

# Hamming error correction techniques for the improvement of robustness in networks using random network coding

S von Solms, ASJ Helberg  
School for Electric, Electronic and Computer Engineering  
North West University, Potchefstroom Campus  
Tel: (081) 299 1961, Fax: (081) 299 1977, E- mail: 12987611@nwu.ac.za

**Abstract – In this paper, we introduce an algorithm for error detection and correction in Random Network Coding. The introduced technique exploits the encoding characteristics of random network coding and uses the well known Hamming Code as a decoding algorithm. For a network where random network coding is applied, this technique can be a useful error detecting and correcting method that will improve the network’s robustness.**

**Index Terms— Error Correction, Hamming Code Network Coding, Random Network Coding, Robustness.**

## I. INTRODUCTION

The concept of network coding was first introduced by Ahlswede, Cai, Li and Yeung in 2000 [1]. Instead of simply forwarding data in a network, as in traditional routing, they proposed that nodes may recombine several input packets into one or more output packets by performing a logical x-or operation on it. The concept of Random Network Coding was introduced by Ho, Koetter, Médard, Karger, and Effros in [2].

Their approach provides an improvement in robustness [11] where the success of information reception does not depend on receiving packets that contain specific information, but on receiving enough independent packets [12].

Random Network Coding, described in [2] works as follows: All the nodes in the network, except the receiver node, perform independent random linear mappings of their inputs. This creates independent linear combinations that are then forwarded to the next node, where once again random linear combinations are made from all inputs to the node, as shown Figure 1. The outputs are chosen independently and randomly and must be non- zero.

The receiver node of the network then obtains a series of independent linear combinations which it can use to decode the transmitted data. The receiver node of the network only needs to know the overall linear combination of the source processes in each of the incoming packets. This information is provided by a coding vector that is included in each message overhead [14].

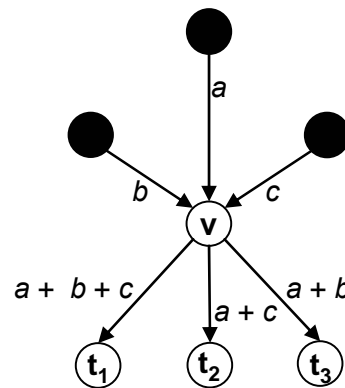


Figure 1. Random network coding subnetwork

The disadvantage of this characteristic is the fact that the network can be very sensitive to errors [4, 5]. A single error packet has the potential to infect the whole network and corrupt all the packets used by the receiver for decoding.

When a corrupt packet is linearly combined with legitimate packets, it can affect all the information gathered in that packet. Another problem that may occur is that an insufficient number of packets containing information of a single source node may reach the receiver, therefore preventing the receiver to decode the correct source messages.

It is possible to address these shortcomings by implementing error correction in the network. An error correction code will be able to correct and detect data packets corrupted due to additive errors.

Yeung and Cai [15] showed that network error correction can be executed in network coding, by using classical coding theory. Their study aims to manage errors that occur in networks by detecting and correcting it. This will enable the receiver of the network to receive the correct information sent over the network.

Koetter et al [10] as well as Silva et al [5, 6] introduced different approaches to error correction in Random Network Coding. Their different approaches both consider networks where the encoded vector are generated by the source node and sent through the network. As the information vector is

sent through the network, the packets may receive an error over any link in the network. These errors can be corrected by implementing the error correcting methods at the receiver end.

It can easily be seen that the topic of error correction in Random Network Coding is very popular. However, the construction of the error correcting codes are in a concatenated form where encoding takes place before transmission. The natural encoding capabilities of Random Network Coding for error correction codes are not considered. This fact opens up great possibilities in the field of network error detection and correction in Random Network Coding.

Several advantages can be achieved by using the natural encoding capabilities of Random Network Coding:

1. *Saving in Bandwidth*: Encoding the information within the network, instead of transmitting the already encoded codeword over the network leads to a saving in network bandwidth [16].
2. *Reduction in Congestion*: This encoding approach reduces the number of transmissions in the network which leads to reduction in network congestion [17].
3. *Higher Throughput*: By reducing the number of transmissions in the network, a higher throughput can be obtained [12, 17].

These factors all contribute to improving the robustness of the network.

## II. ERROR CORRECTION

A well known single bit error correction code is the Hamming Code, invented by Richard Hamming [7]. Hamming Codes have a length  $n$ , where  $n = 2^m - 1$  and a dimension  $k$ , where  $k = 2^m - m - 1$  ( $m \geq 2$ ) [10].

The  $(n, k)$  Hamming Code produces  $n$  output bits out of  $k$  input bits. This code can detect and correct a single-bit error and detect (but not correct) up to two simultaneous bit errors [8].

The goal of the Hamming Code is to develop a set of logical parity bits so that errors (inverted bits) in data or parity bits can be detected and corrected.

In this paper, we present an algorithm for a network error correcting code based on the Hamming Code using random network coding. We provide an example where the  $(7, 4)$  Hamming Code is used.

We focus on multiple unicast networks where only the receiver node applies error correction techniques and the intermediate nodes only create linear combinations of the packets they received. Thus the operations of random network coding are not changed. The network itself acts as an encoder of the source information packets for the error correction to take place.

## III. MODEL

We adopt the notation used in [2, 4] of an acyclic network model. The network is represented by a directed graph  $G = (V, E)$ .  $V$  is the set of nodes in the network and  $E$  the set of edges in  $G$  which represents the communication channels.  $S = \{s_1, s_2, \dots, s_{|S|}\} \in V$  represents the source nodes and  $t \in V$  the sink node in the multiple unicast network.

An edge from node  $a$  to  $b$  is indicated by  $(a, b) \in E$ . Node  $a$  is called the input node of edge  $(a, b)$  and edge  $(a, b)$  is called the input edge of node  $b$ , while node  $b$  is called the output node of edge  $(a, b)$  and edge  $(a, b)$  is called the output edge of node  $a$ .

Each edge in the network has unit capacity; therefore it is able to transmit a single unit of information per unit time.

Random Network Coding is implemented where each network node randomly and independently selects coefficients from a finite field  $F_2$ . The receiver also receives the overall linear combination of the source processes resulting in each information bit.

Let  $\mathbf{X}(v) = \{X(v,1), X(v,2), \dots, X(v, \mu(v))\}$  be a collection of  $\mu(v)$  random linear combinations received by node  $v \in V$ . These linear combinations consist of the processes sent by the source nodes. We want to create another non-zero random linear combination,  $\mathbf{Z}(v)$ , out of a subset of the random linear combinations  $\mathbf{X}(v)$  and forward it to some different node  $v' \in V$ . This concept can be illustrated in Figure 2. This pattern is repeated until this information bit reaches the receiver.

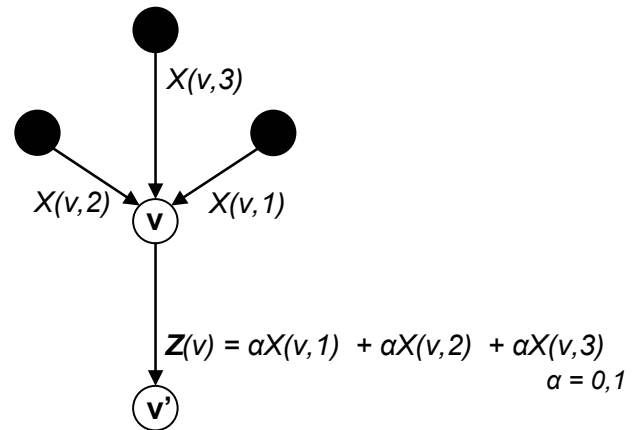


Figure 2. Random linear combinations in Random Network Coding

In this paper, a network is considered where the intermediate nodes do not correct errors, but only create the linear combinations of their inputs.

We assume that an error always occurs on an edge. When an error occurs on edge  $(a, b)$  the symbol sent by node  $a$  differs from the symbol received by node  $b$ . Only the receiver node  $t \in V$  will be able to detect and correct errors.

For the implementation of the Hamming Code on Random Network Coding, a few specific constraints must be put in place.

1. We consider a finite field  $F_2$ , where the information contained in each packet is only a single bit. This means that each node can only send a single bit over an edge in a single time unit.
2. The network source nodes,  $S = \{s_1, s_2, \dots, s_k\} \in V$  each contain one data bit  $\{d_1, \dots, d_k\}$ . The number of source nodes,  $k$ , depends on the  $(n, k)$  Hamming Code used.
3. No errors occur during the first set of transmissions, meaning that no errors occur on edges  $(s_i, b)$ ,  $i = 1, \dots, k$ .
4. For this method to be implemented successfully, only networks are considered where at most two errors occur.
5. This random network has a topology where the receiving node,  $t$ , receives  $n$  or more independent linear equations coming from  $n$  or more independent paths.

#### IV. HAMMING ERROR CORRECTION TECHNIQUES IN RANDOM NETWORK CODING

If we consider a network where no errors occur on the edges, the receiver has a very easy task decoding the information sent from the source nodes. The receiver only needs to find  $k$  linear independent polynomials,  $Y(r) = \{Y(r,1), Y(r,2), Y(r,3), \dots, Y(r,k)\}$  created in the network to decode the source information correctly.

A Random Network Coding environment is not necessarily error free. To ensure that the receiver still receives the correct information, the Hamming error correction technique is implemented by the receiver when decoding.

The network will act as an encoder, while the receiver will decode using the Hamming Code algorithm for the detection and correction of errors.

##### A. Encoding

The  $k$  source nodes  $S = \{s_1, s_2, \dots, s_k\}$  each send one data bit  $d = \{d_1, \dots, d_k\}$  into the network. As these data bits flow through the network, linear combinations are formed from them, shown in Figure 3. With the existence of  $n$  or more independent paths at the receiver, an error in one of the paths does not influence the correct flow of information in the other paths. The receiver waits until it receives  $n$  or more equations, which consist of two sets of linear independent equations of at least size  $k$  and  $(n-k)$  respectively,  $Y(r) = \{Y(r,1), Y(r,2), Y(r,3), \dots, Y(r,n)\}$  containing  $\{d_1, \dots, d_k\}$ .

Firstly, the  $k$  data bits  $\{d_1, \dots, d_k\}$  are decoded by the set of  $k$  linear independent equations,  $Y(r) = \{Y(r,1), Y(r,2), Y(r,3), \dots, Y(r,k)\}$  received.

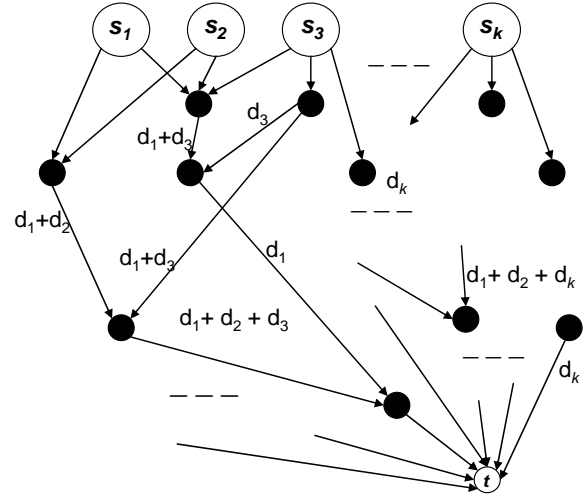


Figure 3. A network using Random Network Coding

The parity bits needed for the Hamming Code consist of the set of  $l$  linear independent equations,  $Y(r) = \{Y(r,k+1), Y(r,k+2), \dots, Y(r,n)\}$ , where  $l = n - k$ . For the parity bits to provide the platform for error detection and correction, they must logically relate to the data bits.

This relationship can be expressed as follows [13]:

$$\begin{aligned}
 p_1 &= z_{11}d_1 + z_{12}d_2 + \dots + z_{1k}d_k \\
 p_2 &= z_{21}d_1 + z_{22}d_2 + \dots + z_{2k}d_k \\
 &\vdots \\
 &\vdots \\
 p_l &= z_{l1}d_1 + z_{l2}d_2 + \dots + z_{lk}d_k \quad \dots (1)
 \end{aligned}$$

where  $z = 0, 1$

Each parity bit must consist of a linear combination of  $k - l$  data bits in order to form a legitimate set of parity bits.

These parity and data bits form a specific sequence, called the codeword  $r$ . This codeword will be used in the process of detecting and correcting errors in the network.

In Hamming Codes, there exists  $m = 2^k$  valid codewords which are expressed in the form  $r = \{d_1 d_2 d_3 \dots d_k p_1 p_2 \dots p_l\}$  [13].

#### V. EXAMPLE

An example is given to illustrate the concept of error detection and correction techniques through the use of a Hamming Code in random network coding.

1. The Shortened Hamming Code or  $(7, 4)$  code will be used.
2. The network contains four source nodes,  $\{s_1, s_2, s_3, s_4\} \in V$  each containing one data bit  $\{d_1, d_2, d_3, d_4\}$ .
3. No errors occur during the first set of transmissions, meaning that no errors occur on edges  $(s_i, b)$ ,  $i = 1, \dots, 4$ .

4. This random network has a topology where the receiving node,  $t$ , receives two sets consisting of four and three independent linear equations each coming from seven or more independent paths.

Suppose the four data bits are sent from the source nodes over the randomized network. The receiver waits until it receives the two sets of linear independent equations. Because each equation is received from an independent path, a single error in one path does not affect the other equations received from different paths.

The first set of four linear equations is used to decode the four data bits sent from the source nodes  $\{d_1, d_2, d_3, d_4\}$ . The other set of three equations are used to calculate the parity bits. They are expressed as follows:

$$\begin{aligned} p_1 &= z_{11}d_1 + z_{12}d_2 + z_{13}d_3 + z_{14}d_4 \\ p_2 &= z_{21}d_1 + z_{22}d_2 + z_{23}d_3 + z_{24}d_4 \\ p_3 &= z_{31}d_1 + z_{32}d_2 + z_{33}d_3 + z_{34}d_4 \end{aligned} \quad \dots (2)$$

The sequence of data and parity bits forms the codeword  $\mathbf{r} = \{d_1, d_2, d_3, d_4, p_1, p_2, p_3\}$ .  $\square$

#### A. Error detection and correction

To check if an error has occurred in the network, a parity check matrix  $\mathbf{H}$  must be generated. Every  $(n, k)$  code has an associated  $(n - k) \times n$  parity check matrix  $\mathbf{H}$ . This matrix has the property of  $\mathbf{H}^* \mathbf{r} = \mathbf{0}$ , where  $\mathbf{r}$  is the codeword [10].

This parity matrix  $\mathbf{H}$  is not unique and must be generated every time a new codeword is formed. The linear equations used for calculating the parity bits provide the information needed to generate  $\mathbf{H}$ .

A  $\mathbf{Z}$  matrix is defined:

$$\mathbf{Z} = \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1k} \\ z_{21} & z_{22} & \dots & z_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ z_{l1} & z_{l2} & \dots & z_{lk} \end{pmatrix} \quad \dots (3)$$

The  $\mathbf{H}$  matrix consists of the  $\mathbf{Z}$  matrix, as well as the identity matrix  $\mathbf{I}$  so that  $\mathbf{H} = (\mathbf{Z} | \mathbf{I})$ .

$$\mathbf{H} = \begin{pmatrix} z_{11} & z_{12} & \dots & z_{1k} & 1 & 0 & \dots & 0 \\ z_{21} & z_{22} & \dots & z_{2k} & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ z_{l1} & z_{l2} & \dots & z_{lk} & 0 & 0 & \dots & 1 \end{pmatrix} \quad \dots (4)$$

All parity check matrices' columns are binary representations of the numbers, though not in order, from 1 through to  $n$ .

The receiver must multiply (modulo 2)  $\mathbf{H}$  and  $\mathbf{r}$  to obtain a syndrome vector  $\mathbf{z}$ , which indicates if an error has

occurred. This vector also indicates which codeword bit has been logically inverted in the network (additive error).

In an error free network, the syndrome vector  $\mathbf{z} = \mathbf{0}$  to indicate that no bit in the codeword is incorrect and that  $\mathbf{r}_{correct} = \mathbf{r}$ . In a network where a single error has occurred, the syndrome vector will be equivalent to one of the columns of the  $\mathbf{H}$  matrix. The error position  $i$  corresponds to column  $i$  of  $\mathbf{H}$  and  $\mathbf{r}_{correct} = \mathbf{r} + e_i$  where  $e_i$  is a zero vector except for a 1 in position  $i$ . [10, 13]

In other words, the bit error can easily be detected by comparing the syndrome vector to the columns of  $\mathbf{H}$ . Once the incorrect bit is determined, the received codeword can be corrected by simply inverting the erroneous bit.

When the correct codeword has been found, the codeword can simply be decoded by using the decoding algorithm of the Hamming Code [10].

#### B. Multiple bit errors

As shown above, this application of the Hamming Code works effectively for any single bit error correction in random network coding. This Hamming Code application can, however, also be used for single and double bit error detection.

When we multiply the codeword received with the parity check matrix  $\mathbf{H}$ , the syndrome vector  $\mathbf{v}$ , will be non-zero whenever errors have occurred.

This method will only tell us if a single or double bit error has occurred, but cannot assist us in the correction of a double bit error, because we cannot distinguish between single or double bit errors.

## VI. MAIN RESULTS

By implementing this error correction method in Random Network Coding, many advantages are achieved.

*a) No increase in bandwidth:* To correct a potential error,  $n$  codeword bits are needed, but only  $k$  ( $k < n$ ) data bits are sent into the network from the source nodes. The other  $(n - k)$  bits are generated by the network. By using the network itself as an encoder, the bandwidth usage (information rate) remains the same, although up to double the information can be received. In other words, the network itself provides the redundant information required to perform error correction and detection.

This saving of bandwidth can clearly be seen in the Hamming (7, 4) example in Figure 4 where (a) is a network where the network is not used as an encoder and (b) where the encoding takes place in the network itself.

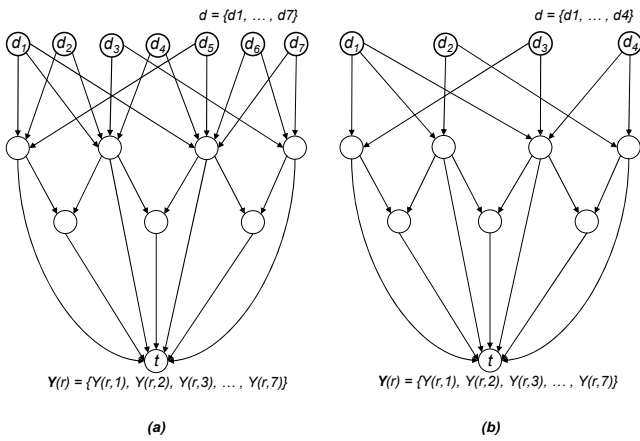


Figure 4. (7, 4) Hamming network (example)

b) *Congestion and Throughput:* It can also be seen in Figure 4 that this coding method reduces the number of transmissions in the network. This reduces network congestion and accordingly produces a higher throughput [17].

c) *Single error detection and correction:* The detection and correction of a single error in a network can be ensured when enough linear independent equations are received from the network to create a codeword. An error occurrence in any of the independent paths of the network can be detected and corrected by this algorithm.

For a network where certain constraints have been placed, this technique can be a useful error correcting method. With no control over the network, the receiver is not guaranteed to receive enough linear independent equations to ensure that this method works. With Random Network Coding, the probability of receiving linear dependant equations becomes considerably small when the field size is large [6]. Therefore, a wider network with a larger depth and high connectivity will increase the probability of success of this method.

d) *Double error detection:* When two single bit errors occur, this method will enable us to detect it. These errors cannot be corrected, but the algorithm will enable us to detect erroneous information.

## VII. CONCLUSION

In this paper, we have presented an algorithm for error detection and correction in random network coding. The introduced technique exploits the encoding characteristics of random network coding and uses the well known Hamming Code as a decoding algorithm.

This technique introduces the concept of the random network coding as the generator of a forward error correction code. Further research should be carried out on how to use packets in the network, instead of bits, as well as implementing other forward error correction codes in random network coding.

## REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li and R. W. Yeung, "Network information flow," *IEEE Trans. on Information Theory*, vol. 46, pp. 1204-1216, 2000.
- [2] T. Ho, R. Koetter, M. Médard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," in *Proc. IEEE Int. Symp. Information Theory*, Yokohama, 29 June-4 July 2003, p. 442.
- [3] T. Ho, M. Médard, R. Koetter, D. R. Karger, M. Effros, J. Shi, and B. Leong, "A random linear network coding approach to multicast," *IEEE Trans. Inform. Theory*, vol. 52, no. 10, pp. 4413-4430, Oct. 2006.
- [4] R. Koetter and M. Médard, "Beyond routing: An algebraic approach to network coding," *IEEE/ACM Transaction on Networking*, vol. 11, pp. 782-796, October 2003.
- [5] Danilo Silva and Frank R. Kschischang, "Using Rank-Metric Codes for Error Correction in Random Network Coding," *ISIT2007*, June 24 - June 29, 2007
- [6] Danilo Silva, Frank R. Kschischang, and Ralf Koetter "A Rank-Metric Approach to Error Control in Random Network Coding," *IEEE International Symposium on Information Theory*, Nice, France, June 2007
- [7] R. Hamming, "Error detecting and error correcting codes," *Bell Syst. Tech. Journal*, vol. 29, pp. 41-56, 1950.
- [8] David J.C. MacKay, "Information theory, inference and learning algorithms," Cambridge University Press, Cambridge, 2003
- [9] S. Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, vol. 49, p. 371, Feb. 2003.
- [10] T. K. Moon, "Error Correction Coding Mathematical Methods and Algorithms," John Wiley & Sons, Hoboken, NJ, USA, 2005
- [11] T. Ho, R. Koetter, M. Médard, M. Effros, J. Shi, and D. Karger, "Toward a random operation of networks," submitted to *IEEE Trans. Inform. Theory*, 2004
- [12] C. Fragouli, J. Le Boudec, and J. Widmer, "Network coding: an instant primer," *SIGCOMM Comput. Commun. Rev.*, 36(1):63-68, 2006.
- [13] A. Leon-Garcia, I. Widjaja, "Communication Networks. Fundamental Concepts and Key Architectures," McGraw-Hill Professional, 2003.
- [14] R. Koetter and F. Kschischang, "Coding for errors and erasures in random network coding", in *International Symposium on Information Theory (ISIT)*, June 2007.
- [15] N. Cai and R. W. Yeung, "Network coding and error correction," *Proceedings of IEEE Information Theory Workshop*, pages 119-122, October 2002.
- [16] D. Axel, "Network Coding: an Overview," *Institute for Communications Engineering (LNT)*, January 2005, pp. 1-19.
- [17] S. Katti, H. Rahul, W. Hu, D. Katabi, M. Médard, and J. Crowcroft, "Xors in the air: Practical wireless network coding," in *Proc. of ACM Sigcomm 2006*, Pisa, Italy, Sept. 2006.

**Suné von Solms** is born in Port Elizabeth on 26 January 1985.

She completed her Bachelor of Engineering degree in Computer and Electronic Engineering in 2007 at the North West University, Potchefstroom campus.

She is currently a Master's degree student in Computer Engineering for the Telkom COE at the North West University.