# Encoding for belief propagation decoding in random network codes

Suné von Solms, Albert S. J. Helberg TeleNet Research Group School of Electric, Electronic and Computer Engineering North-West University, Potchefstroom Campus Tel: (018) 299 1966, Fax: (018) 299 1977 E- mail: <u>sune.vonsolms@nwu.ac.za</u>, <u>albert.helberg@nwu.ac.za</u>

Abstract—The Hybrid- Luby Transform network code is an encoding method proposed for the implementation in communication networks employing random linear network coding. This method enables receiver nodes to implement low complexity belief propagation decoding. In this paper we show that the implementation of sparse random linear network coding and a less frequent buffer flushing policy to H-LTNC enables near optimal belief propagation decoding in a random linear network coding scenario.

# Index Terms— Belief propagation decoding, Fountain codes, LT codes, Random Linear Network Coding.

#### I. INTRODUCTION

One of the criteria for measuring the effectiveness of a communication network is to determine its ability to transmit a bulk of data from the source to receiver nodes. There exist several challenges regarding effective information transmission in networks due to interference from other devices, environmental factors, as well as limited available resources.

In a network where the transmission between source and receiver nodes can be modelled by an erasure channel, fountain codes can be a very effective method of communication. Fountain codes, which are rateless codes, include Luby Transform (LT) codes [1] and Raptor codes [2]. Fountain codes require the source node to transmit n encoded source packets to the receiver via intermediate network nodes that only implement a store-and-forward algorithm to the packets they receive. A receiver is then able to decode the transmitted data when it receives N encoded packets, where  $N = n + \delta$  with  $\delta$  small in relation to n [1]. LT codes and Raptor codes require the source packets to be encoded according to a specific degree distribution as this distribution of packets allows for the implementation of a low complexity belief propagation (BP) decoding algorithm.

The store-and-forward technique implemented at the intermediate nodes of the communication network does not allow for the optimal utilisation of the communication channel. In order to utilise the communication channel optimally, Ho et al. [3] suggested the implementation of

random linear network coding (RLNC). The implementation of linear coding at the intermediate nodes of the network leads to an improvement in the utilisation of network capacity which improves network throughput [4].

RLNC is a method that can easily be implemented in a practical network scenario by allowing intermediate nodes to randomly and linearly encode the packets received on their incoming edges to produce a new encoded packet. When the encoding at the intermediate nodes is done randomly and the operations are in a large enough finite field, the multicast capacity of the network can be reached [3].

The use of fountain codes in conjunction with RLNC in a communication network offers the advantage of low complexity BP decoding in a network that communicates at multicast capacity. LT codes require the encoding of packets to be according to the Robust Soliton (RS) degree distribution [1]. The random linear encoding at the intermediate network nodes, however, leads to *degree degeneration* where the specified input degree distribution degenerates with each random recoding at the receivers fail [6].

Several methods [6]-[8] have been presented to prevent the occurrence of degree degeneration at intermediate nodes allowing for the successful implementation of fountain codes in a RLNC environment. In this paper, we propose improvements on the method presented in [7] to allow linear encoding at most of the intermediate network nodes and low complexity BP decoding at the receivers.

#### II. BACKGROUND

#### A. Random linear network coding

Consider an acyclic network  $\mathbf{G} = (\mathcal{V}, \mathcal{E})$  implementing RLNC. The network consists of a single source node  $s \in \mathcal{V}$  and a set of sink nodes  $Z = \{z_1, \dots, z_{|z|}\}, Z \subset \mathcal{V}$ . The achievable rate at which *s* can multicast the source packets reliably to the set of receivers *Z* is r(s, Z). The maximum flow of the network for any  $z \in Z$  is the upper bound on r(s, z), thus min-cut  $(s, z) \ge n$  [9].

The source data is divided into *n* packets,  $X = [x_1, x_2, ..., x_n]$  where  $x_i$  represents the *i*th source packet of size *m* in a finite field  $\mathbb{F}$  of size *q*. These source packets are multicast over the edges  $\mathcal{E}$  of the network from *s*. The intermediate nodes randomly and linearly combine, through X-OR operations, the packets received on their incoming edges e' to form a new encoded packet for transmission on their outgoing edges *e*. The encoding complexity at each intermediate node equals  $\mathcal{O}(mb)$ , where *b* is the size of the buffer [6]. In the header of each transmitted packet is a coding vector of length *n*, describing the included source packets  $x \subseteq X$  [10].

Each receiver node  $z \in Z$  collects a set of  $N \ge n$  encoded packets  $Y = [y_1, y_2, ..., y_N]$  from the network where the packets' global encoding vectors form the column vectors of a  $n \times N$  matrix **G** where

$$X \times G = Y. \tag{1}$$

The finite field  $\mathbb{F}_q$  of the global encoding vectors is sufficiently large so that **G** is invertible with high probability when N is only slightly larger than n. The solution of the linear system of equations in (1) decodes the source packets **X** [10].

Traditionally the decoding method employed in RLNC networks is Gaussian Elimination whose matrix inversion algorithm is computationally complex and of order  $\mathcal{O}(n^3 + mn^2)$  [6]. However, when the global encoding vectors in **G** resemble that of the RS degree distribution, low complexity BP decoding would be a more efficient decoding method.

#### B. Fountain codes

Fountain codes, which include LT codes, are a low complexity approach to linear coding. The optimal degree distribution for LT codes is described by the RS distribution. The encoding and decoding complexity of LT codes are  $O(mn \log n)$ .

The BP decoding process can be described by the following algorithm [1]:

- 1. Find an encoded packet,  $y_j, 1 \le j \le N$ , that only contains a single source packet,  $x_i, 1 \le i \le n$ .
- 2. Set source packet  $x_i = y_j$  and delete  $y_j$ .
- 3. Subtract the value of  $x_i$  from all the other encoded packets  $\{y_p\}_{p=1}^N$  that contains source packet  $x_i$ .
- 4. Repeat from (1) until all source packets  $x_i, 1 \le i \le n$  are determined.

# C. RLNC and Fountain codes

It can be seen that if fountain codes are to be implemented in a RLNC environment, all the intermediate network nodes are not able to simply create random linear coded packets for transmission. BP decoding at the receivers require the encoding of packets to be according to the RS degree distribution. Thus an encoding method that prevents the occurrence of degree degeneration needs to be implemented at intermediate nodes allowing for the successful implementation of fountain codes in a RLNC environment.

A method called LT network codes (LTNC) was suggested in [8] where each intermediate network node is forced to encode packets according to the specified RS degree distribution. The implementation of LTNC allows for BP at the receiver nodes of the network, but the algorithm implemented at each intermediate node is of very high complexity as it runs sub-optimal coding and refining steps.

An improvement to LTNC, called Hybrid-LT network coding (H-LTNC), was proposed in [7]. In H-LTNC most of the intermediate nodes implement RLNC and only nodes connected to the receiver nodes implement a simplified LTNC encoding algorithm. It was found that the implementation of the LTNC algorithm at all the intermediate nodes is unnecessary as the receivers only obtain packets from network nodes they are connected to. The use of H-LTNC in a RLNC network reduces the encoding complexity at the intermediate network nodes and still allows for the implementation of BP decoding at the receiver nodes.

In this paper we present an improved H-LTNC, Enhanced H-LTNC (EH-LTNC) to ensure accurate encoding of packets of the needed target degrees at intermediate network nodes. This optimisation reduces the number of additional packets required for decoding and reduces the decoding delay of the method.

#### III. HYBRID-LT NETWORK CODES

In this section we shall provide a brief description of the H-LTNC method at intermediate nodes as presented in [7].

Consider *n* source packets of size *m* in a finite field  $\mathbb{F}$  of size *q*. Each intermediate node  $v \in \mathcal{V}$  collects packets from its incoming edges  $e', \mathbf{y}(e'_u)$ , where *u* is the number of incoming edges. As these packets arrive at the node, they are stored in a buffer. As soon as the node is presented with a transmission opportunity, an outgoing packet  $\mathbf{y}(e)$  is created to be transmitted on the outgoing edges *e*. This outgoing packet is a linear combination of the packets present in the coding buffer of the node:

$$\mathbf{y}(e) = \sum_{e'} \boldsymbol{\beta}_{e}(e') \mathbf{y}(e')$$
(2)

where  $\beta_e$  is the local encoding vector of packet y(e).

The aim of this method is to employ RLNC as far as possible in the network, while still ultimately implementing a low complexity BP decoding algorithm. The BP decoding algorithm relies on the statistical properties of the encoded packets collected from the incoming edges at a network receiver node, which are in the form

$$\mathbf{y}(e') = \sum_{i=1}^{n} \boldsymbol{g}_{e'} \boldsymbol{x}_i \tag{3}$$

with  $g_{e'}$  the global encoding vector of received encoded packet y(e').

When the degrees of the received packets  $d[y(e'_u)]$  adhere to the RS distribution, BP can be implemented successfully.

In H-LTNC, before an encoded packet can be transmitted from one node to another, a connection is established between the neighbouring nodes. The receiving node transmits a message via the feedback channel stating whether it is a receiver node or not. Based on the feedback information, each node is categorised as a *random coding node* or *fountain coding node* and then proceeds with the suitable encoding algorithm.

#### A. Random coding nodes

When the connection established by an intermediate node is not with a receiver, the node implements low complexity RLNC for packet encoding of order  $\mathcal{O}(mb)$ . The local encoding vectors  $\boldsymbol{\beta}_{e}$  as shown in (2) are chosen randomly and independently from  $\mathbb{F}_{q}$  to construct an encoded packet  $\boldsymbol{y}(e)$  of a random degree  $d[\boldsymbol{y}(e)]$ .

# B. Fountain coding nodes

When the connection established by an intermediate node is with a receiver node, the encoding node applies a different encoding procedure so that the receiver node receives packets encoded according to the RS degree distribution. This method is formally presented in [7].

Firstly, the receiver node draws a target degree  $d_T$  from the RS distribution and communicates this value to the fountain coding node. The fountain coding node then examines the encoded packets  $y(e'_u)$  in its buffer and the degrees of the packets in the buffer are determined  $d[y(e'_u)]$ .

If a packet of the target degree  $d_T$  is present in the buffer it is selected as the new outgoing packet where

$$\mathbf{y}(e) = \mathbf{y}(e'_i) \tag{4}$$

and  $d[\mathbf{y}(e'_i)] = d_T$ . Thus the node only acts as a forwarding node and runs an algorithm of order  $\mathcal{O}(mb)$ .

If there is no packet of  $d_T$  in the buffer of the node, packets whose linear combination can produce a packet where  $d[\mathbf{y}(e)] = d_T$  are selected for encoding of  $\mathbf{y}(e)$ . When the target degree  $d_T$  cannot be reached, the packet with the closest degree to  $d_T$  is used. This encoding method is complex and scales exponentially.

This encoding algorithm employed at the fountain coding nodes enables the use of BP decoding at the receiver nodes due to the arriving packets being from the RS degree distribution. The standard method for BP decoding described in Section II B is implemented at the receiver nodes for decoding.

# IV. ENHANCED H-LTNC

In [7] it was shown that the H-LTNC method enables the successful use of BP decoding at the receiver nodes. The disadvantage of this method, however, was an increase in the additional packets needed to be collected by the receiver nodes before BP decoding could be completed successfully. This resulted in a longer decoding delay at the receiver nodes when compared to LTNC where all nodes are forced to encode packets according to the RS degree distribution.

The reason for the requirement of more additional packets and the longer decoding delay was a result of a received degree distribution that does not match that of the required RS distribution. The encoding method employed at the fountain coding nodes was not optimally constructed in order to produce the required distribution, resulting in sub optimal decoding. We now present two modifications to the intermediate network nodes to ensure the accurate encoding of packets of the needed target degrees. This optimisation reduces the number of additional packets required for decoding which in turn will render minimum decoding delays.

#### A. Sparse RLNC

The first improvement made to the H-LTNC method is to allow random coding nodes to employ sparse RLNC. Previously in the random coding nodes, packets were encoded randomly, but non-sparse. The probability of successful decoding for sparse RLNC is comparable to that of traditional RLNC when coding is done in a large finite field  $\mathbb{F}_q$  and the density of non-zero symbols in the global encoding vectors  $g_{e'}$  are greater than a certain threshold value [11].

In [7] when the fountain coding nodes received non-sparse packets to encode a packet of a low degree, the target degree was frequently not attainable. This interfered with the statistical properties of the packets needed for BP decoding.

In our network scenario the fountain coding nodes are required to construct packets of mostly low degrees adhering to the RS degree distribution. As shown in (2), the local encoding vector  $\beta_e$  for each encoded packet y(e) formed is chosen from a sufficiently large finite field  $\mathbb{F}_q$ . As an improvement, the encoding vectors are chosen to be sparse so that the average degrees of the encoded packets remain low. Thus when fountain coding nodes receive encoded packets of relatively low degrees, the construction of a packet of a low degree from the RS degree distribution is simplified greatly and a packet of the target degree can be constructed successfully with high probability.

At the random coding nodes when RLNC are performed with sparse linear combinations the encoding complexity at the nodes is also reduced.

## B. Buffer flushing policy

In a wireless network environment the buffers of the

intermediate nodes are flushed periodically according to a flushing policy [10]. Thus packets received at the incoming edges of a node are stored in the buffer and then flushed from it after a certain time has passed. This allows for the periodic construction of new encoded packets consisting of possibly new source packets.

In our network environment modelled by a random geometric graph (RGG) with R nodes and a minimum cut between source and receiver nodes of min-cut  $(s, z) \ge n$ , the average number of incoming edges per intermediate nodes are  $|e'|_{ave} = \sqrt{R}$ . In the previous work done in [7] the flushing policy of the network was set to flush the nodes' buffers at intervals equating to the reception of approximately  $\sqrt{R}$ packets. Thus each node must construct a new encoded packet from approximately  $\sqrt{R}$  received packets. For the fountain coding nodes that must construct a packet of a specific degree, the limited number of packets in its buffer can limit the success of packet encoding. Adjusting the flushing policy of these nodes to flush incoming packets at less frequent intervals, the buffers would contain more packets. This gives each fountain coding node a wider selection of packets which would enable it to construct a packet of a specific degree more accurately.

# V. SIMULATION RESULTS

In this section we evaluate the BP decoding performance when different encoding methods are implemented in the RLNC network environment. We evaluate the decoding delay, the received degree distribution and the encoding complexity for each encoding method used.

As discussed in Section IV *B*, we consider a network environment that can be modelled by a random geometric graph (RGG) with R = 100 nodes and a single source *s* and receiver node *z*. The minimum cut between source and receiver nodes is min-cut  $(s, z) \ge n$ . The data transmitted by the source to the receiver consists of approximately 10000 packets in the finite field  $\mathbb{F}_{2^8}$ . These packets are divided into *n* transmission packets  $\{x_i\}_{i=1}^n$  of size *m*. We conducted 1000 Monte-Carlo simulations for various values of *n*. This experimental setup is based on that of [10].

We consider a multicast communication network and assume a feedback channel allowing communication between nodes regarding connectivity to receiver nodes. The receiver node implements low complexity BP decoding.

#### A. Decoding delay

Decoding delay can be seen as the elapsed time between the reception of a packet at a receiver node and the decoding thereof [12]. When packets are received that adhere to the RS distribution, the decoding delay should be equal to zero as this distribution ensures optimal decoding.

We denote t as the timestep of the simulation when z obtains a new packet from the network. We denote the global

rank of the network as  $R_n$ , which is equal to the number of source packets n. The rank present at receiver node z at time t is defined as  $R_z(t)$ . The source packets decodable by node z are defined as *effective packets* and the total number of effective packets at z up to time t is denoted as  $E_z(t)$ .

Fig. 1 shows the normalised  $E_z(t)/R_n$  decoding curves for H-LTNC, EH-LTNC and a simplified version of LTNC for n = 35.





The curve  $R_z(t)/R_n$  shows the normalised value of the rank available at z, which expresses the total number of source packets possibly decodable at time t. This curve gives the lower limit of decoding delay for any system at time t.

The graph shows that EH-LTNC renders a large improvement in the decoding delay compared to H-LTNC. Where the H-LTNC method has an approximate decoding delay of t = 10, the decoding delay of EH-LTNC is approximately zero. This shows that EH-LTNC is an accurate encoding method to produce packets suitable for BP decoding in a RLNC network. When compared to the LTNC method, the results are approximately the same.

#### B. Received degree distributions

Next we evaluate the degree distributions of the packets obtained by the receiver nodes for each encoding method. Fig. 2*a* shows the RS degree distribution for n = 35 where c = 0.2 and  $\delta = 0.5$ . The degree distribution of the received packets are shown in Fig. 2 *b*,*c*,*d* for the implementation of H-LTNC, EH-LTNC and a simplified version of LTNC.

It can be seen that H-LTNC produces a degree distribution that is not comparable to the RS distribution. EH-LTNC produces packets with degrees that are comparable to the RS distribution, which shows that the improvement of sparse encoding and extended flushing times allow for the accurate encoding of packets from the RS distribution. This corresponds to the results shown in Fig. 1 that the production of packets of the RS distribution is done accurately and that packets are decoded successfully via BP decoding with minimal decoding delay.

The results of EH-LTNC are also comparable to the computationally complex LTNC method which supports the findings depicted in Fig. 1.



Figure 2: Received degree distributions

#### C. Encoding complexity

The EH-LTNC method produces the same results for decoding as the LTNC method. The advantage of the presented method, however, is that it has a lower encoding complexity than LTNC.

The LTNC method requires a complex encoding algorithm at all the intermediate network nodes. In our network environment with a min-cut  $(s, z) \ge n$ , the effective number of incoming edges of a receiver node are |e'| = n. Thus most of the intermediate nodes perform RLNC and only nodes connected to a receiver implement the complex encoding algorithm. With a network of *R* nodes and min-cut  $(s, z) \ge n$ , approximately *n* nodes are fountain coding nodes and (R - n)are random coding nodes. Thus the relationship between network size *R* and *n* determines the encoding advantage of EH-LTNC over that of LTNC.

#### VI. CONCLUSION

In this paper we presented improvements to the H-LTNC method presented in [7]. The Enhanced H-LTNC method allows for the use of fountain codes in conjunction with RLNC in a communication network to allow low complexity BP decoding in a network that communicates at multicast capacity. The presented method allows low complexity linear

encoding at most of the intermediate network nodes as well as a low decoding complexity with the use of BP decoding.

We showed that when RLNC are performed with sparse linear combinations and packet buffers are flushed at less frequent intervals, EH-LTNC renders a decoding delay which approaches that of the lower limit. This is because the method allows for the accurate encoding of packets that closely resembles the RS degree distribution enabling receiver nodes to successfully implement BP decoding. The EH-LTNC method retains the low complexity encoding of H-LTNC.

The presented method has the largest encoding complexity advantage in networks where the ratio between min-cut and number of nodes (n/R) is small. In wireless sensor networks information packets are traditionally small and may only consist of a few bits [13] where the data is transmitted to a sink via a group of intermediate nodes. Thus the wireless sensor network environment is suitable for the implementation of this method.

#### VII. ACKNOWLEDGEMENT

This work was completed with funding from the Telkom Centre of Excellence at the North-West University, Potchefstroom Campus.

#### VIII. REFERENCES

- M. Luby, "LT codes," Foundations of Computer Science, in *Proc. The* 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 271-280, 2002.
- [2] A. Shokrollahi, "Raptor codes," *IEEE/ACM Trans. Netw., IEEE Press*, Vol. 14, pp. 2551-2567, 2006
- [3] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *IEEE Int. Symp. Information Theory*, Yokohama, pp. 442, July 2003.
- [4] H. Wang, J. Goseling, J.H. Weber, "Network coded flooding," Master's thesis, Dept of Telecommunic., Delft University of Technology, June 2009.
- [5] A. Hessler, T. Kakumaru, H. Perrey, D. Westhoff, "Data obfuscation with network coding," *Computer Communications*, Nov 2010.
- [6] S. Yang and R.W. Yeung, "Coding for a network coded fountain," in Proc. 2011 IEEE International Symposium on Information Theory (ISIT'11), Saint Petersburg, Russia, pp. 2583–2587, July 2011.
- [7] S. von Solms and A.S.J. Helberg, "The implementation of LT network coding in resource limited RLNC networks," in *Southern Africa Telecommunication Networks and Applications Conference (SATNAC)*, East London, South Africa, 2011.
- [8] M. Champel, K. Huguenin, A. Kermarrec, N. Le Scouarnec, "LT Network Codes", *Institut National de Recherch en Informatique et en Automatique (INRIA)*, November 2009.
- [9] R. Koetter and M. Medard, "Beyond routing: an algebraic approach to network coding," in *Proc. Twenty-First Annual Joint Conf. of the IEEE Computer and Communications Societies*, vol. 1, pp. 122-130, 2002.
- [10] P. A. Chou, Y. Wu, and K. Jain, "Practical network coding," *Allerton Conference on Communication, Control and Computing*, Monticello, IL, October 2003.
- [11] M. Wang and B. Li, "How Practical is Network Coding?," *IEEE International Workshop on Quality of Service 2006*, Yale University, Connecticut, June 19–21, 2006.
- [12] J. K. Sundararajan, D. Shah and M. Medard, "Online network coding for optimal throughput and delay – the three receiver case," in *Proc. Int. Symp. on Inform. Theory and its Applic.*, Aukland, New Zealand, 2008.

[13] M. Jafari, L. Keller, C. Fragouli and K. Argyraki, "Compressed Network Coding Vectors," in *Proc. IEEE International Symposium on Information Theory (ISIT 2009)*, Seoul, Korea, pp. 109-113, June 2009.

Suné von Solms was born in Port Elizabeth. She completed her Bachelor of Engineering degree in Computer and Electronic Engineering (Cum Laude) in 2007 and her Master's degree in Computer Engineering (Cum Laude) in 2009 at the North-West University, Potchefstroom campus. She is currently a Ph.D student in Computer Engineering for the Telkom COE at the North-West University.

Prof. Albert Helberg obtained the B.Eng (Cum Laude 1989), M.Eng (Cum Laude - 1991) and D.Eng (1993) degrees in electronic engineering from the Rand Afrikaans University. Prof. Albert Helberg is the research team leader in the TeleNet research group of the Telkom CoE at the North-West University and has filled several management positions in the Faculty, including Director of the School of Electrical, Electronic and Computer Engineering (2002-2006) and Acting Dean of the Faculty of Engineering (2009). His personal research interests include digital coding techniques for error correction.