Error Correction with the Implicit Encoding Capability of Random Network Coding

Suné von Solms, Magdalena J. Grobler, and Albert S.J. Helberg

School for Electric, Electronic and Computer Engineering,
North West University – Potchefstroom Campus, Potchefstroom, South Africa
{sune.vonsolms,leenta.grobler,albert.helberg}@nwu.ac.za

Abstract. In this paper, we present a technique for a network error correcting code using Random Network Coding. We introduce a novel error correction scheme that uses the implicit encoding capability of Random Network Coding. This scheme does not add redundancy to the data prior to transmission, but ads redundancy based on information already contained in the network. Random Network Coding within a large network generates enough redundant information to perform error correction on transmitted data.

Keywords: Error Correction, Network Coding, Random Network Coding.

1 Introduction

The concept of Network Coding was first introduced by Ahlswede et al. 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. In [1], the combinations formed by the nodes are based on a specific topology.

The concept of Random Network Coding was introduced by Ho et al. in [2]. Their approach provides an improvement in robustness [5] where the success of information reception does not depend on receiving packets that contain the specific transmitted information, but on receiving enough linearly independent packets [6]. Knowledge of the linear combinations of the information contained in each data packet is used to solve a set of simultaneous equations to obtain the transmitted data.

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 formed from the inputs of the node. 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 has to wait a certain amount of time in order to receive a set of equations that can be used to decode the transmitted message. 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 [7].

J. Zheng et al. (Eds.): ADHOCNETS 2009, LNICST 28, pp. 704–717, 2010.

A Random Network Coding environment is not necessarily error free. This disadvantage means that the network can be very sensitive to errors [3]. A single error packet has the potential to infect the whole network and corrupt other packets used by the receiver for decoding. When a corrupted packet is linearly combined with legitimate packets, it can corrupt all the information contained in that packet.

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. This will improve the robustness of the network where we will be able to obtain the correct information, even when only partially correct information is received.

Yeung and Cai [8] constructed such a linear network code with error-correcting capabilities. In erroneous network channels, Network Error Correction can be applied so that errors occurring in the network can be detected and corrected. Lower and upper bounds are also defined for the specific error-correcting capability of the code. Jaggi et al. [10] addressed the problem of error correction by adding redundancy to the source information that satisfies certain constraints and achieves optimal rates. This redundancy will enable the receiver node to correct network errors.

It can be seen that the topic of error correction in Random Network Coding is of current interest. However, the construction of the error correcting codes is in a concatenated form where error correcting encoding takes place prior to transmission. The implicit encoding capabilities of Random Network Coding for error correction codes are not considered. This fact opens up possibilities in the field of network error detection and correction in Random Network Coding which we aim to exploit.

In this paper, we aim to exploit the implicit encoding capabilities of a network implementing Random Network Coding. This method will encode the information sent by the source within the network, instead of transmitting a codeword encoded at the source.

2 Network Model

We adopt the notation used in [2], [3] 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. $S \in V$ represents the source node and $T \in V$ the sink node in the multiple unicast network. The source node sends messages selected from a source alphabet, Z. Let X be the finite set of code alphabet for the network where $X \in Z$ in a finite field F_q. Each edge, $(a, b) \in E$, in the network has unit capacity; therefore it is able to transmit a single unit of information per unit time.

Definition 1: Information Rate (R) is a measure of the average amount of information that is being carried by a symbol [1]. Information Rate is represented by information symbols sent from the source node into the network, and channel symbols received by the receiver from the network.

$$R = \frac{information\ symbols}{channel\ symbols}$$

Theorem 1: Min-Cut Max-Flow: A network with a single source, S, and receiver, T, is given. This connection can be described as $c = \left(S,\ T,\ X(S,\ T)\right)$. This network problem can only be solved if and only if the rate of the connection *R(c)* is less than or equal to the minimum value of all the cuts between S and T [3]

$$maxflow(T) \geq R(c). \tag{1}$$

Fragouli et al. stated in [11] that when a network $G = (V,\ E)$ with node S ∈ V and any non-source node $T \in V$ has a min-cut between S and T of K and a connection of c; then the information can be sent from S to T at a maximum rate $R(c)$ of K. They prove that there exist exactly K edge-disjoint paths between S and T when the min-cut between them is K, therefore:

$$R(c) = K \tag{2}$$

and

$$maxflow(T) \geq K. \tag{3}$$

By the min-cut max-flow theorem, it can be seen that equation (3) is a required condition for any node T to solve the message sent from the source node. This means that for K independent information packets to be sent successfully to the receiver, K edge-disjoint paths must exist in the network that connects the source to the receiver.

According to [7], if K linearly independent packets are sent into the network, the min-cut between the source and the receiver in the network must be large enough to support their transmission. This means that the rate of information transmission between the source node and the receiver node is upper-bounded by the min-cut between the nodes in the network [10].

2.1 Random Network Coding

Assume that a packet contains a sequence of n symbols from the finite field F_q. Let $x_1, x_2, \dots, x_K$ be the K information packets (vectors of length n over finite field F_q) transmitted by the source node S into the network with a $min - cut \geq K$. Let $y_1, y_2, \dots, y_{K'}$ be the K' channel packets received by the receiver node [4].

Random Network Coding is implemented where each network node randomly and independently selects coefficients from a finite field F_q. As described in [12], each information packet y_i in the network formed are random linear combinations of $x_1, x_2, \dots, x_K$,

$$y = \sum_{k=1}^{K} \alpha_k \boldsymbol{x}_k \tag{4}$$

where α is called the global encoding vector of x. References [12] as well as [7] assume that this vector is sent along with x in its header. We will adopt the same assumption and send the global encoding vector inside the information packet.

Example 1: Suppose that the source node, S, sends K information packets into the network with a $min - cut \geq K$. Each information packet $x_i, i = 1,2, \dots, K$ has a

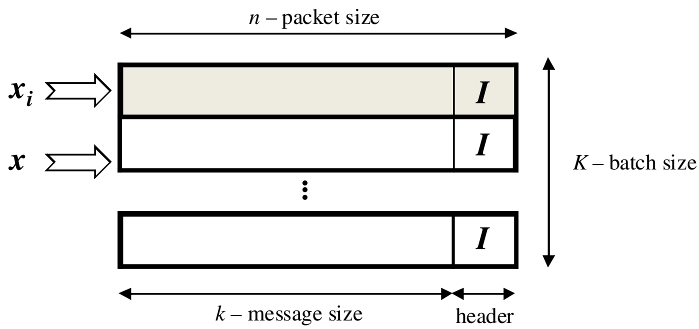

Fig. 1. Information Packets sent from the source through the network

length n that consists of a information message (length k) along with the header (length $n-k$), as can be seen in Fig 1.

The overhead (header) sent with the information packet has a size $n-k = K\, log_2\, q$ bits. This is negligibly small if the packet size, n, is sufficiently large [12].

Let the receiver obtain K' channel packets:

$$\begin{aligned} y_1 &= \alpha_1 x_1 \\ y_2 &= \alpha_2 x_2 \\ &\vdots \\ y_{K'} &= \alpha_3 x_K \end{aligned} \tag{5}$$

where $x_i, i = 1,2,\ldots,K$ are the K information packets sent from the source and α_i are random coefficients. Equation (5) can also be written as

$$y = \alpha x, \tag{6}$$

where $x = [x_{ij}]$ is the $K \times n$ transmitted array formed by stacking the information packets $x_1, x_2, \ldots, x_K$ as the rows of x, where the subscript of x_{ij} indicates the j'th entry of packet $x_i, i = 1,2,\ldots,K$. Also, $y = [y_{ij}]$ is the $K' \times n$ received array formed by stacking the received channel packets $y_1, y_2, \ldots, y_{K'}$ as the rows of y where the subscript of y_{ij} indicates the j'th entry of packet $y_i, i = 1,2,\ldots,K'$. α is a $K' \times K$ matrix over F_q corresponding to the overall transfer function of the network from the source to the receiver [4].

Linearly dependent packets (packets with linearly dependent global encoding vectors) are useless for the decoding of the channel messages at the receiver. The min-cut between the source node and receiver nodes must therefore be large enough to support the transmission of the K linearly independent packets. When the receiver receives K channel packets with linearly independent global encoding vectors, it will be able to decode the K message packets [12].

It can be clearly seen that the information rate of this network is

$$R = \frac{K}{K} = 1,$$

where K information symbols are sent into the network $(min - cut\, \geq\, K)$ and K channel symbols are received.

3 Error Correction in Random Network Coding: Traditional Method

We now assume that errors occur in the network. We assume that packet errors occur on the edge of the network. If K information packets are sent over the network, let z_k denote error packets applied to the packet $k \in \{1, \ldots, K\}$. Equation (6) then becomes

$$y = \sum_{k=1}^{K} \alpha_k x_k + \sum_{k=1}^{K} \beta_k z_k \tag{7}$$

or

$$y = \alpha x + \beta z, \tag{8}$$

where $z = [z_1^T, z_2^T, \ldots, z_K^T]$ is an array consisting of all the erroneous packets introduced in the network and β is the overall transfer matrix of these packets from the source to destination. If $z_k = 0$, no errors were applied to message packet $i \in \{1, \ldots, K\}$.

3.1 Redundant Symbols

Jaggi et al. described in [10] that the errors that occur in the network can be thought of as a second source. The information received at the receiver is linear combinations of the information of the source as well the error information. This can be seen in (7).

Reference [10] addressed the topic of extracting the source information from the received mixture of channel information and errors. He addressed this problem by adding redundancy to the source information that satisfies certain constraints. This information packet is constructed as in Fig. 2 [10]:

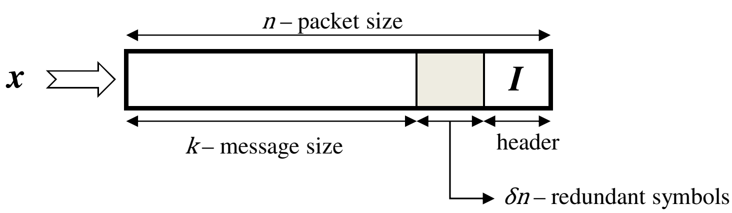

Fig. 2. Information Packet with redundancy symbols

Each packet contains a sequence of n symbols from the finite field F_q. Out of the n symbols in the information packet; δn symbols are redundancy added by the source. The δn redundant symbols are chosen as parity symbols in order for the receiver to decode the channel packet. Also included in the packet is the identity matrix, I, that acts as the global encoding vector by reflecting the linear combinations formed on the channel packet.

3.2 Redundant Packets

Definition 2: A network code is *t-error-correcting* if it can correct all γ-errors for $\gamma \leq t$, i.e., if the total number of errors in the network are at most *t*, then the source message can be recovered by the sink node $T \in V$ [8].

Definition 3: *A block code* is a rule for converting a sequence of source symbols of length K into a transmitted sequence of length N symbols [13].

Yeung and Cai [9] correct errors in network coding not by adding redundancy to each information packet sent, but by adding redundant packets at the source to be sent over the network. They use (N, K) linear block codes and consider these as a linear network code. The source node takes K information packets as its input and outputs N coded message packets, basically adding packets as parity.

We describe this process as follows: Let the information packets $\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_K$ (vectors of length n over finite field F_q) be the K message packets of $\boldsymbol{x} = [\boldsymbol{x}_{ij}]$ that must be transmitted over the network. The source node uses a (N, K) block code to encode these K information packets into N outgoing coded packets, denoted as $\boldsymbol{x'}_1, \boldsymbol{x'}_2, \dots, \boldsymbol{x'}_N$, where $N > K$ and

$$x'_i = \sum_{j=1}^{K} g_{ij}\boldsymbol{x}_j \quad (9)$$

These redundant packets are generated by using randomly generated coefficients g_{ij} from a finite field F_{2q}. The set of coefficients $g_{i1}, g_{i2}, \dots, g_{iK}$ can be referred to as the *encoding vector* for $\boldsymbol{x}_i$ [6] and are sent in the information packet as the overhead.

This method can be represented by Fig. 3.

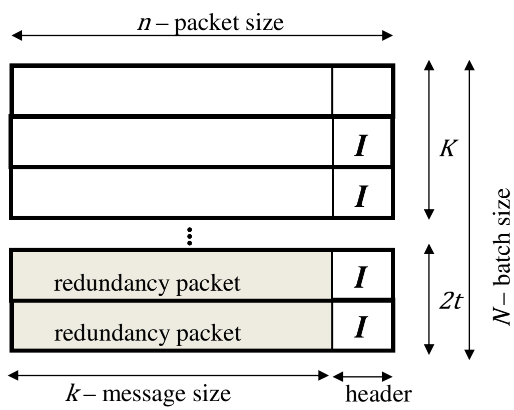

Fig. 3. Information packets with redundancy (parity) packets

Example 2: Assume in this example that the global encoding vector is sent along with the information packet, $\boldsymbol{x}$, in its header. This overhead, however, is negligible because the information packets are sufficiently large, therefore $n \approx k$.

Let $\boldsymbol{x} = \boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_K$ (vectors with the length of k symbols over finite field F_q) be the K information packets that must be transmitted over the network. The source node applies a (N, K) forward error correcting block code to linearly combine these K information packets into N coded packets, as can be seen in Fig. 3.

These packets are then transmitted by the source node, S, into the network with a $min - cut \geq N$. Let $\boldsymbol{y} = \boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_N$ be the N channel packets received by the receiver node, where

$$y = \sum_{n=1}^{N} \alpha_n x_n + \sum_{n=1}^{N} \beta_n z_n \tag{10}$$

The receiver only has to decode the (N, K) block code to successfully regenerate the sent data. The receiver can decode the message correctly when at most t errors occur, where $t = (N - K)/2$. This means that $\boldsymbol{x}$ is a classical error correcting code that can detect (N, K) and correct $(N, K)/2$ errors.

For Example 2, the information rate of the network ($min - cut \geq N$) is

$$R = \frac{N}{N} = 1,$$

where N information symbols are sent into the network and N channel symbols are received.

4 Error Correction in Random Network Coding: Proposed Method

Our proposed scheme for network error correction aims to eliminate:

1. the redundancy added to the source packet, or
2. the redundant packets added at the source.

By waiting for more channel packets, the receiver obtains additional information for decoding that may be used for error correction. Thus, the network acts as an error correction encoder. The coded information packets obtained by the receiver provides the redundancy required for error correction. This method differs from that in Section 3, because no redundant information is added at the source node.

4.1 Network Configurations

We introduce a method where the min-cut between the source and the receiver nodes in the network must be large enough to support the transmission of K linearly independent information packets, although N channel packets will be used by the receiver node.

According to [7], the receiver node would normally collect as many channel packets as possible in order to decode the source message. Because of network properties, such as the min-cut between source and receiver node, more than K linearly independent equations are redundant information.

We propose to use this redundant information received by the receiver node to apply error correction to the source message. This means that redundant information transmitted by the source node will no longer be necessary, because the network will transmit sufficient redundancy to the receiver for error correction.

K independent information packets are sent from the source node where the packets propagate through the network. The min-cut of the network remains K, therefore there exist K edge-disjoint paths. The receiver then obtains N channel packets. The extra $(N-K)$ received packets are what we intend to use in order to correct any possible errors. The values of N and K are determined by the specific (N,K) linear block code used. These N channel packets must consist of two sets of linearly independent packets, of size K and (N,K), respectively.

The first set of linearly independent packets is the traditional K packets needed to decode the sent message packets.

$$\begin{aligned} \boldsymbol{y_1} &= \alpha_1 \boldsymbol{x_1} \\ \boldsymbol{y_2} &= \alpha_2 \boldsymbol{x_2} \\ &\vdots \\ \boldsymbol{y_K} &= \alpha_3 \boldsymbol{x_K} \end{aligned} \tag{11}$$

or

$$y_{data} = \sum_{k=1}^{K} \boldsymbol{\alpha_k} x_k, \tag{12}$$

where $y_{data} = [y_{ij}]$ is a $K\ x\ n$ array formed by stacking the received message packets $\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_K$ as the rows of y_{data}, where the subscript of y_{ij} indicates the j'th entry of packet y_i , $i = 1, 2, \dots, K$.

The other set of linearly independent packets must be of size $(N-K)$ and will be used for error correction.

$$\begin{aligned} \boldsymbol{y_{1+K}} &= \alpha_1 \boldsymbol{x_{1+K}} \\ \boldsymbol{y_{2+K}} &= \alpha_2 \boldsymbol{x_{2+K}} \\ &\vdots \\ \boldsymbol{y_N} &= \alpha_3 \boldsymbol{x_N} \end{aligned} \tag{13}$$

or

$$y_{parity} = \sum_{k=1+K}^{N} \alpha_k x_k , \tag{14}$$

where $y_{parity} = \left[y_{ij}\right]$ is a $(N-K) \times n$array formed by stacking the received message packets $\boldsymbol{y}_{1+K}, \boldsymbol{y}_{2+K}, \dots, \boldsymbol{y}_N$ as the rows of y_{parity} where the subscript of y_{ij} indicates the j'th entry of packet y_i , $i = 1+K, 2+K, \dots, N$.

These redundant $(N-K)$ symbols are linear functions of the original K message packets and will act as the parity symbols providing the platform for error detection and correction. When the receiver obtains both sets of channel packets, it decodes the messages as a $(N-K)$ block code. The receiver can decode the message correctly when at most t errors occur, where $t = (N,K)/2$.

4.2 Example 3

Example 2 revisited: Assume in this example that the global encoding vector is sent along with information packet, $\boldsymbol{x}$, in its header. This overhead, however, is negligible because the packets are sufficiently large, therefore $n \approx k$.

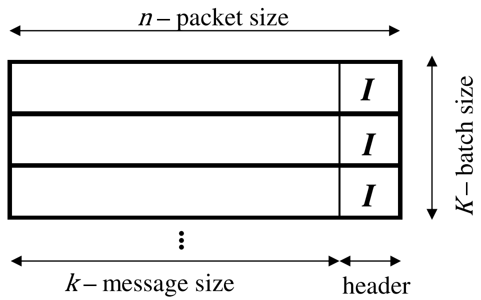

Fig. 4. Information packets without redundancy packets

Let $\boldsymbol{x} = \boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_K$ (vectors with the length of k symbols over finite field F_q) be the K information packets that must be transmitted over the network. The source node ***does not*** apply a (N, K) forward error correcting block code to the K information packets. The K information packets are transmitted by the source node, S, into the network with a $min - cut \geq K$. The K information packets propagate through the network; linear combinations are formed from them by intermediate nodes and the receiver waits until it receives N or more channel packets.

Let $\boldsymbol{y} = \boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_N$ be the N message packets received by the receiver node, where

$$y = \sum_{n=1}^{N} \alpha_n x_n + \sum_{n=1}^{N} \beta_n z_n \tag{15}$$

The receiver then only has to decode the (N, K) block code to successfully regenerate the sent data. The receiver can decode the message correctly when at most t errors occur, where $t = (N - K)/2$. This means that $\boldsymbol{x}$ is an error correcting code that can detect $(N - K)$ and correct $(N - K)/2$ errors.

It can be seen that exactly the same decoding process is used at the receiver end. The difference is that less information is injected into the network. The min-cut of the network is smaller and the information rate is

$$R = \frac{K}{N} < 1 \, , K < N$$

This method offers benefits in terms of energy efficiency, because less information is injected into the network: the source node transmits K packets instead of N, into a network with a $min - cut \geq K$, instead of $min - cut \geq N$. Example 3 can by summarized by Fig. 5:

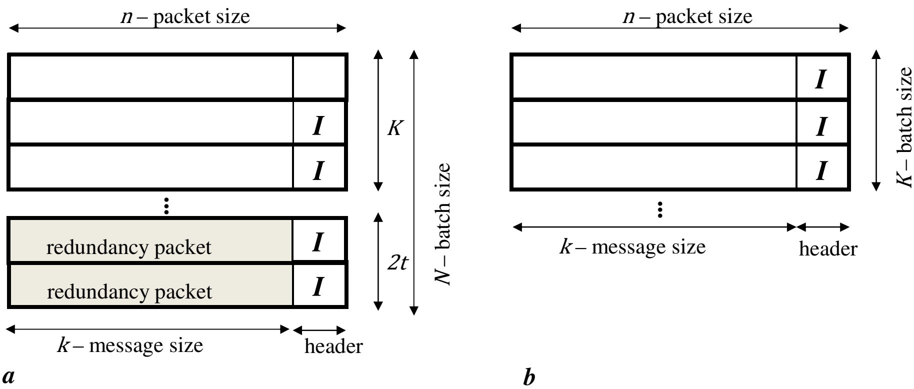

Fig. 5. Sent information messages for (a) concatenated and (b) Implicit Error Correction scheme

5 Analysis

In order to analyze the performance of the proposed and existing schemes, we investigate four aspects of the methods:

1. Probability of receiving enough valid parity packets for decoding
2. Complexity of the decoding algorithms
3. Time delay of the decoding algorithms
4. The error correction capability of the methods in a network with a specific min-cut.

For the proposed method, the possibility exists that the channel packets obtained by the receiver may not contain valid parity packets. This will prevent the receiver from decoding the information successfully. We investigate the probability of receiving a set of valid parity packets from a network so that this Implicit Error Correction method can be applied effectively.

We assume that the network under consideration is a non-cyclic, generic, random network as illustrated in Fig. 6. This network contains a single source node $S \in V$, and a single sink node, $T \in V$. The source node sends the K data packets to the network, which consist of p intermediate nodes. The nodes in the network can send multiple encoded packets to other nodes in the network, but only a selection of $q \leq p$ nodes are connected to the receiver. The receiver node only receives a single encoded packet from each of the q nodes.

We generated a set of 1000 randomly generated networks in order to analyze the Implicit Error Correction capabilities of it. These networks have the following properties:

1. Finite field, F_q.
2. Network size (number of nodes).
3. $Min - cut \geq K$.
4. Intermediate nodes consisting of p nodes, where q is connected to the receiver, $q \leq p$.
5. Each node in the network randomly and independently generates a linear combination of its inputs.

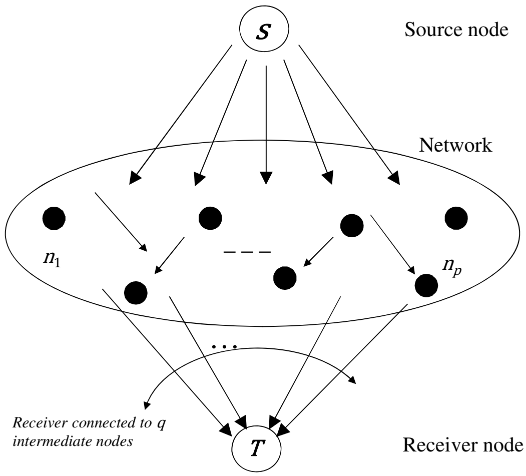

Fig 6. Random Network which contains K source nodes, p + q intermediate nodes (layer 1 and 2) and a single receiver

5.1 Probability of Decoding

The linear equations obtained by the receiver are evaluated to determine if the received combinations are valid sets of parity packets. The average percentage of valid sets received in each size network can be viewed in Fig. 7.

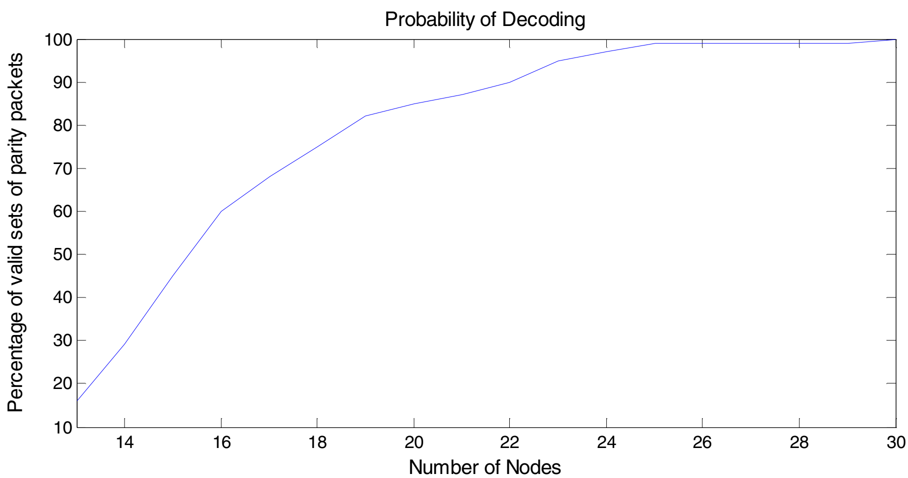

Fig. 6. Valid sets of Parity packets received in the network

It is clear that one can only expect to receive a guaranteed set of parity packets with a network containing about 30 nodes. We have chosen a 30-node network for all the following calculations and simulations, because a valid set of parity packets will be guaranteed, under the parameters for network size and complexity as discussed.

5.2 Discussion of Complexity

In the Error Correction method proposed in Section III, the time complexity of the decoding method is estimated to be $O(N)$, where all operations are in F_q. As discussed in Section IV.A, the receiver of the Implicit Error Correction method obtains N packets from the network. From these, data packets and valid parity packets must first be calculated, and then decoded. The estimated complexity of this decoding algorithm is $O(N^2 - K^2)$.

It can be seen that the computing complexity of this method is higher than that of the existing decoding method.

5.3 Time Delay

The cost of the higher computing complexity is an extended waiting period for decoding. Fig. 8 shows the time of the decoding algorithm of the Implicit Error Correcting method relative to the time for traditional decoding.

It is clearly visible that the time consumption of the implicit method is higher for decoding. However, for very large block codes ($n < 2000$), the time consumption for the Implicit Error Correcting method approaches that of the existing method.

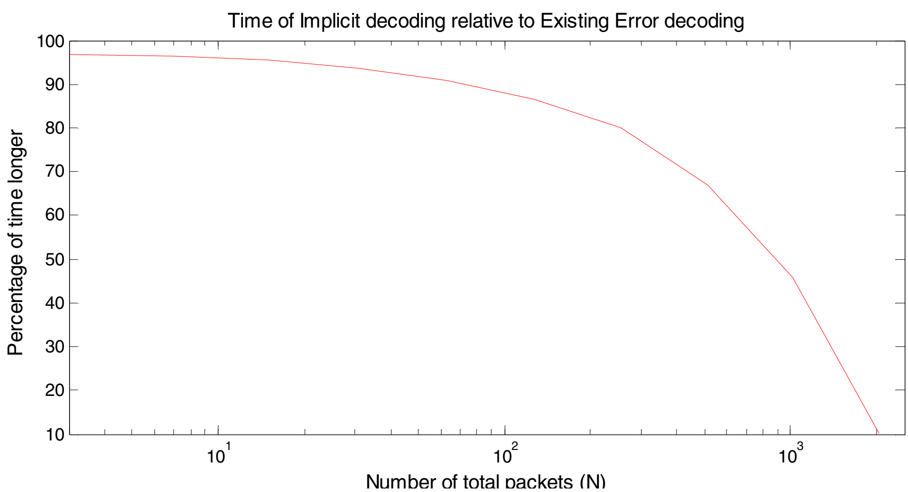

Fig. 7. Time of decoding algorithm of implicit error correcting scheme relative to the time for decoding described in Section 3

5.4 Error Correcting Capability

One advantage of the Implicit Error Correcting method is the fact that a t-error-correcting code can be applied successfully to a network with $min - cut \geq K$ instead of a network with $min - cut \geq N$, where $t = (N - K)/2$. This advantage can be seen in Fig. 9.

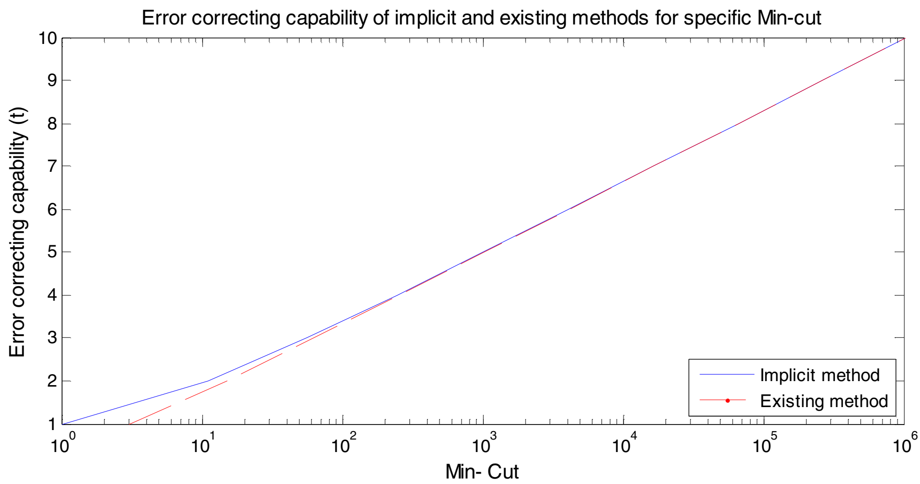

Fig. 8. Error correcting capability of the Implicit and Existing methods for a specified *t*- error correcting code

6 Advantages

We have introduced a novel scheme that makes effective use of information implicitly generated in the network to perform error correction. The redundant information needed for error correction is generated in the network, and not sent from the source node. This forward error correction method maps a set of information symbols to a set of code symbols resulting in an information rate of less than 1.

The biggest advantage achieved by using the implicit encoding capabilities of Random Network Coding is the fact that the source does not implement error correction. Only the receiver applies error correction codes. This means that the receiver can apply any error correction code it chooses (example: Hamming, Reed Solomon etc.) without informing the source. The receiver bases the decision purely on the information it receives. Another advantage of the Implicit Error Correcting method is that the scheme allows greater error correcting capability than the existing scheme for a network with the same min-cut, or vice versa. The requirements of the network are reduced, both in connectivity (min-cut) and bandwidth required.

The time consumption due to decoding complexity concerning this method is the biggest trade-off for the advantage of effective error correction in this scheme.

6 Conclusion

Encoding the information within the network, instead of transmitting the already encoded codeword over the network leads to a saving in network bandwidth. This method leads to an improvement in the network's information rate. The saving of energy may be of interest to energy constraint networks such as wireless sensor networks.

References

1. Ahlswede, R., Cai, N., Li, S.-Y.R., Yeung, R.W.: Network information flow. IEEE Trans. on Information Theory 46, 1204–1216 (2000)
2. Ho, T., Koetter, R., Médard, M., Karger, D.R., Effros, M.: 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 (2003)
3. Koetter, R., Médard, M.: Beyond routing: An algebraic approach to network coding. IEEE/ACM Transaction on Networking 11, 782–796 (2003)
4. Silva, D., Kschischang, F.R., Koetter, R.: A Rank-Metric approach to error control in random network coding. In: IEEE International Symposium on Information Theory, Nice, France (June 2007)
5. Ho, T., Koetter, R., Médard, M., Effros, M., Shi, J., Karger, D.: Toward a random operation of networks. Submitted to IEEE Trans. Inform. Theory (2004)
6. Fragouli, C., Le Boudec, J., Widmer, J.: Network coding: an instant primer. SIGCOMM Comput. Commun. Rev. 36(1), 63–68 (2006)
7. Koetter, R., Kschischang, F.: Coding for errors and erasures in random network coding. In: International Symposium on Information Theory (ISIT) (June 2007)
8. Cai, N., Yeung, R.W.: Network coding and error correction. In: Proceedings of IEEE Information Theory Workshop, October 2002, pp. 119–122 (2002)
9. Yeung, R.W., Cai, N.: Network Coding, Algebraic Coding, and Network Error Correction. In: Proc. Information Theory and Applications Workshop, La Jolla, CA (February 2006)
10. Jaggi, S., Langberg, M., Katti, S., Ho, T., Katabi, D., Médard, M.: Resilient network coding in the presence of Byzantine adversaries. In: Proc. 26th IEEE Int. Conf. on Computer Commun., Anchorage, AK, May 2007, pp. 616–624 (2007)
11. Fragouli, C., Soljanin, E.: Network Coding Fundamentals. Foundations and Trends in Networking 2(1) (2007)
12. Deb, S., Effros, M., Ho, T., Karger, D.R., Koetter, R., Lun, D.S., Médard, M., Ratnakar, N.: Network Coding for Wireless Applications; A Brief Tutorial. In: International Workshop on Wireless and Ad-hoc Networks (IWWAN) (May 2005)
13. MacKay, D.J.C.: Information Theory, Inference, and Learning Algorithms. Cambridge University Press, Cambridge (2003)