

Chapter 1: Introduction

Contents

1.1	Network coding and its challenges	1-2
1.2	Network coding.....	1-3
1.3	Forward error correction codes in network coding	1-6
1.4	Challenges regarding the implementation of RLNC.....	1-7
1.5	Research problem	1-8
1.6	Research contributions	1-13
1.7	Thesis outline	1-15
1.8	Overview of journal and conference contributions.....	1-15

1.1 Network coding and its challenges

Our daily communication infrastructure comprises a wide variety of communication channels with different capacities and characteristics. One fundamental task of such a communication network is to communicate data effectively from a source node to a set of receiver nodes [1]. The aim to make better use of the available network resources has motivated studies to find methods that can improve the network throughput.

The work by Ahlswede et al. in [2] proposes that data should not be seen as a commodity that must be independently transported over the network and delivered at the receivers; rather, they showed that data packets received at nodes from different input edges can be treated as mathematical entities that can be encoded, meaning that information can be mixed together. This mixing of information throughout the network has been termed *network coding* and has been proven to maximise the throughput performance of a multicast communication network [3], [4], [5]. With the increased interest in network coding, it has been found that it offers a wide range of advantages which includes throughput improvement, enhancement of robustness, as well as energy efficiency and improvement in security, when compared to straightforward switching methods [3].

The concept of *random linear network coding* (RLNC) was introduced by Ho et al. in [5] to provide a decentralised realisation of network coding. RLNC is a multicast communication scheme where all intermediate network nodes transmit coded packets which are formed from the random linear combinations of packets received on the incoming edges of the nodes. Each receiver node in the network can successfully decode the transmitted data if any set of sufficient linearly independent encoded packets is received. RLNC has been proven to be capacity achieving with high probability [3], [5], [6], [7], and to have robust to random packet loss and packet delays, as well as network topology and capacity changes, [4], [5], [8].

The random encoding of packets in the network results in all the received packets being functions of all the source symbols with high probability. This complicates the decoding procedure as packets potentially cannot be decoded once they have been received and a decoding procedure must therefore be employed where a series of linear equations is solved. Thus, the advantage of improved throughput comes at the expense of high decoding delays, as several encoded packets need to be received before decoding can commence to recover the original information [7], [9], [10], [11]. Also, RLNC has been proven to impose high computational overheads as decoding complexity increases when the number of transmitted packets increases [12], [13].

As RLNC is generally proposed for most practical network coding applications, [10] states that it is crucially important to design and implement RLNC in networks so that the coding performance is optimised for practical applications. The authors in [12] pose the following question: “Can we design a network code that preserves the communication efficiency of a random linear code and achieves better computational efficiency?”

In this thesis we aim to contribute to the field of practical network coding by finding effective decoding solutions to harness the advantages of RLNC in a practical network environment. By

observing both the advantages and challenges that RLNC offers, we ask the following question: *Can we design effective decoding methods that preserve the communication characteristics of a random linear network code and achieve better computational efficiency and decoding delay?*

Sections 1.2 and 1.3 give an overview of network coding and network error correction, respectively. Section 1.4 states the challenges regarding RLNC and Section 1.5 the research question and the current situation with regard to the solutions proposed in the literature on which the contributions in this thesis are based. Section 1.6 discusses the approach followed in contributing to this research field, as well as listing the specific contributions that this work makes. Finally, Sections 1.7 and 1.8 conclude with the outline and paper contributions of this thesis, respectively.

1.2 Network coding

Network coding and the advantages it offers in different scenarios have been studied by many researchers. The theoretical background on network coding, as well as a detailed study thereof, can be seen in [2], [3], [14], [15], [16], [17] and an extensive survey of network coding applications in [4]. In this section we give a basic overview of network coding and how it can be implemented in practical network scenarios. This topic is discussed in detail in Chapter 2.

The concept of network coding and the advantages it offers can be illustrated by a directed communication network known as a butterfly network, as illustrated in Figure. 1.1 [2]. This network has a single source node S and two receivers, z_1 and z_2 , which must obtain two independent data symbols $\{x_1, x_2\}$. We assume that time is slotted and that each channel can send one symbol from finite field \mathbb{F}_q per time slot.

Both data symbols must be transmitted to both z_1 and z_2 ; this is known as *multicasting* information. In the first instance, all nodes function as a switch where received symbols are merely replicated and transmitted to outgoing links. As only one symbol can be transmitted over each channel in a single unit of time, a decision has to be made at node c as to where x_1 or x_2 has to be transmitted over edge $\{cd\}$. In a multicast scenario, where both receivers must obtain both source symbols, edge $\{cd\}$ forms a bottleneck in the network, because information flows must be kept separate and only one symbol can be forwarded per time slot. This will result in packets being queued and transmitted sequentially. Thus, for z_1 and z_2 to obtain both source symbols, two sequential transmissions over edge $\{cd\}$ are required.

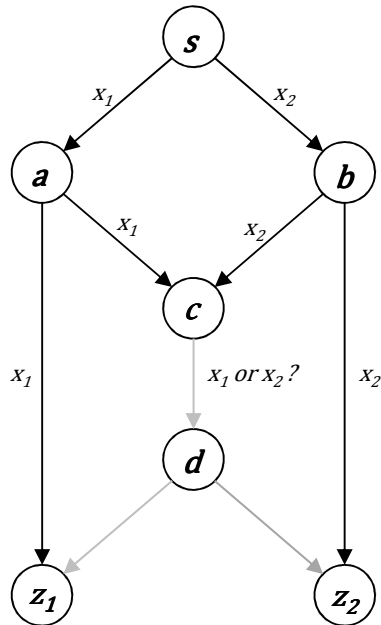


Figure 1.1: Multicast routing in a butterfly network [2]

Network coding allows for the symbols that are received from different input links to be encoded together and transmitted over the output links. Thus, node c can take the received symbols, x_1 and x_2 , and perform a bitwise x-or (i.e. linear combination) on them to create $x_3 = x_1 \oplus x_2$. Symbol x_3 can then be transmitted over paths $\{cd, dz_1\}$ and $\{cd, dz_2\}$. Receiver node z_1 receives $\{x_1, x_1 \oplus x_2\}$, which can be solved to retrieve x_1 and x_2 . Similarly, z_2 retrieves x_1 and x_2 by solving $\{x_2, x_1 \oplus x_2\}$. This process is illustrated in Figure 1.2. The figure shows that network coding can be used to overcome the congestion at node c and that the use of network coding can increase the throughput in directed multicast networks.

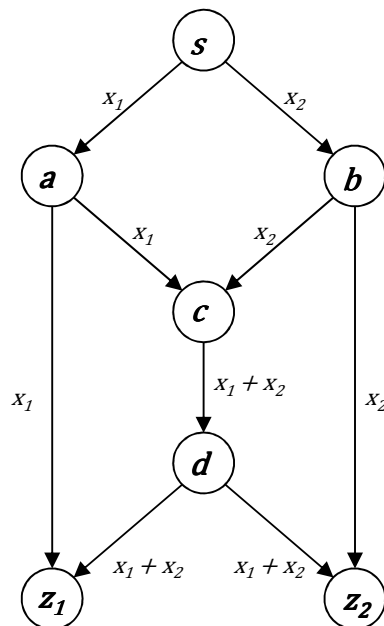


Figure 1.2: Network coding in a butterfly network

Other advantages offered by network coding are energy efficiency, better bandwidth utilisation and an improvement in the length of delay [3]. These advantages are achieved by the better use of resources in the network, where each node implemented with network coding receives the information from all the input nodes, encodes it and sends it out on the outgoing edges.

In deterministic networks, such as a butterfly network, each node uses fixed linear coefficients to form the linear combinations. The disadvantage of this type of network coding is that the network is not very robust owing to its inability to change its functionality in the event of an error, erasure or link failure. It can also be seen that the linear functions created by the intermediate network nodes are controlled by a centralised system where the network topology must be known. In practical networks the topology of the network is not always known and a centralised control system may be impractical.

The introduction of random linear network coding (RLNC) by Ho et al. in [5] provided a practical approach to network coding with many potential applications. RLNC allows for the distributed implementation of network coding in a decentralised environment where nodes carry no knowledge of the topology of the network and each intermediate node requires only the information sent by its neighbour.

The first contribution to the practical implementation of network coding was presented in [8] where a practical RLNC scheme was developed which included realistic network characteristics such as asynchronous transmissions, packet delay and loss, as well as decoding complexities and decoding delay. The practical RLNC scheme included the use of packet buffers and the inclusion of global encoding vectors in packet headers, as well as dividing source symbols into packet generations. The work showed that RLNC can be practically designed to be robust to packet loss, delay and changes to network topology [8], [10].

In RLNC all nodes, except the receiver, select random coefficients from a finite field \mathbb{F}_q for encoding the symbols they receive. As source symbols are divided into generations, only packets from the same generation can be encoded together. The selected coefficients and the generation number are then included in a coding vector which is appended to the header of the transmitted packet. The overhead required for the coding vector becomes negligibly small as the number of source symbols increases [8], [18]. An example of a transmitted packet is illustrated in Figure 1.3.

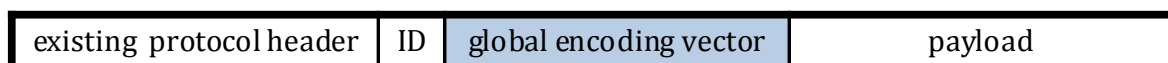


Figure 1.3: Example of the structure of an encoded packet

In an RLNC environment, where the source data is encoded and recoded in a random manner, random structures of coding vectors over \mathbb{F}_q are created. These coding vectors are collected by receiver nodes to construct a generator matrix \mathbf{G} . This matrix relates the source symbols to the packets received for each source-receiver pair [8], [33].

The receiver is able to decode the source symbols by determining the inverse of the encoding operations performed on the encoded packets [13]. The receiver has to wait a certain amount of time in order to receive a set of linearly independent packets that can be used to decode the transmitted message. This can be done when \mathbf{G} consists of sufficient innovative coding vectors, meaning that sufficient linearly independent packets have been received for successful decoding. Not all the packets collected by the receivers are innovative, however. Linearly dependent packets provide no new information to the receiver nodes as they are linear combinations of packets that are already present at the receivers. Thus, they cannot be used by the receivers for decoding and are discarded [5], [19].

It can be seen that RLNC offers the same advantages as network coding and is useful in scenarios where the topology of a network is unknown or constantly changing, or when the maintenance of coordination is expensive or impractical [20].

1.3 Forward error correction codes in network coding

Although the implementation of RLNC offers many advantages in a practical network scenario, the advantages come at a cost. The benefits of RLNC are achieved through the concept of each encoded packet being statistically dependent on almost every other packet. This dependency also makes the network very sensitive to packet loss and errors [21], [22]. An error is defined as the occurrence where the output symbol of a channel differs from the corresponding input symbol [23]. A single corrupted packet in the network has the potential to render an entire transmission useless if it is linearly combined with legitimate packets [22]. The unreliable transmission of packets can lead not only to packet errors but also to erased packets. Subsequently, an erased packet can lead to receiver nodes being unable to decode the source data owing to insufficient linearly independent packets. When a block of packets from a generation cannot be decoded, all the packets from that generation are useless [24].

Countering the effects of errors and erasures is one of the important characteristics of practical networks today. Channels that feed back from receivers to the source can be used to communicate whether incorrect or insufficient packets have been received in order to control possible packet retransmissions. In a multicast or broadcast channel, multiple receivers may require the retransmission of different packets. This can lead to multiple retransmissions where receivers may obtain redundant non-innovative packets [25].

Shannon and Weaver state that reliable communication is possible over a noisy channel with a certain capacity and that the channel capacity can be approached with the use of appropriate coding methods [26]. Many coding techniques have been proposed to reduce or eliminate the number of retransmissions for data lost in the transmission channels [23], [25], [27], [28], [29]. Accordingly, network error correction is widely addressed by implementing forward error correction (FEC) in networks, as introduced in [23].

FEC codes implemented with network coding comprise two techniques, namely, *network error correction codes* and *network erasure correcting codes* [4]. These codes improve the robustness of a

network where successful decoding at the receiver nodes does not depend on receiving packets that contain specific information, but on receiving enough linearly independent packets [30].

Linear FEC codes can be implemented in network coding to ensure the reliable transmission of source data from the source node to the receivers. The implementation of FEC codes in network coding requires the code to be implemented as an outer code at both the source and the receiver nodes of the network. The source node subsequently encodes the original set of source symbols into a larger set of coded symbols to be transmitted over the network. This redundancy will enable the receiver node to decode the source data successfully in the midst of possible errors or erasures [4], [23].

Owing to the fact that the FEC code is applied as an outer code, only the source and the receiver nodes apply FEC techniques. Consequently, the intermediate nodes are unaware of the outer FEC code and simply perform random network coding by creating random linear combinations of their inputs and forward them to the next node [22].

In the case of packet erasures, RLNC itself acts as an end-to-end erasure code where intermediate nodes can potentially generate redundant encoded packets so that any full set of linearly independent packets is sufficient for decoding [31], [32]. When the number of edges out of an intermediate node is larger than the number of flows the network can support, the node generates encoded packets which are redundant. These redundant packets provide limited protection against packet loss [24], [31]. In scenarios where this erasure protection is insufficient, erasure correction codes can also be implemented at the source node. Accordingly, an erasure correction code is applied to the source symbols by converting the symbols into a larger set of coded symbols. When a fraction of the coded symbols has been successfully decoded, the original source symbols can be recovered [34].

There is a wide range of FEC codes that can be implemented at the source node of the network. The selection of such a code, however, involves trade-offs regarding the rate of the code (number of non-redundant symbols relative to the total number of symbols) and its encoding/decoding efficiency [34], [35]. For example: a Reed-Solomon code [36] has an effective code rate, but the decoding complexity is high [35]. On the other hand, Tornado codes [37], [38] have efficient encoding and decoding complexities, but a lower code rate [34]. A detailed discussion on these trade-offs is presented in [35].

1.4 Challenges regarding the implementation of RLNC

RLNC was discussed in Section 1.2, where it was shown that the random encoding of packets in networks leads to encoded packets that are potentially linear functions of all the source symbols. This characteristic of RLNC leads to decoding procedures where a full set of linearly independent encoded packets must be received before decoding is possible, since the generator matrix \mathbf{G} must be inverted. A disadvantage associated with matrix inversion is the fact that decoding can only take place once all the innovative packets have been received, leading to a decoding delay that is proportional to the number of source symbols [7], [9], [10], [11], [12], [13]. Matrix inversion also has

a decoding complexity which scales cubic to the number of source symbols [7], [8], [39]. Thus, the use of matrix inversion in an RLNC network can cause large decoding delays and complexities when a large number of source symbols have to be decoded.

The disadvantage of high computational complexity and delay with RLNC may outweigh the practical advantages of improved throughput, especially for large blocks of information [10], [31]. Moreover, in some practical applications the network nodes are unable to perform complex operations owing to limited time or computational resources [40], [31]. Because of these disadvantages, the coding complexities and the delay of random linear network codes have received attention by many researchers with the intention of developing more efficient coding methods for RLNC [8], [10], [12], [31], [41]. These coding methods aim to provide for the successful decoding of a source symbol as soon as an innovative packet is received.

Another disadvantage associated with an RLNC network is that it is sensitive to packet loss and errors [8], [21], [22]. This sensitivity to errors can be countered by a network error correction code, but requires additional computations at the source and receiver nodes, as this code is implemented as an outer code [4], [23]. As discussed in Section 1.3, RLNC acts as an end-to-end erasure code where block decoding can be implemented at the receiver nodes as soon as sufficient linearly independent packets have been received. With the implementation of decoding methods of lower complexity and delay, an erased packet may negatively affect the decoding efficiency, as some of these methods depend on the systematic reception of packets. Owing to the inherent generation of redundant packets in the RLNC network, the packets may still be decodable, but only when the whole generation of packets has been received, thus eliminating the decoding complexity and delay advantages.

Research has been done to develop decoding methods that would enable the decoding of packets as they are received in the midst of packet erasures, for example Luby Transform (LT) codes [42]. Further, when an outer code is added to the LT codes, the decoding delay and the complexity involved in decoding these codes can be further reduced, as the source symbols can be successfully recovered when a fraction of received packets has been decoded [38]. These codes are called Raptor codes [43].

1.5 Research problem

From the discussion in the previous sections, it can be seen that RLNC can offer a wide range of advantages, but that it comes with a range of challenges that cannot be ignored. In this section we shall state the research problem, discuss the literature related to our research questions and state the challenges that still remain. Thereafter we discuss our approach to resolving some of these challenges and the impact of our solutions.

1.5.1 Research question

Random linear network codes are generally proposed as the solution for practical network coding applications [10]. Therefore, it is important to design and implement RLNC in such a way that the encoding and decoding methods perform efficiently in practical scenarios. In this thesis, we contribute to the field of practical network coding by proposing effective coding methods for RLNC networks that lead to encoding and decoding algorithms of low complexity and delay, as well as successful decoding in the face of packet errors and erasures.

This leads to our research question:

Can we design effective decoding methods that preserve the communication characteristics of a random linear network code and improve computational efficiency and decoding delay as compared to current methods?

1.5.2 Related literature

Much research has been done on the development of more efficient coding methods for RLNC in order to use it in practical networks; the reason being that network nodes may have limited storage, time and computational resources [31]. We subsequently discuss research done by other individuals and academic institutions that relates to our research question as well as the challenges that remain.

Forward error correction in RLNC

FEC codes can easily be implemented in an RLNC environment as an outer code where additional parity symbols, which are functions of the original source symbols, are added at the source node. The implementation of an FEC code in RLNC networks was briefly discussed in Section 1.3 and is the topic of Chapter 3.

RLNC codes can potentially also generate a large number of redundant coded symbols at intermediate network nodes. This relies on the topology of the network: if the capacity at an intermediate node exceeds the global capacity of the network, redundant packets are created by the node [24], [33]. This redundancy is widely used to combat erasures in the network, as the redundant packets can replace packets lost in the network [8], [10].

In [50] an FEC technique was presented that allows for errors to be detected without the addition of an outer code at the source node of the network. The source transmits just the original source symbols over the network. As an RLNC can potentially generate redundancy at intermediate nodes, these redundant packets were not discarded at the receiver nodes, but used for error detection. This technique utilises the excess packets formed by the RLNC process for use as parity for the detection of errors. This method enables error detection in the network without the use of additional network resources in order to implement an outer code at the source node.

The FEC scheme in [50] was only presented and simulated for limited network scenarios, with no further investigation being done on the efficiency and practicality of the method. Owing to the limited research done in this field, mathematical analysis could prove useful in determining how efficient this method is in practical RLNC scenarios.

Fountain codes in RLNC networks

Firstly, we discuss different approaches to the implementation of fountain codes, which include Luby Transform (LT) [42] and Raptor codes [43] in the RLNC environment. The successful implementation of fountain codes in an RLNC network environment can replace the need for high-complexity matrix inversion at receiver nodes with the use of the low-complexity decoding method of LT codes, called belief propagation (BP) [42], [44].

BP is practically implementable as it employs an iterative decoding algorithm with low complexity, which scales logarithmically to the size of the source data [42]. In addition, the decoding of source packets can potentially start before all the encoded packets have been collected by a receiver node, thus reducing the decoding delay as well. Raptor codes further improve on LT codes where the use of an outer code in conjunction with LT codes leads to linear coding and decoding complexities in the RLNC network. The use of Raptor codes also allows a receiver node to decode a fraction of the transmitted symbols to recover the original source data.

The success of BP is, however, dependent on the statistical properties of the encoded packets. The Robust Soliton (RS) distribution was developed by Luby to ensure the decoding of the packets [42]. If BP decoding is to be implemented in RLNC networks, the packets cannot be encoded randomly at the intermediate network nodes, as this type of decoding requires packets to be encoded according to the RS distribution. This requires intermediate nodes to decode and re-encode packets to fit the required distribution, which defeats the original purpose of network coding and leads to delays at the intermediate nodes [10], [31].

An encoding procedure called LT network codes (LTNC) was developed [45] for content dissemination networks where all network nodes require the data to be transmitted from the source. In order for all network nodes to successfully implement BP, all the nodes were required to encode the received packets according to the RS distribution. However, this encoding procedure proved to be complex and resource intensive and required a centralised control system. It is stated in [31] that the sparse encoding of packets to approximate fountain codes may be possible in specific network scenarios like [45], but is difficult to implement in a communication network where intermediate nodes do not require the source information.

A method called Batched sparse codes was introduced in [1], [31] for a communication network where data is transmitted via intermediate network nodes. This method allows for the implementation of BP decoding at the receiver nodes without the need for complex decoding and re-encoding at network nodes. The encoding complexity of these codes scales linearly to the number of source symbols, but a pre-decoding step of quadratic complexity is required at the receiver nodes.

It is clear from the literature that the use of BP at receiver nodes can have great advantages in reducing decoding complexity and delay in an RLNC network. Studies on the implementation of fountain codes in RLNC networks are, however, limited. To our knowledge, the first paper relating to the combination of fountain codes and random linear network codes was published in 2009 [45] for content dissemination networks, while the first work on communication in multicast networks was presented in 2011 [1]. The recent interest in this field points toward the feasibility of implementing BP as a practical decoding method in RLNC, which motivates the development of innovative methods to do so.

Earliest decoding in RLNC networks

The second approach followed to simplify the encoding and decoding complexity of RLNC is to group the source symbols into disjoint generations (also named chunks or classes), as first suggested in [8]. With the use of generations in RLNC, the source data is divided into smaller equal chunks where the task of transmitting a large file is subdivided into the transmission and decoding of smaller files [31], [46]. This means that packets of the same generation can be encoded together at the intermediate network nodes and decoding is performed on each generation separately [8], [46], [47].

For example, assume that hn source symbols are present at the source node. With RLNC where Gaussian elimination (GE) is implemented, a receiver node must first collect hn encoded packets with linearly independent coding vectors to invert a generator matrix \mathbf{G} of size hn with decoding complexity $\mathcal{O}((hn)^3)$. If hn is large, the scheme has a large decoding delay and becomes difficult to implement, as the inversion of a large \mathbf{G} matrix becomes complex [31]. When the hn source symbols are divided into h generations of size n , decoding can commence after the reception of n linearly independent packets from the same generation leading to decoding complexities of $\mathcal{O}(hn^3)$.

In practical RLNC networks the use of generations can lead to increased overhead and storage costs if the number of generations is large [31]. Additionally, packets from certain generations can accumulate faster at certain nodes than others. This can lead to the decoding of a generation when packets belonging to that generation can still be received, resulting in the reception of redundant packets as well as a delay in the decoding process [31], [46]. On the other hand, if insufficient packets from a generation are received, the whole generation cannot be decoded and is rendered undecodable.

These challenges have been addressed by various methods, including the use of overlapping generations where decoded generations assist in the decoding of other generations [7], [31], [46], [47]. When the generations are allowed to overlap, the decoded generations can reduce the number of undecoded source symbols in other generations, which reduce the number of packets needed to decode those generations. Overlapping generations lead to the systematic decoding of generations while packets are still being received. The effect of generation and overlap size in RLNC on throughput and coding complexity was investigated by [46].

The use of generations can be useful in decreasing the computational complexity and decoding delay at receiver nodes. This decoding delay and complexity can be further improved by enabling the decoding of packets as they arrive within a generation. *Instantaneous decodability* is defined in [7], [48] as the ability of a receiver to decode a new source symbol with every successful innovative packet received. Decoding methods able to decode packets without delay as they are received are thus useful in improving decoding efficiency in RLNC networks.

A method called earliest decoding (ED) was proposed by [39] for RLNC networks that offer practical decoding delays. ED has the potential to decode source packets in a generation while innovative packets are still being received. The decoding delay of ED is approximately constant and independent of the number of source symbols in a generation [8], [39], [49]. 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 decoding efficiency of ED, however, is dependent on a specific lower triangular structure of the generator matrix obtained by a receiver node. It is stated in [8], [39] that the causality of computations in an RLNC network can lead to received packets having a lower triangular structure. This means that the m th received packet is a linear combination of the first m source symbols with high probability. Thus, for each generation to be successfully decoded with ED, not only must enough linearly dependent packets be received, but they should also preferably be in a lower triangular form to reduce decoding delay.

An erased packet can have a large influence on the decoding efficiency of ED, as it can halt the systematic decoding in a generation owing to the requirement for a lower triangular structure of the generator matrix. When a packet is erased, the generation may still be decodable as a result of the reception of redundant packets, but only at the end of the generation, as the lower triangular structure of the packets is compromised. This inability of ED to handle possible packet erasures leads to an opportunity to improve the decoding method in the presence of packet erasures.

1.5.3 Research approach

In our discussion on existing research relating to our research question, it can be seen that current solutions in the literature are obtained through many different approaches and a wide variety of methods. The solutions include the construction of innovative schemes to overcome a challenge or to find solutions in other research fields and incorporate it in this one.

In this thesis we contribute to this research field by building on the existing research discussed in the previous section. Three different techniques have been used in this thesis:

- The construction of new and innovative schemes and algorithms to solve problems in the field.
- The utilisation of techniques and methods from other research areas to resolve challenges in this field.
- The utilisation of methods and schemes familiar to this research field in unique and different ways to solve a problem.

The methods developed in this thesis are evaluated through the use of theoretical analysis and simulations. In our theoretical analysis we examine different aspects of the proposed methods in order to gain an understanding of the theoretically expected performance in RLNC networks. We use simulations to validate the theoretical conclusions made; evaluate possible trade-offs of parameters that can arise in practical systems; and determine the practical effects of the respective methods in practical RLNC networks.

The various methods and approaches found in the literature are also evaluated by means of analysis and simulations in order to ensure an accurate comparison of the existing and the new methods.

1.6 Research contributions

This thesis aims to provide practically oriented solutions to the challenges relating to coding complexity and delay in RLNC environments. We propose the use of different decoding methods in conjunction with RLNC to offer opportunities for reduced decoding delay and complexity. Accordingly, the design of efficient decoding techniques may assist in the saving of processing, memory and energy requirements in network coding networks.

As can be seen from Section 1.5.2, the work done in this thesis can be divided into three main areas: exploiting redundancy in RLNC for error detection; implementing fountain codes in RLNC networks; and developing a practical decoding method for RLNC. A brief description of the work presented in this thesis, as well as the contributions made in each area, will be subsequently discussed.

1.6.1 Exploiting redundancy in RLNC for error detection

We contribute to the field of network error correction by further investigating the method proposed in [50]. By means of analysis we determined that by adding no outer code to the RLNC network, basic error detection is possible due to the redundancy generated by RLNC. This error detection method is not optimal, but does not effectively cost the network any additional resources when implemented.

Our contributions in this research area are the following:

- We improved the method developed in [50] in terms of matrix construction and packet selection for error detection.
- We constructed and validated an analytical expression considering a network where randomly encoded packets are received. This expression was developed to describe the reception of packets in an RLNC network and can be used to
 - analyse the probability of constructing a valid generator matrix \mathbf{G} for error detection after the reception of n packets

- calculate the number of additional packets required to guarantee error detection in an RLNC network.
- We show that with the reception of approximately two additional packets, single errors can be detected without the addition of a pre-code at the source node.
- We evaluate the influence of network topology on the scheme in order to find the correlation between the developed mathematical model and the network environment and show that the analytical expression that was developed models a specific network topology accurately.
- We show that the implementation of this scheme can offer several advantages in resource limited networks.

1.6.2 Implementation of fountain codes in RLNC networks

We contribute to the field of fountain codes in RLNC networks by building on the work done in [45] on content dissemination networks. We present an encoding scheme for data communication between source and multiple receivers via intermediate nodes, where the decoding and recoding of packets are not required, as in [45]. This scheme allows for the implementation of BP at the receiver nodes and requires just a small encoding alteration to a fraction of intermediate nodes. We show that the encoding complexity associated with the implementation of BP in RLNC can remain relatively low without affecting the decoding efficiency of BP.

Our contributions in this area include the following:

- We present an improved coding scheme, called Hybrid-LTNC (H-LTNC), adapted from the LTNC method for data communication between the source and multiple receivers.
- We propose further improvements to the H-LTNC in terms of coding procedures and buffer flush times at the intermediate nodes.
- Through the use of simulation and analysis we show that the scheme provides low complexity linear encoding at most network nodes and low complexity BP decoding at the receivers.
- We present a further improvement to the new method where the implementation of an outer erasure code leads to even lower decoding complexities at the receiver nodes.

1.6.3 Development of a practical decoding method for RLNC

We contribute to the field of practical decoding methods in RLNC by modifying the earliest decoding method proposed in [8] by combining it with the iterative decoding principle of BP found in fountain codes. This modification changes the decoding procedure of ED by eliminating the use of matrix inversion and also makes the ED procedure more robust to erased packets. We show that the

modified earliest decoding method can decode with low delay even in the presence of packet erasures.

Our contributions in this research area include the following:

- We present a practical network configuration where the specific arrangement of source-encoded packets in an RLNC network can lead to the receivers obtaining packets of a lower triangular structure, which is essential for the success of ED.
- We evaluate the structure of the received lower triangular generator matrix and present a mathematical model for calculating decoding delay and complexity in these network scenarios.
- We propose a new decoding algorithm, called modified earliest decoding (MED), designed for the network scenarios where ED is implemented. By means of theoretical analysis and simulations we show that this scheme outperforms ED in terms of decoding delay and also requires fewer arithmetical operations for successful decoding.
- We investigate the efficiency of ED and the new MED in the presence of packet erasures.
- We show how MED can continue decoding after the occurrence of erased packets. We present further improvements to the developed decoding scheme in order to allow for effective decoding in the midst of lost packets.

1.7 Thesis outline

In Chapters 2 and 3 we discuss the necessary mathematical background on RLNC and network error correction and establish the notation used in this thesis. Chapter 4 presents the research on the use of redundant packets in RLNC networks for error detection. Chapter 5 presents the implementation of fountain codes in an RLNC network environment. The design and evaluation of a practical network configuration and the MED method are presented in Chapter 6. The analytical and practical evaluation of MED and existing decoding methods is presented in Chapter 7. Chapter 8 provides our conclusions and suggestions for further work.

1.8 Overview of journal and conference contributions

The results obtained and presented in this thesis have been published in the following journal and conference papers. These papers are included in this thesis in Appendix A.

- J1: S. von Solms and A.S.J. Helberg, "An evaluation of redundancy for Implicit Error Detection in Random Network Coding," Errata to C1, improved and extended version of C1 for possible future submission.

- J2: S. von Solms and A.S.J. Helberg, “Random Linear Network Coding for Belief Propagation Decoding,” submitted to the Journal of Network and Computer Applications, Sept 2012.
- J3: S. von Solms and A.S.J. Helberg, “Modified Earliest Decoding in networks that implement Random Linear Network Coding,” Africa Research Journal, vol. 103, no. 4, pp. 165–171, Dec 2012.
- J4: S. von Solms and A.S.J. Helberg, “Evaluation of Decoding Methods for Random Linear Network codes,” submitted to the Journal of Network and Computer Applications, Nov 2012.
- C1: S. Von Solms, S. J. de Wet and A. S. J. Helberg: “Error correction for resource limited random network coding networks,” Proceedings of the IEEE Africon, Livingstone, Zambia, September 2011.
- C2: S. von Solms and A.S.J. Helberg, “The implementation of LT network coding in resource limited RLNC networks,” Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC), East London, South Africa, 2011
- C3: S. von Solms and A.S.J. Helberg, “Encoding for belief propagation decoding in random network codes,” Proceedings of the Southern Africa Telecommunication Networks and Applications Conference (SATNAC), Fancourt, George, South Africa, 2012.
- C4: S. von Solms and A.S.J. Helberg, “Modified Earliest Decoding for Random Network Codes,” Proceedings of the 2011 International Symposium on Network Coding (NetCod), Beijing, China, 2011.

Table 1.1 shows where the contributions of the journal or conference paper can be found in this thesis.

Table 1.1: Journal and conference contributions

	Chapter 4	Chapter 5	Chapter 6	Chapter 7
J1	x			
J2		x		
J3			x	
J4			x	x
C1	x			
C2		x		
C3		x		
C4			x	