to be randomly encoded with other packets possibly containing the same source symbols.

The obtained probabilities for the various diagonals of matrix $\boldsymbol{G}$ are summarised in Table 6.1.

Table 6.1: Probabilities of non-zero elements in G

	Diagonal	Probability
$\boldsymbol{P_1}$	1 (main)	0.94
$\boldsymbol{P_2}$	2	0.05
$\boldsymbol{P_3}$	3	0.016
$\boldsymbol{P_i, i \geq 4}$:	≥ 4	< 0.011
$\boldsymbol{P_x}$	below main	0.5

The mathematical model describing the generator matrices at the receiver nodes correlates with the practical RLNC network scenario when we use the probabilities obtained from the practical RLNC network.

In the evaluations done in Chapter 7, we only consider the probabilities of P_x, P_1 and P_2, thus assuming $P_i = 0$, where , $i \geq 3$. It can be seen that the probability of obtaining non-zero entries on the third diagonal is $P_3 = 0.016$ and, above the third diagonal, less than 1%.

The exclusion of P_3would result in the mathematical model differing from the practical network in the scenarios where a non-zero entry is obtained on the third diagonal. In the practical network evaluated, a non-zero entry is found on the third diagonal approximately 1.6% of the time. Thus the exclusion of P_3 from our mathematical model is an inaccurate representation of the practical network 1.6% of the time.

The inclusion of P_3, however, would greatly complicate the mathematical model. We chose to build a model that is uncomplicated in the knowledge that it would only represent a practical network accurately 98.4% of the time. The practical networks represented form a specific category of networks, thus the change in a network parameter such as topology would have a greater influence on the mathematical model than the exclusion of the probabilities above the second diagonal.

6.6 Modified earliest decoding

The reception of non-strict lower triangular generator matrices forms an ideal platform for further improvements in decoding. In the following section we propose a new decoding algorithm, called modified earliest decoding (MED), designed for implementation in the network configuration presented above. In the network environment, where lower triangular packets are received, this scheme can offer lower decoding delay and number of arithmetic operations when compared to earliest decoding.

ED is an effective decoding method in this network environment, as it can decode source symbols in blocks as they are collected by a receiver node. The disadvantage, however, is that if the received global encoding vectors are not strictly lower triangular (α block), the receiver is subjected to a decoding delay scalable to size of the β block. The decoding complexity of inverting a β block scales cubic to the size of the block, as shown in Example 6.1.

We present a modification to ED in order to further reduce the decoding delay and the number of arithmetic operations when β blocks are received. The lower triangular structure of $\boldsymbol{G}$ enables us to reduce the use of matrix inversion and implement a decoding method inspired by BP decoding, which was presented in Chapter 5.

6.6.1 Modified earliest decoding

Consider the network $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$ presented in Section 6.3, where non-strict lower triangle generator matrices $\boldsymbol{G}$ are obtained by the receiver nodes $Z = \{z_1, \dots, z_{|z|}\}, Z \subset \mathcal{V}$. The mth received packet $y(e_m)$ tends to be a linear combination of the first m transmitted source packets $\{\boldsymbol{x}_i\}_{i=1}^{m}$. Thus, $\boldsymbol{G}$ largely consists of encoded packets where the packets following each other contain only a single additional source packet.

When an innovative packet is collected by a receiver, ED determines the number of undecoded source symbols at the node (number of unknowns), as well as the number of linearly independent global encoding vectors (number of linear equations). When the number of unknowns and linear equations is equal, decoding can commence as a sub matrix $\boldsymbol{G_{sub}} \subseteq \boldsymbol{G}$ can be successfully inverted. The decoded source symbols are linearly combined with all the incoming packets to eliminate the decoded symbol from the fresh packets. It can therefore be seen that ED cannot continue if the number of linearly independent equations received does not equal the number of undecoded source symbols.

As can be seen from Section 6.4, the probability of an encoded packet adding a single new source symbol to the receiver is very high. As a result of this characteristic and the fact that the generator matrix is binary, it is possible that the linear combination of two successive packets can render a packet which contains a single source symbol. This characteristic of the generator matrix prompted the development of MED.

As with ED, MED runs the decoding algorithm every time a new innovative packet is obtained at a sink node. With the collection of a new innovative packet, the Hamming distances between the global encoding vectors of the packets are evaluated and a single source symbol can be decoded if the Hamming distance between coding vectors of two received packets is equal to 1.

Definition 6.1 [**76**]: We define the *Hamming distance* between two coding vectors $\boldsymbol{g}(e_i), \boldsymbol{g}(e_j)$ as the number of coefficients $[g_1(e), g_2(e), \dots, g_n(e)]$ in which they differ, which is denoted by

$$d_H\left(\boldsymbol{g}(e_i), \boldsymbol{g}(e_j)\right). \tag{6.23}$$

Definition 6.2: We define the *Hamming weight* of the coding vector $\boldsymbol{g}(e_i)$ as the number of non-zero coefficients $[g_1(e), g_2(e), \dots, g_n(e)]$, which is denoted by

$$w_H\big(\boldsymbol{g}(e_i)\big). \tag{6.24}$$

When a receiver node collects an innovative α packet, which presents just a single unknown source symbol, it can be decoded as the global encoding vector of the packet has a Hamming weight of 1. When a block of β packets is received that cannot be decoded immediately as a result of insufficient rank, the decoder calculates the Hamming distances of the packets' coding vectors. If the Hamming distance between two coding vectors is $d_H\left(\boldsymbol{g}(e_i), \boldsymbol{g}(e_j)\right) = 1$, the linear combination of these packets produces a packet with a coding vector $\boldsymbol{g}_{x_i} = \boldsymbol{g}(e_i) \oplus \boldsymbol{g}(e_j)$ of degree one, which is equivalent to a packet containing only a single source symbol $\boldsymbol{x}_i = \boldsymbol{y}(e_i) \oplus \boldsymbol{y}(e_j)$ and thus can be seen as a native or decoded packet.

This native packet can now be used for decoding in a similar way as with BP decoding. By linearly combining $\boldsymbol{x}_i$ with packets already in the buffer, as well as newly received packets containing $\boldsymbol{x}_i$, the degrees of these packets are reduced, as $\boldsymbol{x}_i$ is effectively eliminated from the packets. Thereafter, the decoder determines whether the reduction in degrees of these packets produced a new native packet or whether newly received packets can be linearly combined with others to produce new decoded packets. This process is iterated until all the source packets have been decoded.

6.6.2 Algorithm and example

The MED decoding algorithm is summarised in Algorithm 6.3. Note that, for the sake of simplicity, in the algorithms $\boldsymbol{y}(e_i)$ and $\boldsymbol{g}(e_i)$ are written as $\boldsymbol{y}_i$ and $\boldsymbol{g}_i$ respectively.

Algorithm 6.3: Modified earliest decoding

Initialise $\boldsymbol{g_0} = \boldsymbol{0}, \boldsymbol{g_0} \in \boldsymbol{G}$, and $\boldsymbol{y_0} = \boldsymbol{0}, \boldsymbol{y_0} \in \boldsymbol{Y}$
while not all $\{\boldsymbol{x_i}\}_{\boldsymbol{i=1}}^{\boldsymbol{n}}$ are decoded **do**
  Collect $\{\boldsymbol{g_m}\}_{\boldsymbol{m>0}} \in \boldsymbol{G}$ of received packet $\{\boldsymbol{y_m}\}_{\boldsymbol{m>0}} \in \boldsymbol{Y}$
  while $\boldsymbol{g_m}$ contains a decoded (unit) vector $\boldsymbol{g_{x_i}}$ **do**
    $\boldsymbol{g_m} \leftarrow \boldsymbol{g_m} \oplus \boldsymbol{g_{x_i}}$
    update $\boldsymbol{G}$, $\boldsymbol{Y}$
  end while
  determine $\boldsymbol{d_H(g_p, g_q)}, \{\boldsymbol{g_p}, \boldsymbol{g_q}\} \in \boldsymbol{G}$, $q > p$
  if $\boldsymbol{d_H(g_p, g_q) = 0}$ **do**
    remove $\boldsymbol{g_q}$ from $\boldsymbol{G}$
    update $\boldsymbol{Y}$
  end if
  if $\boldsymbol{d_H(g_p, g_q) = 1}$ **do**
    $\boldsymbol{g_{x_i}} \leftarrow \boldsymbol{g_p} \oplus \boldsymbol{g_q}$
    determine $\boldsymbol{w_H(g_p)}$, $\boldsymbol{w_H(g_q)}$
    remove $\boldsymbol{g_q}$, $\boldsymbol{w_H(g_p) < w_H(g_q)}$
  else remove $\boldsymbol{g_p}$, $\boldsymbol{w_H(g_q) < w_H(g_p)}$
    set $\boldsymbol{x_i} \in \boldsymbol{X}$ equal to $\boldsymbol{y'} \leftarrow \boldsymbol{y_p} \oplus \boldsymbol{y_q}$
    mark $\boldsymbol{g_{x_i}}$ as a native (unit) coding vector
    update $\boldsymbol{Y}$
    if any $\{\boldsymbol{g_m}\}_{\boldsymbol{m \in N}}$ in $\boldsymbol{G}$ contains decoded (unit) vector $\boldsymbol{g_{x_i}}$ **do**
      $\boldsymbol{g_m} \leftarrow \boldsymbol{g_m} \oplus \boldsymbol{g_{x_i}}$
      update $\boldsymbol{G}$, $\boldsymbol{Y}$

24: **end if**
25: **end if**
26: if no MED is possible, perform ED if possible
27: **end while**

Although the structure of $\boldsymbol{G}$ tends to be lower triangular, allowing MED to function successfully, the random encoding of packets does not always guarantee the reception coding vectors with a Hamming distance of 1. In instances where ED can decode $\boldsymbol{G_{sub}} \subseteq \boldsymbol{G}$ and MED cannot, ED is employed, which allows decoding to continue. An example of MED is given below.

Example 6.2:

This example is the same coding matrix $\boldsymbol{G}$ as used to illustrate ED in Example 6.1. Assume a single generation where $n = 5$ and the sink has obtained five encoded packets $\boldsymbol{Y} = [y(e_1), y(e_2), \ldots, y(e_5)]$ from the network that implements RLNC. Note that, for the sake of simplicity, $\boldsymbol{g}(e_i)$ can be written as $\boldsymbol{g}_i$.

I

$$\begin{matrix} y(e_1) \\ y(e_2) \\ y(e_3) \\ y(e_4) \\ y(e_5) \end{matrix} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \\ \boldsymbol{x_4} \\ \boldsymbol{x_5} \end{matrix}$$

II

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}(e_1) \\ \boldsymbol{g}(e_2) \\ \boldsymbol{g}(e_3) \\ \boldsymbol{g}(e_4) \\ \boldsymbol{g}(e_5) \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

III

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}(e_1) \\ \boldsymbol{g}(e_2) \\ \boldsymbol{g}(e_3) \\ \boldsymbol{g}(e_4) \\ \boldsymbol{g}(e_5) \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 \end{pmatrix}$$

$d_H(\boldsymbol{g_0}, \boldsymbol{g_1}) = 1$

IV

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}_{x1} \\ \boldsymbol{g}(e_2) \\ \boldsymbol{g}(e_3) \\ \boldsymbol{g}(e_4) \\ \boldsymbol{g}(e_5) \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{matrix} \\ g_{x1} = g_1 \oplus g_0 \\ g_2 = g_{x1} \oplus g_2 \\ \\ g_4 = g_{x1} \oplus g_4 \\ g_5 = g_{x1} \oplus g_5 \end{matrix}$$

$d_H(\boldsymbol{g_0}, \boldsymbol{g_2}) = 1$

V

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}_{x1} \\ \boldsymbol{g}_{x2} \\ \boldsymbol{g}(e_3) \\ \boldsymbol{g}(e_4) \\ \boldsymbol{g}(e_5) \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{pmatrix} \quad g_{x2} = g_{x1} \oplus g_2$$

$d_H(\boldsymbol{g_0}, \boldsymbol{g_3}) = 1$

VI

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}_{x1} \\ \boldsymbol{g}_{x2} \\ \boldsymbol{g}_{x3} \\ \boldsymbol{g}(e_4) \\ \boldsymbol{g}(e_5) \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{matrix} \\ \\ \\ g_{x3} = g_0 \oplus g_3 \\ g_4 = g_{x3} \oplus g_4 \\ g_5 = g_{x3} \oplus g_5 \end{matrix}$$

VII

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}_{x1} \\ \boldsymbol{g}_{x2} \\ \boldsymbol{g}_{x3} \\ \boldsymbol{g}(e_4) \\ \boldsymbol{g}(e_5) \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

$d_H(\boldsymbol{g_4}, \boldsymbol{g_5}) = 1$

VIII

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}_{x1} \\ \boldsymbol{g}_{x2} \\ \boldsymbol{g}_{x3} \\ \boldsymbol{g}_{x4} \\ \boldsymbol{g}(e_5) \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad g_{x4} = g_4 \oplus g_5$$

$d_H(\boldsymbol{g_0}, \boldsymbol{g_5}) = 1$

IX

$$\begin{matrix} \boldsymbol{g}(e_0) \\ \boldsymbol{g}_{x1} \\ \boldsymbol{g}_{x2} \\ \boldsymbol{g}_{x3} \\ \boldsymbol{g}_{x4} \\ \boldsymbol{g}_{x5} \end{matrix} \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{pmatrix} \quad g_{x5} = g_0 \oplus g_5$$

X

$$\begin{pmatrix} \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{pmatrix}$$

Figure 6.10: MED example

Frame (I): The global encoding vectors of the received packets are shown. In this example sufficient encoded packets are received before the decoding process starts. In this example, this is for clarity purposes. Practically, decoding can commence as soon as two packets with a Hamming distance of one have been received.

Frame (II): By default a sink adds the zero-vector as coding vector $\boldsymbol{g}(\boldsymbol{e}_0)$ to matrix $\boldsymbol{G}$ and then adds the received coding vectors $\{\boldsymbol{g}(\boldsymbol{e_i})\}_{i=1}^{5}$ to $\boldsymbol{G}$.

Frame (III): It can be seen that $d_H[\boldsymbol{g}(\boldsymbol{e}_0), \boldsymbol{g}(\boldsymbol{e}_1)] = 1$. The degrees of the packets are evaluated where $\boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_1)] = 1$ and $\boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_0)] = 0$.

Frame (IV): Because $\boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_0)] < \boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_1)]$, coding vector $\boldsymbol{g}(\boldsymbol{e}_1)$ is removed and replaced by native vector $\boldsymbol{g}_{x_1} = \boldsymbol{g}(\boldsymbol{e}_1) \oplus \boldsymbol{g}(\boldsymbol{e}_0)$. The decoded source packet x_1 is removed from all the packets in $\boldsymbol{G}$ containing x_1. The process starts again where $d_H[\boldsymbol{g}_{x1}, \boldsymbol{g}(\boldsymbol{e}_2)] = 1$.

Frame (V): Coding vector $\boldsymbol{g}(\boldsymbol{e}_2)$ is replaced by native vector $\boldsymbol{g}_{x_2} = \boldsymbol{g}_{x1} \oplus \boldsymbol{g}(\boldsymbol{e}_2)$. The decoded source packet x_2 is removed from all the packets in $\boldsymbol{G}$ containing x_2. The next iteration shows that $d_H[\boldsymbol{g}(\boldsymbol{e}_3), \boldsymbol{g}(\boldsymbol{e}_0)] = 1$ where $\boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_0)] < \boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_3)]$.

Frame (VI): Coding vector $\boldsymbol{g}(\boldsymbol{e}_3)$ is marked as a native vector $\boldsymbol{g}_{x_3}$. The decoded source packet x_3 is removed from all the packets in $\boldsymbol{G}$ containing x_2.

Frame (VII): The next iteration shows that $d_H[\boldsymbol{g}(\boldsymbol{e}_4), \boldsymbol{g}(\boldsymbol{e}_5)] = 1$ where $\boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_5)] < \boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_4)]$. Coding vector $\boldsymbol{g}(\boldsymbol{e}_4)$ is replaced by native vector $\boldsymbol{g}_{x_4}$ where $\boldsymbol{g}_{x_4} = \boldsymbol{g}(\boldsymbol{e}_4) \oplus \boldsymbol{g}(\boldsymbol{e}_5)$.

Frame (VIII)–(IX): The last undecoded vector is $\boldsymbol{g}(\boldsymbol{e}_5)$ which can be replaced by native vector $\boldsymbol{g}_{x_5}$ because $d_H[\boldsymbol{g}(\boldsymbol{e}_5), \boldsymbol{g}(\boldsymbol{e}_0)] = 1$ and $\boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_0)] < \boldsymbol{w}_H[\boldsymbol{g}(\boldsymbol{e}_5)]$.

Frame (X): An identity matrix can be seen which shows that all the source packets have been determined and the transmitted data successfully decoded.

Note: In this example only the linear operations performed on the coding vectors are shown. The same operations are, however, performed on the corresponding data packets $\boldsymbol{Y}$.

From the above example it can be seen that MED is a customised version of ED, where MED also forms a sub-matrix from the global encoding matrix to decode. MED, however, does not perform matrix inversion, but decodes packets by finding those with a Hamming distance of one and linearly combining them.

In Chapter 7 we evaluate the performance improvements of MED over ED in terms of decoding delay, arithmetic operations and resilience to packet erasure.

6.7 Conclusion

In this chapter we presented a network configuration so that the causality of the RLNC network would result in receivers obtaining packets of a non-strict lower triangular structure, which is suitable for ED. This network environment was described in the literature where ED is presented and used, but no coding method or network configuration was presented on how this lower triangular structure was obtained.

We analysed the structure of the lower triangular generator matrix and presented a mathematical model to model the probability of obtaining a generator matrix of such a structure. This model will be used in Chapter 7 for calculations on decoding delay and complexity regarding different decoding algorithms.

Through the analysis of the lower triangular structure of the received packets, a modified version of ED, called MED, was presented. This method does not employ matrix inversion, but decodes packets using an algorithm derived from BP used in LT codes.