

# Protection against link errors and failures using network coding in overlay networks

Shizheng Li and Aditya Ramamoorthy

Department of Electrical and Computer Engineering, Iowa State University

Ames, IA, 50010

Email: {szli, adityar}@iastate.edu

**Abstract**—We propose a network-coding based scheme to protect multiple bidirectional unicast connections against adversarial errors and failures in a network. The end nodes of the bidirectional connections are connected by a set of shared protection paths that provide the redundancy required for protection. Suppose that  $n_e$  paths are corrupted by an omniscient, computationally unbounded adversary. Under our proposed protocol, the errors can be corrected at all the end nodes with  $4n_e$  protection paths. More generally, if there are  $n_e$  adversarial errors and  $n_f$  failures,  $4n_e + 2n_f$  protection paths are sufficient. The number of protection paths only depends on the number of errors and failures being protected against and is independent of the number of unicast connections.

## I. INTRODUCTION

The technique of network coding was recently applied to the problem of network protection in [1], [2] and was found to significantly reduce the amount of protection resources required as compared to conventional approaches. The protection strategies for link-disjoint connections in [1], [2] perform network coding over cycles or paths that are shared by the connections to be protected. The protection paths have a specific structure that induces a topology that allows the judicious use of network coding. These schemes deal exclusively with link failures and assume that each node knows the location of the failures at the time of decoding. In this work we are interested in the more general problem of protection against errors in the specific network topology that is induced by the protection paths considered in [1], [2]. We consider adversarial errors, such that an adversary may be limited in the number of links she can control, but for those links, she can corrupt the transmission in an arbitrary manner.

Several network error correction coding schemes have been proposed in the literature [3], [4], [5], [6]. However, these schemes work in the context of network-coded multicast connections.

In this work we attempt to protect multiple unicast connections using network coding, where the topology is constrained to be such that each individual unicast operates over a single primary path and the protection paths pass through the end nodes of each unicast connection (see Figure 1). This model is well motivated practically and numerous protection strategies have worked under this abstraction.

Our work is a significant generalization of [2]. We assume an error model, under which the adversary has unlimited

computational power and full knowledge of all details of the protocol (encoding algorithms, coefficients, etc.) and has no secrets hidden from her. The adversary changes data units on several paths, which may be primary paths or protection paths. The number of errors equals the number of paths the adversary attacks. If multiple paths share one link and the adversary controls that link, it is treated as multiple errors. Our schemes enable all nodes to recover from  $n_e$  errors, provided that  $4n_e$  protection paths are shared by all the connections. More generally, if there are  $n_e$  adversarial errors and  $n_f$  failures, we show that a total of  $4n_e + 2n_f$  protection paths are sufficient. We emphasize that the number of protection paths only depends on the number of errors and failures being protected against and is independent of the number of primary path connections.

Section II introduces the network model and our encoding protocol, which is a generalization of [2]. The error model is explained in Section III. In Section IV, we present the decoding algorithm and conditions when a single error happens. Generalizations to multiple errors and combinations of errors and failures are considered in Section V and Section VI.

## II. NETWORK MODEL AND ENCODING PROTOCOL

Suppose that  $2n$  nodes in the network establish  $n$  bidirectional connections with the same capacity. These nodes are partitioned into two disjoint sets  $\mathcal{S}$  and  $\mathcal{T}$  such that each node in  $\mathcal{S}$  connects to one node in  $\mathcal{T}$ . The  $n$  connections are labeled by numbers  $1, \dots, n$  and the nodes participating in the  $i$ th connection are given index  $i$ , i.e.,  $S_i$  and  $T_i$ . Each connection contains one bidirectional primary path  $S_i - T_i$ .  $S_i$  and  $T_i$  send data units they want to transmit onto the primary path. The data unit sent from  $S_i$  to  $T_i$  (from  $T_i$  to  $S_i$ ) on the primary path is denoted by  $d_i$  ( $u_i$ ). The data unit received on the primary path by  $T_i$  ( $S_i$ ) is denoted by  $\hat{d}_i$  ( $\hat{u}_i$ ).

A protection path  $\mathbf{P}$  is a bidirectional path going through all  $2n$  end nodes of  $n$  connections. It has the same capacity as those primary paths and consists of two unidirectional paths  $\mathbf{S}$  and  $\mathbf{T}$  in opposite directions. Each round the data units are transmitted on primary paths and appropriately encoded data units are transmitted on the protection paths.  $M$  protection paths are used and we assume that there are enough resources in the network so that these protection paths can always be found and provisioned. In this paper for the sake of simplicity we assume that all protection paths pass through all the connections and they are denoted by  $\mathbf{P}^{(1)}, \dots, \mathbf{P}^{(M)}$ .

In general, if different primary path connections are protected by a different number of protection paths then our results can be applied with suitable modifications. All operations are over the finite field  $GF(q)$ ,  $q = 2^r$ , where  $r$  is the length of the data unit in bits.

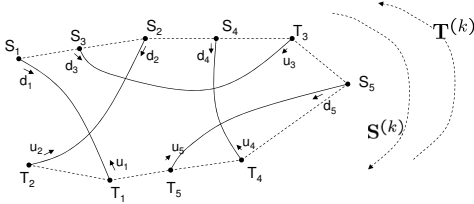


Fig. 1. Five primary path connections  $S_i - T_i$ ,  $i = 1, \dots, 5$  being protected by a single protection path  $\mathbf{P}^{(k)}$  (dashed line) consisting of  $\mathbf{S}^{(k)}$  and  $\mathbf{T}^{(k)}$ .

The encoding operation is executed by each node in  $\mathcal{S}$  and  $\mathcal{T}$ . Each connection  $S_i - T_i$  has  $2M$  encoding coefficients:  $\alpha_i^{(1)}, \dots, \alpha_i^{(M)}, \beta_i^{(1)}, \dots, \beta_i^{(M)}$ , where  $\alpha_i^{(k)}$  and  $\beta_i^{(k)}$  are used for encoding on protection path  $\mathbf{P}^{(k)}$ . Each protection path uses the same protocol but different coefficients in general. We specify the protocol for protection path  $\mathbf{P}^{(k)}$ , which consists of two unidirectional paths  $\mathbf{S}^{(k)}$  and  $\mathbf{T}^{(k)}$ . For this we first define the following notation.

- $\sigma(S_i)/\sigma(T_i)$ : the next node downstream from  $S_i$  (respectively  $T_i$ ) on  $\mathbf{S}^{(k)}$ .  $\sigma^{-1}(S_i)/\sigma^{-1}(T_i)$ : the next node upstream from  $S_i$  (respectively  $T_i$ ) on  $\mathbf{S}^{(k)}$ .
- $\tau(S_i)/\tau(T_i)$ : the next node downstream from  $S_i$  (respectively  $T_i$ ) on  $\mathbf{T}^{(k)}$ .  $\tau^{-1}(S_i)/\tau^{-1}(T_i)$ : the next node upstream from  $S_i$  (respectively  $T_i$ ) on  $\mathbf{T}^{(k)}$ .

For example, in Figure 1,  $\sigma(S_4) = T_3$ ,  $\tau^{-1}(S_5) = T_4$ . Each node is assumed to have a large enough buffer that can store the received linear combinations as long as required. Every packet is assigned a round number and encoding only takes place between packets of the same round. This ensures that the protocol works even without any explicit time synchronization between transmissions. Each node transmits to its downstream node the summation of the data units from its upstream node and a linear combination of the data units it has on each unidirectional protection path. Denote the data unit transmitted on link  $e \in \mathbf{S}^{(k)}$  ( $e \in \mathbf{T}^{(k)}$ ) by  $\mathbf{S}_e$  ( $\mathbf{T}_e$ ). Note that node  $S_i$  knows  $d_i$  and  $\hat{u}_i$ , and  $T_i$  knows  $u_i$  and  $\hat{d}_i$ . The encoding operations can be specified as follows.

$$\begin{aligned} \mathbf{S}_{S_i \rightarrow \sigma(S_i)} &= \mathbf{S}_{\sigma^{-1}(S_i) \rightarrow S_i} + \alpha_i^{(k)} d_i + \beta_i^{(k)} \hat{u}_i, \\ \mathbf{T}_{S_i \rightarrow \tau(S_i)} &= \mathbf{T}_{\tau^{-1}(S_i) \rightarrow S_i} + \alpha_i^{(k)} d_i + \beta_i^{(k)} \hat{u}_i, \\ \mathbf{S}_{T_i \rightarrow \sigma(T_i)} &= \mathbf{S}_{\sigma^{-1}(T_i) \rightarrow T_i} + \alpha_i^{(k)} \hat{d}_i + \beta_i^{(k)} u_i, \text{ and} \\ \mathbf{T}_{T_i \rightarrow \tau(T_i)} &= \mathbf{T}_{\tau^{-1}(T_i) \rightarrow T_i} + \alpha_i^{(k)} \hat{d}_i + \beta_i^{(k)} u_i. \end{aligned}$$

Once a node receives data units over both  $\mathbf{S}^{(k)}$  and  $\mathbf{T}^{(k)}$  it adds these data units. We denote the summation as  $P^{(k)}$ , e.g.,  $T_i$  gets two values  $\mathbf{S}_{\sigma^{-1}(T_i) \rightarrow T_i}$  and  $\mathbf{T}_{\tau^{-1}(T_i) \rightarrow T_i}$  from  $\mathbf{P}^{(k)}$ ,  $P^{(k)1}$  equals to

$$\begin{aligned} \mathbf{S}_{\sigma^{-1}(T_i) \rightarrow T_i} + \mathbf{T}_{\tau^{-1}(T_i) \rightarrow T_i} &= \sum_{l: S_l \in \mathcal{S}} \alpha_l^{(k)} d_l + \sum_{l: T_l \in \mathcal{T} \setminus \{T_i\}} \beta_l^{(k)} u_l \\ &+ \sum_{l: S_l \in \mathcal{S}} \beta_l^{(k)} \hat{u}_l + \sum_{l: T_l \in \mathcal{T} \setminus \{T_i\}} \alpha_l^{(k)} \hat{d}_l \end{aligned} \quad (1)$$

<sup>1</sup> $P^{(k)}$  are different at different end nodes. Here we focus our discussion on node  $T_i$ . To keep the notation simple, we use  $P^{(k)}$  instead of  $P_{T_i}^{(k)}$

In the absence of any errors,  $d_l = \hat{d}_l, u_l = \hat{u}_l$ , most terms cancel out and  $P^{(k)} = \alpha_i^{(k)} d_i + \beta_i^{(k)} \hat{u}_i$ . We refer the reader to [2] for a more detailed description of the encoding protocol and the associated assumptions.

### III. ERROR MODEL

If the adversary changes data units on one (primary or protection) path, an error happens. We assume that if a path is under the control of an adversary, she can arbitrarily change the data units in each direction on that path. If  $d_i \neq \hat{d}_i$  or  $u_i \neq \hat{u}_i$  (or both), we say there is an error on primary path  $S_i - T_i$  with error values  $e_{d_i} = d_i + \hat{d}_i$  and  $e_{u_i} = u_i + \hat{u}_i$ . As for protection path error, although the error is bidirectional, each node will see only one error. In fact, even multiple errors on the same protection path will only have effect as one error at one node. Because each node only use the sum ( $P^{(k)}$ ) of data units from two directions of the protection path to decode. If this data unit is changed due to several error values, they can be modeled as one variable  $e_{p_k}$  at the node. However, different nodes could have different values of  $e_{p_k}$ . If there is a primary path failure on  $S_i - T_i$ , we have  $\hat{d}_i = \hat{u}_i = 0$ . i.e. failures are not adversarial. All nodes know the locations of failures but do not know the locations of errors.

When there are errors in the network, the error terms will not cancel out in (1) and  $T_i$  obtains  $P^{(k)} = \alpha_i^{(k)} d_i + \beta_i^{(k)} (u_i + e_{u_i}) + \sum_{l \in I_{\setminus i}} (\alpha_l^{(k)} e_{d_l} + \beta_l^{(k)} e_{u_l}) + e_{p_k}$  on protection path  $\mathbf{P}^{(k)}$ , where  $I_{\setminus i} = \{1, \dots, n\} \setminus \{i\}$ , the index set excluding  $i$ , and  $e_{p_k}$  is the error on protection path  $\mathbf{P}^{(k)}$  seen by  $T_i$ . Note that since  $T_i$  knows  $u_i$ , we can subtract it from this equation. Together with the data unit  $P_m$  from the primary path,  $T_i$  has the following data units.

$$\begin{aligned} P_m &= d_i + e_{d_i} \quad (2) \\ P^{(k)'} &= \alpha_i^{(k)} d_i + \beta_i^{(k)} e_{u_i} \\ &+ \sum_{l \in I_{\setminus i}} (\alpha_l^{(k)} e_{d_l} + \beta_l^{(k)} e_{u_l}) + e_{p_k} \quad (3) \end{aligned}$$

where  $k = 1, \dots, M$ .

We can multiply (2) by  $\alpha_i^{(k)}$  and add to  $k^{\text{th}}$  equation in (3) to obtain the system of equations.

$$[H|I_{M \times M}]E = P_{syn} \quad (4)$$

where the length- $M$  vector  $P_{syn} = [\alpha_i^{(1)} P_m + P^{(1)'}, \alpha_i^{(2)} P_m + P^{(2)'}, \dots, \alpha_i^{(M)} P_m + P^{(M)'}]^T$ ,  $H$  is a  $M \times 2n$  coefficient matrix

$$H = \begin{bmatrix} \alpha_1^{(1)} & \beta_1^{(1)} & \dots & \alpha_n^{(1)} & \beta_n^{(1)} \\ \alpha_1^{(2)} & \beta_1^{(2)} & \dots & \alpha_n^{(2)} & \beta_n^{(2)} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \alpha_1^{(M)} & \beta_1^{(M)} & \dots & \alpha_n^{(M)} & \beta_n^{(M)} \end{bmatrix},$$

$I_{M \times M}$  is an identity matrix, and

$$E \triangleq [e_{d_1}, e_{u_1}, \dots, e_{d_n}, e_{u_n}, e_{p_1}, \dots, e_{p_M}]^T.$$

Analogous to classical coding theory, we may call  $P_{syn}$  the syndrome available at the decoder. Denote the coefficient matrix of (4) as  $H_{ext} = [H|I_{M \times M}] = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_{2n} | \mathbf{v}_1^p, \dots, \mathbf{v}_M^p]$ . Note that for a connection  $S_i -$

$T_i$ ,  $\mathbf{v}_{2i-1}$ ,  $\mathbf{v}_{2i}$  are the columns consisting of encoding coefficients  $\alpha_i$ 's and  $\beta_i$ 's for it.  $T_i$  knows  $H_{ext}$  and  $P_{syn}$  and shall find error locations and values from (4). Once error locations and values are known, simply use (2)  $d_i = P_m + e_{d_i}$  to obtain  $d_i$ . Node  $S_i$  gets very similar equations to those at  $T_i$ . Thus we will focus our discussion on  $T_i$ . Each end node uses the same decoding algorithm and works individually without cooperations.

#### IV. RECOVERY FROM SINGLE ERROR

In this section, we focus on the case when there is only one error in the network. We first present the decoding algorithm and then prove that it yields the correct solution under appropriate conditions.

##### A. Decoding algorithm

At  $T_i$ , the node performs the following.

- 1) Attempts to solve the following system of equations

$$[\mathbf{v}_{2i-1} \mathbf{v}_{2i}] \begin{bmatrix} e_{d_i} \\ e_{u_i} \end{bmatrix} = P_{syn} \quad (5)$$

- 2) If (5) has a solution  $(e_{d_i}, e_{u_i})$ , compute  $d_i = P_m + e_{d_i}$ , otherwise,  $d_i = P_m$

This algorithm also works when the error happens on one of the protection paths.

##### B. Condition for one primary path error correction

In this subsection, we consider primary path error only. An *error pattern* is two columns in  $H$  corresponding to the erroneous primary path. If error happens on  $S_i - T_i$ , the error pattern is  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}\}$ . An *error value vector* corresponding to an error pattern is obtained by letting the error values corresponding to other  $n - 1$  primary paths to be zero. The error value vector corresponding to error pattern  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}\}$  is  $E_i = [0, \dots, e_{d_i}, e_{u_i}, \dots, 0]^T$ . Assume that  $e_{d_i}$ 's and  $e_{u_i}$ 's are not all zero. The case when all of them are zero is trivial because it implies no error happens.

*Theorem 1:* Suppose there is at most one error on a primary path. The decoding algorithm outputs the correct data unit at every node if and only if the vectors in the set  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_{2j-1}, \mathbf{v}_{2j}\}$  for all  $i, j = 1, \dots, n, i \neq j$  are linearly independent.

*proof:* First assume that the vectors in the sets  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_{2j-1}, \mathbf{v}_{2j}\}$  are linearly independent. Let  $E_a$  and  $E_b$  be error value vectors corresponding to errors happening on different primary paths  $S_a - T_a$  and  $S_b - T_b$  respectively. Suppose there exist  $E_a$  and  $E_b$  such that  $HE_a = HE_b$ , i.e.,  $H(E_a + E_b) = 0$ . Note that the vector  $(E_a + E_b)$  has at most four error values  $[e_{d_a}, e_{u_a}, e_{d_b}, e_{u_b}]$  which are not all zero and such that  $[\mathbf{v}_{2a-1}, \mathbf{v}_{2a}, \mathbf{v}_{2b-1}, \mathbf{v}_{2b}] [e_{d_a}, e_{u_a}, e_{d_b}, e_{u_b}]^T = 0$ . This implies  $\{\mathbf{v}_{2a-1}, \mathbf{v}_{2a}, \mathbf{v}_{2b-1}, \mathbf{v}_{2b}\}$  are linearly dependent, which is a contradiction. Therefore, under our condition that  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_{2j-1}, \mathbf{v}_{2j}\}$  for all  $i, j = 1, \dots, n, i \neq j$  are linearly independent, there does not exist  $E_a, E_b$  such that  $HE_a = HE_b$ . This means that if we try to solve the system of linear equations according to every possible error value vectors  $E_1, \dots, E_n$ , it either has no solution or its solution is the actual error in the network. Since one node  $T_i$  is only

interested in  $d_i$ , in our decoding algorithm, it tries to solve the equations (5) according to the error value vector  $E_i$ . If it has a solution, the error happens on  $S_i - T_i$ . Under our condition, the matrix  $[\mathbf{v}_{2i-1}, \mathbf{v}_{2i}]$  has rank two, so equations (5) have unique solution for  $e_{d_i}$ .  $d_i = P_m + e_{d_i}$  gives decoded  $d_i$ . If (5) does not have solution, the error is not on  $S_i - T_i$ .  $T_i$  simply picks up  $d_i = P_m$  from the primary path  $S_i - T_i$ .

Conversely, suppose that a vector set  $\{\mathbf{v}_{2i_1-1}, \mathbf{v}_{2i_1}, \mathbf{v}_{2j_1-1}, \mathbf{v}_{2j_1}\}$  is linearly dependent. There exist  $E_{i_1}$  and  $E_{j_1}$  such that  $HE_{i_1} = HE_{j_1}$ . At node  $T_{i_1}$ , both equations  $HE_{i_1} = P_{syn}$  and  $HE_{j_1} = P_{syn}$  have solution. Suppose the error in fact happens on  $S_{j_1} - T_{j_1}$ , the decoder at  $T_{i_1}$  can also find a solution to  $HE_{i_1} = P_{syn}$  and use the solution to compute  $d_i$ . This leads to decoding error. ■

If there is no error in the network,  $P_{syn} = 0$  and solving (5) gives  $e_{d_i} = e_{u_i} = 0$ . In order to make  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_{2j-1}, \mathbf{v}_{2j}\}$  independent, we need the length of vector  $\mathbf{v}$ 's to be at least four, i.e.,  $M \geq 4$ . In fact, we shall see that several coefficient assignment strategies ensure that four protection paths are enough. The condition in Theorem 1 can be stated as all  $M \times M$  ( $4 \times 4$ ) matrices of the form

$$[\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_{2j-1}, \mathbf{v}_{2j}], i, j = 1, \dots, n, i < j \quad (6)$$

as having full rank.

##### C. Coefficient assignment methods

We shall introduce several ways to assign encoding coefficients, i.e., to construct  $H$  matrix so that (6) has full rank. The first method is to choose  $n$  distinct elements  $\gamma_1, \dots, \gamma_n$  from  $GF(q)$ . For all  $i = 1, \dots, n$ ,  $\alpha_i^{(1)} = 1$ ,  $\alpha_i^{(2)} = \gamma_i$ ,  $\beta_i^{(3)} = 1$ ,  $\beta_i^{(4)} = \gamma_i$  and all other coefficients are zero. After Gaussian elimination we can see that the matrix (6) has full rank as long as  $\gamma$ 's are distinct. The minimum field size needed is  $q \geq n$ . The second way is to choose  $2n$  distinct elements from  $GF(q)$ :  $\gamma_{\alpha_1}, \gamma_{\beta_1}, \dots, \gamma_{\alpha_n}, \gamma_{\beta_n}$  and let encoding coefficients to be  $\alpha_i^{(k)} = \gamma_{\alpha_i}^{k-1}$ ,  $\beta_i^{(k)} = \gamma_{\beta_i}^{k-1}$ . (6) becomes a Vandermonde matrix and as long as  $\gamma$ 's are distinct, it has full rank. Random choice from a large field also works due to the following claim.

*Claim 1:* When all coefficients are randomly, independently and uniformly chosen from  $GF(q)$ , for given  $i$  and  $j$ , the probability that  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_{2j-1}, \mathbf{v}_{2j}\}$  are linearly independent is  $p_1 = (1 - 1/q^3)(1 - 1/q^2)(1 - 1/q)$ .

In (6) we require  $\binom{n}{2}$  such matrices have full rank. By union bound, the probability for successful decoding at all nodes is at least  $1 - (1 - p_1) \binom{n}{2}$ , which is close to 1 when  $q$  is large.

##### D. Taking protection path error into account

In this subsection, we take protection path error into account. The error (assume one error in this section) can happen either on one primary path or one protection path. Besides  $n$  error value vectors  $E_1, \dots, E_n$ , we have  $M$  more error value vectors for the protection path error:  $[0|e_{p_1}, 0, \dots, 0]^T, \dots, [0|0, 0, \dots, e_{p_M}]^T$ . Denote them by  $E_{p_1}, \dots, E_{p_M}$ . There are  $n + M$  error value vectors in total. Using a similar idea to Theorem 1, we have the following:

*Theorem 2:* If there is one error on one primary path or protection path, the decoding algorithm works for every node if and only if vectors in the sets  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_{2j-1}, \mathbf{v}_{2j}\}, i, j =$

$1, \dots, n, i \neq j$  and  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_l^p\}, i = 1, \dots, n, l = 1, \dots, M$  are linearly independent. Note that  $\mathbf{v}_l^p$  is the  $l$ th column in  $I_{M \times M}$  in (4).

In fact,  $M = 4$  and the three coefficient assignment methods we described in the previous subsection work in this case. The first two schemes ensure vectors in each set  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_l^p\}$  to be independent. When the coefficients are randomly chosen from  $GF(q)$ , for given  $i$  and  $l$ , the probability that  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}, \mathbf{v}_l^p\}$  are linearly independent is  $p_2 = (1 - 1/q^3)(1 - 1/q^2)$ . The probability of successful decoding at all nodes is at least  $1 - (1 - p_1) \binom{n}{2} - (1 - p_2)nM$ , which approaches 1 when  $q$  is large.

#### E. Remark

We can compare our results with classical results in coding theory. In classical coding theory, in the presence of two adversarial errors, we need a code with minimum distance at least five for correct decoding. This means that to transmit one symbol of information, we need to transmit a codeword with at least five symbols. In our problem each connection has a total of five paths (one primary and four protection). A single error on a bidirectional primary path induces two errors, one in each direction. Therefore in an approximate sense we are using almost the optimal number of protection paths. It is important to note that the protection paths are shared so the cost of protection per primary path connection is small.

### V. MULTIPLE ERRORS

Our analysis can be generalized to multiple errors on primary and protection paths. Assume that  $n_c$  errors happen on primary paths and  $n_p = n_e - n_c$  errors happen on protection paths. As described in Section III, a given primary path error corresponds to two specific columns in  $H_{ext}$  while a protection path error corresponds to one specific column in  $H_{ext}$ .

*Definition 1:* A subset of columns of  $H_{ext}$  denoted as  $A(m_1, m_2)$  is an error pattern with  $m_1$  errors on primary paths  $\{c_1, \dots, c_{m_1}\} \subseteq \{1, \dots, n\}$  and  $m_2$  errors on protection paths  $\{p_1, \dots, p_{m_2}\} \subseteq \{1, \dots, M\}$  if it has the following form:  $A(m_1, m_2) = A_c(m_1) \cup A_p(m_2)$ , where  $A_c(m_1) = \{\mathbf{v}_{2c_1-1}, \mathbf{v}_{2c_1}, \dots, \mathbf{v}_{2c_{m_1}-1}, \mathbf{v}_{2c_{m_1}}\}$ ,  $c_i \in \{1, \dots, n\}$  and  $A_p(m_2) = \{\mathbf{v}_{p_1}^p, \dots, \mathbf{v}_{p_{m_2}}^p\}$ ,  $p_i \in \{1, \dots, M\}$ . Thus, the set of columns in  $H_{ext}$  can be expressed as  $A(n, M)$ .

We let  $\mathbf{A}(m_1, m_2)$  denote the family of error patterns with  $m_1$  primary path errors and  $m_2$  protection path errors (for brevity, henceforth we refer to such errors as  $(m_1, m_2)$  type errors). i.e. if an error pattern  $A(x, y) \in \mathbf{A}(m_1, m_2)$ , then  $x = m_1$  and  $y = m_2$ . Note that  $|\mathbf{A}(m_1, m_2)| = \binom{n}{m_1} \binom{M}{m_2}$ .

*Definition 2:* Consider a subset of  $\mathbf{A}(m_1, m_2)$  denoted  $\mathbf{A}(m_1, m_2)_i$ , with the following property.  $A(m_1, m_2) \in \mathbf{A}(m_1, m_2)_i$  if and only if  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}\} \subseteq A(m_1, m_2)$ . Then  $\mathbf{A}(m_1, m_2)_i$  is the family of  $(m_1, m_2)$  type error patterns such that each error pattern includes an error on  $S_i - T_i$ .

Denote the family of error pattern including an error on  $S_i - T_i$  with  $n_e$  errors in total as:  $\mathbf{A}_i = \cup_{n_c=1}^{n_e} \mathbf{A}(n_c, n_e - n_c)_i$ . Our definition of an error pattern has only specified the location of the error but not the actual values. To facilitate the later development we now define an error value vector corresponding to an error pattern. An error value vector  $E$  has the

following form:  $[e_{d_1}, e_{u_1}, \dots, e_{d_n}, e_{u_n}, e_{p_1}, \dots, e_{p_M}]^T$ . Each entry of the vector corresponds to one column in  $H_{ext}$ . An error value vector  $E$  corresponds to an error pattern  $A(m_1, m_2)$  if in  $E$ , the entries corresponding to  $A(n, M) \setminus A(m_1, m_2)$  are zero, while the other entries may be non-zero. We are now ready to present the decoding algorithm in the presence of multiple errors.

#### A. Multiple errors decoding algorithm

Consider decoding at  $T_i$ ,

- 1) Try to solve the system of linear equations obtained from (4) according to each error pattern in  $\mathbf{A}_i$ . The indeterminates are given by the error value vector corresponding to the error pattern.
- 2) Suppose that the decoder finds a solution to one of these system of equations. Compute  $d_i = P_m + e_{d_i}$ , where  $e_{d_i}$  is recovered as part of the solution. If none of these systems of equations has a solution, set  $d_i = P_m$ .

#### B. Condition for error correction

*Theorem 3:* Suppose that there are at most  $n_e$  errors in the network (both primary path error and protection path error are possible). The result of the decoding algorithm is correct at every node if and only if the column vectors in  $A(m_1, m_2)$  are linearly independent for all  $A(m_1, m_2) \in \cup_{n_c, n'_c \in \{0, \dots, n_e\}} \mathbf{A}(n_c + n'_c, 2n_e - (n_c + n'_c))$ .

*proof:* Suppose  $E_1$  and  $E_2$  denote two error value vectors corresponding to error patterns in  $\mathbf{A}(n_c, n_e - n_c)$  and  $\mathbf{A}(n'_c, n_e - n'_c)$  respectively and  $E_1 \neq E_2$ .  $n_c$  and  $n'_c$  are the number of errors on primary paths in  $E_1$  and  $E_2$  respectively.  $E_{sum} = E_1 + E_2 \neq 0$  has at most  $n_c + n'_c$  errors on primary paths and  $n_p + n'_p = 2n_e - (n_c + n'_c)$  errors on protection path. These errors correspond to a member (which is a set of column vectors)  $A(n_c + n'_c, 2n_e - (n_c + n'_c)) \in \mathbf{A}(n_c + n'_c, 2n_e - (n_c + n'_c))$ . By arguments similar to those in Theorem 1, the linear independence condition in this theorem implies that there do not exist  $E_1$  and  $E_2$  such that  $HE_1 = HE_2$ . This means that if a decoder tries to solve every system of linear equations according to every possible error patterns with  $n_e$  errors, it either gets no solution, or gets the same solution for multiple solvable systems of linear equations. A decoder at  $T_i$  is only interested in error patterns in  $\mathbf{A}_i$ . If in step 1 it finds a solution  $E$  for one system of equation,  $e_{d_i}$  in  $E$  is the actual error value for  $d_i$  and  $d_i = \hat{d}_i + e_{d_i}$ , otherwise, no error happens on  $S_i - T_i$ .

Conversely, if there exist some  $n_c, n'_c$  such that some member in  $\mathbf{A}(n_c + n'_c, 2n_e - (n_c + n'_c))$  is linearly dependent, there exist  $E'_1$  and  $E'_2$  such that  $HE'_1 = HE'_2$  and  $E'_1 \neq E'_2$ . Suppose  $e_{d_i}$  is different in these two error value vectors. At node  $T_{i_1}$ , the decoder has no way to distinguish which one is the actual error value vector and the decoding fails. ■

The above condition is equivalent to the fact that all vector sets  $A(m_1, m_2) \in \cup_{m \in \{0, \dots, 2n_e\}} \mathbf{A}(m, 2n_e - m)$  are linearly independent.  $|A(m, 2n_e - m)| = 2n_e + m$  and its maximum is  $4n_e$ . Thus, the length of the vectors should be at least  $4n_e$ . In fact,  $M = 4n_e$  is sufficient under random chosen coefficients. Suppose coefficients are randomly and uniformly

chosen from  $GF(q)$ . For a fixed  $m$ , first compute  $p_1(m)$ , the probability that  $A(m, 2n_e - m) = A_c(m) \cup A_p(2n_e - m)$  is linearly independent.  $A_p(2n_e - m)$  contains  $2n_e - m$  basis vectors of  $M$ -dimensional vector space. By similar argument to Claim 1,  $p_1(m) = \prod_{i=0}^{2m-1} (1 - q^{2n_e - m + i} / q^M)$ . Considering all members in  $\mathbf{A}(m, 2n_e - m)$  and all values of  $m$ , by union bound, the probability for successful decoding is at least  $1 - \sum_{m=0}^{2n_e} (1 - p_1(m)) \binom{n}{m} \binom{M}{2n_e - m}$ , which approaches 1 when  $q$  is large.

### C. Reed-Solomon like efficient decoding for primary path error only case

A Reed-Solomon code has a parity check matrix with a very specific form. It leads to several efficient algorithms for decoding, e.g., Berlekamp-Massey algorithm (BMA) [7, Chapter 7]. In our context we can choose  $H$  so that  $H_{ij} = (\alpha^i)^{j-1}$ , where  $\alpha$  is the primitive element over  $GF(q)$ ,  $q > 2n$ . Denote it by  $H_{RS}$ . This is a parity check matrix of a  $(2n, 2n - M)$  RS code and any  $M$  columns in  $H_{RS}$  are linearly independent. If the errors only happen on primary paths, the condition in Theorem 3 becomes that each member of  $\mathbf{A}(2n_e, 0)$  is linearly independent.  $H_{RS}$  satisfies this condition. Thus, (4) becomes  $H_{RS}[e_{d_1}, e_{u_1}, \dots, e_{d_n}, e_{u_n}]^T = P_{syn}$ , in which  $H_{RS}$  and  $P_{syn}$  are known by every node. The problem becomes to find an error pattern with at most  $n_e$  errors and the corresponding error value vector. Note that in fact there are  $2n_e$  error values to be decided. This problem can be viewed as RS hard decision decoding problem while the number of errors is bounded by  $M/2 = 2n_e$ .  $P_{syn}$  can be viewed as the *syndrome* of the received message. The efficient BMA can then be applied.

## VI. A COMBINATION OF ERRORS AND FAILURES

We now consider a combination of errors and failures on primary and protection paths. Assume that there are a total of  $n_f$  failures in the network, such that  $n_{f_c}$  failures are on primary paths and  $n_{f_p} = n_f - n_{f_c}$  failures are on protection paths. If a protection path has a failure it is basically useless and we remove the equation corresponding to it in error model (4). Thus, we shall mainly work with primary path failures and error model (4) will have  $M' = M - n_{f_p}$  equations. In our error model, when a primary path error happens,  $e_{d_i} = d_i + \hat{d}_i$  ( $e_{u_i} = u_i + \hat{u}_i$ ) and when a primary path failure happens,  $\hat{d}_i = 0$  ( $\hat{u}_i = 0$ ). Therefore, we can treat a primary path failure as a primary path error with error value  $e_{d_i} = d_i$  ( $e_{u_i} = u_i$ ). The decoding algorithm and condition in this case are very similar to multiple error case. An important difference is that the decoder knows the location of  $n_f$  failures.

*Definition 3:* A subset of columns of  $H$  denoted by  $F(n_{f_c})$  is said to be a failure pattern with  $n_{f_c}$  primary path failures if it has the following form:  $F(n_{f_c}) = \{\mathbf{v}_{2f_1-1}, \mathbf{v}_{2f_1}, \dots, \mathbf{v}_{2f_{n_{f_c}}-1}, \mathbf{v}_{2f_{n_{f_c}}}\}, f_i \in \{1, \dots, n_f\}$ .

*Definition 4:* An error/failure pattern with  $m_1$  primary path errors,  $m_2$  protection path errors and failure pattern  $F(n_{f_c})$  is defined as  $A^F(m_1, m_2, F(n_{f_c})) = A(m_1, m_2) \setminus F(n_{f_c}) \cup F(n_{f_c})$ , where  $A(m_1, m_2) \setminus F(n_{f_c}) \in \mathbf{A}(m_1, m_2)$  and is such that  $A(m_1, m_2) \setminus F(n_{f_c}) \cap F(n_{f_c}) = \emptyset$ .

We let  $\mathbf{A}^F(m_1, m_2, F(n_{f_c}))$  denote the family of error/failure pattern with  $m_1$  primary path errors,  $m_2$  protection

path errors ( $(m_1, m_2)$  type errors) and a failure pattern  $F(n_{f_c})$ . Note that  $|\mathbf{A}^F(m_1, m_2, F(n_{f_c}))| = \binom{n - n_{f_c}}{m_1} \binom{M'}{m_2}$ .

*Definition 5:* Consider a subset of  $\mathbf{A}^F(m_1, m_2, F(n_{f_c}))$  denoted as  $\mathbf{A}^F(m_1, m_2, F(n_{f_c}))_i$ , with the following property.  $A^F(m_1, m_2, F(n_{f_c})) \in \mathbf{A}^F(m_1, m_2, F(n_{f_c}))_i$  if and only if  $\{\mathbf{v}_{2i-1}, \mathbf{v}_{2i}\} \subseteq A^F(m_1, m_2, F(n_{f_c}))$ . This is the family of error/failure patterns such that each pattern includes an error or failure on  $S_i - T_i$ .

We say that an error/failure value vector  $E$  corresponds to an error/failure pattern  $A^F(m_1, m_2, F(n_{f_c}))$  if in  $E$ , the entries corresponding to  $A(n, M) \setminus A^F(m_1, m_2, F(n_{f_c}))$  are zero, while the other entries may be non-zero.

The decoding algorithm is similar to multiple error case. Since  $T_i$  knows the primary path failure pattern  $F(n_{f_c})$ , it tries to solve equations of (4) form according to all possible error/failure patterns in  $\cup_{n_c=1}^{n_e} \mathbf{A}^F(n_c, n_e - n_c, F(n_{f_c}))_i$ . The rest is the same as multiple error case and we have a theorem similar to Theorem 3.

*Theorem 4:* Suppose there are at most  $n_e$  errors and  $n_{f_c}$  primary path failures in the network (both primary path error and protection path error are possible). The decoding algorithm works at every node if and only if the column vectors in  $A(m_1, m_2)$  are linearly independent for all  $A(m_1, m_2) \in \cup_{n_c, n'_c \in \{0, \dots, n_e\}} \mathbf{A}(n_{f_c} + n_c + n'_c, 2n_e - (n_c + n'_c))$ .

*proof:* The condition implies that for all  $n_c, n'_c \in \{0, \dots, n_e\}$  and all possible failure pattern  $F(n_{f_c})$ , each member in  $\mathbf{A}^F(n_c + n'_c, 2n_e - (n_c + n'_c), F(n_{f_c}))$  is linearly independent. The rest of the proof is similar to Theorem 3. ■

The maximum number of vectors contained in each such vector group is  $4n_e + 2n_{f_c}$ . Thus, we need at least  $M' = 4n_e + 2n_{f_c}$  equations in (4) which implies in turn that  $M = 4n_e + 2n_{f_c} + n_{f_p}$ . Since we don't know  $n_{f_c}, n_{f_p}$  a priori, we need at least  $M = 4n_e + 2n_f$ . In fact,  $4n_e + 2n_f$  is enough under random choice of coefficients from a large enough field. We note that as explained in the previous section if we restrict the errors/failures to be only on the primary paths, then we can choose the coefficient matrix  $H$  to be the parity-check matrix of a  $(2n, 2n - 4n_e - 2n_f)$  RS code. The decoding problem can be viewed as the RS hard decision decoding problem while the number of error values is bounded by  $2n_e$  and the number of failure values is bounded by  $2n_f$ . It can be done by a modified BMA [7] that works for errors and erasures.

## REFERENCES

- [1] A.E.Kamal, "1+n protection against multiple faults in mesh networks," in *proc. of the IEEE Intl. Conf. on Communications (ICC)*, 2007.
- [2] A. E. Kamal and A. Ramamoorthy, "Overlay protection against link failures using network coding," in *42nd Conf. on Info. Sci. and Sys. (CISS)*, 2008.
- [3] Z. Zhang, "Linear network error correction codes in packet networks," *IEEE Trans. on Info. Th.*, vol. 54, no. 1, pp. 209–218, Jan. 2008.
- [4] S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros, "Resilient network coding in the presence of Byzantine adversaries," in *IEEE INFOCOM*, 2007, pp. 616–624.
- [5] R. Koetter and F. Kschischang, "Coding for errors and erasures in random network coding," in *Proc. IEEE Int. Symp. Inf. Theory*, Jun. 2007.
- [6] S. Yang, R. W. Yeung, and C. K. Ngai, "Refined coding bounds and code constructions for coherent network error correction," preprint.
- [7] S. Lin and D. J. Costello, *Error control coding: fundamentals and applications*. Prentice Hall, 2004.