IEEE Information Theory Workshop, Bergen, Norway, 2007.

Robust Network Coding in the Presence of Link Failure Using a Multicast Rate-Diversity Trade off

Hossein Bahramgiri, Mohammad Jabbari, Farshad Lahouti
Wireless Multimedia Communications Laboratory
School of ECE, College of Engineering
University of Tehran
[h.bahramgiri, m.jabari]@ece.ut.ac.ir, lahouti@ut.ac.ir

Abstract- **We introduce a centralized joint network-channel coding scheme for maximized throughput and increased robustness against link failure in a communication network. We introduce *RNC1(h,k)* to multicast *k* independent data in a directed and acyclic network with capacity *h*, $k \le h$. The redundancy through path diversity increases the resistance of the network against link failure. An improved version of *RNC1*, *RNC2(h,k)*, is introduced to guarantee the rate *k* for *h-k* flow failures. The proposed schemes exploit the network structure for code design, which results in a manageable design complexity.**

Index terms- **Network coding, joint network-channel coding, link failure, multicast**

I. INTRODUCTION

Network coding is a coding scheme in the network layer and improves upon routing algorithms in communication networks. Ahlswede et al. in [1] demonstrated that the multicast capacity, deducted from the max-flow min-cut theorem, can be achieved through network coding. Next in [2], Li et al. showed that capacity is achieved by linear network coding even with finite alphabet size. Koetter and Medard in [3] introduced an algebraic approach to network coding and studied a variety of information flows. Next in [4], Ho et al. introduced a simple, distributed and random algorithm to design linear network codes. Jaggi et al. in [5] introduced polynomial time centralized algorithms that guarantee the true solution. They presented two algorithms: deterministic linear information flow (*DLIF*), a completely deterministic scheme, and random linear information flow (*RLIF*), which is random in the middle stages. Fraguoli et al. in [6], develop a distributed and deterministic method for network code design by translating the problem as a graph coloring problem.

Considering errors in the network channels, the above algorithms may fail or degrade. One approach is to combat the error in other layers, specifically physical and data link layers. A cross layer approach is to consider a joint network-channel coding scheme. Chou et al. in [7], introduced a practical distributed network coding approach based on the random network coding algorithm of [4]. The approach shows resistance against errors by transmitting some extra random combinations of data. In [8,9], Cai and Yeung derive network generalization of Hamming, Singleton and Gilbert-Varshamov bounds. In [10], Matsumoto suggested a centralized deterministic algorithm for construction of linear network error-correcting codes that attain the Singleton bound. Koetter and Medard in [3] defined link failure pattern as an error pattern, in which a subset of links fail to transmit data. They demonstrated that in a network with multicast rate of *h*, there exists a receiver-based coding scheme, which provides rate $k \le h$ for all link failure patterns that do not reduce the capacity below *k*. In [5] a method to find such a solution is introduced, where it is required that prior to the network code design, all the corresponding link failure patterns are determined. In this scheme, the complexity and the field size grow linearly with the number of failure patterns, which is in general a large number in practical situations. Ho et al in [11], showed the benefits of random network coding towards robustness in the presence of network changes and link failures. In [12], the two types of receiver-based and network-wide recovery schemes are analyzed and results for quantifying network management are presented. They show that there is not a single coding solution for all recoverable single-link failures and at least the receivers must react to the failure pattern. In [13], El Rouayheb et al. investigate network codes that enable instantaneous recovery from single edge failure for unicast connections

In this work, we introduce robust network coding *RNC*(*h*,*k*), as a centralized joint network-channel coding scheme, which increases the resistance of network against link failure. The idea is to trade-off multicast rate of *h*, creating a diversity of size *h-k* to increase the robustness. The design is based on the network structure rather than specific failure patterns, which leads to a reduction of the design complexity.

Two robust network coding algorithms, referred to as *RNC1* and *RNC2*, are proposed. The former is a simple and polynomial-time algorithm, which facilitates increased robustness in the presence of link failures, independent of the failure patterns. The latter, *RNC2*(*h*,*k*), guarantees the rate *k* for *h-k* flow failures of each of the receivers. In this scheme, failure patterns are considered as they affect the flows, and the code is checked only for flow failures, resulting in a manageable complexity.

The rest of this paper is organized as follows: In section II, the problem is defined in detail. In section III, we introduce robust network coding in the presence of link failures in a multicast session. The field size and the complexity of the algorithm are studied. In section IV, an improvement to the proposed scheme is introduced, which guarantees the rate *k* for *h*-*k* flow failures. Finally, the results are presented in section V.

II. Problem definition

A network can be represented by a graph *G*=(*V*,*E*), which is assumed to be directed and acyclic. *V* represents the set of nodes and *E* is the set of links or edges. We assume that each edge has unit capacity. If an edge between two nodes has a positive integer capacity of *c*, we consider *c* parallel edges of unit capacity between two nodes. Thus, an edge $e \in E$ can be represented by $e = (u, v, i)$, where $u = tail(e), v = head(e)$ and *i* indicates the edge number. Each node $v \in V$, has some input edges represented by the set $\Gamma_I(v)$. Consider a multicast session in the network with a transmitter $s \in V$ and a set of receivers $T \subset V$. We assume that the transmission in the network is synchronous.

The source generates a set of *k* independent data symbols $X = \{x_1, ..., x_k\}$, which must be transmitted to all receivers. The local encoding vector of the edge *e*, denoted by $\mathbf{m}_e(.)$, represents the combination coefficients of the input edges of the *tail*(*e*). The symbol on the edge *e* is constructed from the global encoding vector of *e*, **b**(*e*), which is a vector containing the coefficients of data units in the set *X* for the edge *e*:

$$\mathbf{b}(e) = \sum_{e' \in \Gamma_I(tail(e))} \mathbf{m}_e(e')\mathbf{b}(e') . \qquad (1)$$

The network encounters link failure, a non-ergodic error model, which may damage the data transmission process. The error position and duration is not known a priori. With a link failure, the link becomes completely unavailable. The purpose is to combat this type of error.

III. Robust Network Coding

The network capacity is given by the max-flow min-cut theorem. According to the fundamental result of [1],

$$c_{multicast} = \underset{t \in T}{Min}\{Maxflow(t)\} , \qquad (2)$$

and applying network coding, $h = c_{multicast}$ units of data can be transmitted to all the receivers in a multicast session. Any link failure may reduce the capacity of the network in some durations of time. In these durations, the algorithms for transmitting rate *h* collapses and as the coding techniques combine the data symbols, it may lead to missing all the data. Here, we introduce an algorithm designed based on the network structure, which increases the overall resistance of the network against link failure. We refer to this algorithm as Robust Network Coding 1 (*RNC1*).

In *RNC1*(*h*,*k*), the purpose is to transmit *k* units of independent data in a network with error-free capacity of *h*. This scheme provides a diversity (redundancy) of size *h*-*k*, which determines the robustness of the network code against link failure. To exploit this redundancy effectively, we propose the following algorithm. The algorithm is based on a centralized approach, and is inspired by the work of Jaggi et al. [5]. However, the ideas and techniques presented for robust network code design could also be cast into other design algorithms.

The design of *RNC1*(*h*,*k*) involves two steps. The first step is to find *h* flows from the source to each of the receivers. For every receiver *t*, we find the set $\Phi^t = \{f_1^t, ..., f_h^t\}$, containing *h* edge-disjoint flows from the source to the receiver *t*. Note that the flows of any two receivers may have common edges. An edge that is present in at least one flow is referred to as an *active edge*.

In the second step, fixing the flows, *RNC1* determines the transmission strategy in the network. There are *h* edge-disjoint flows from the source to each of the receivers. The transmitter sends *h* symbols to each of the receivers as combinations of the *k* independent symbols in X. Receiving any *k*-element subset of *h* symbols is enough to extract all data.

As in [5], the set C_t is defined for receiver *t*, consisting of *h* edges transmitting *h* symbols to this receiver. The set of the global encoding vectors of these edges, corresponding to the transmitting symbols, is denoted by B_t. The set B_t must have the property that each of its *k*-element subsets forms an independent set of vectors. We refer to this property as *k-independence*. To initialize the algorithm, we add a new node s' to the network and connect it to *s* with *h* directed edges from s' to *s*. Therefore, we have the new graph $G' = (V', E')$ with $V' = V \cup \{s'\}$, and $E' = E \cup \{(s', s, 1), ..., (s', s, h)\}$. The set C_t is initialized by the set of added edges, $\{(s', s, 1), ..., (s', s, h)\}$. The algorithm considers the active edges, one by one. The edge *e* is considered, only after all the edges in $\Gamma_I(tail(e))$ have been processed. The graph is acyclic, so all the links are considered by this strategy. The edge *e* replaces $\Gamma_{\leftarrow}^t(e)$ in C_t for each receiver $t \in Q(e)$, where *Q*(*e*) denotes the set of all receivers, for which the edge *e* is in one of their flows. After replacement, *k*-independence is verified for each updated B_t. The algorithm concludes when the edges ending at the receivers are processed.

To process edge *e*, $\mathbf{m}_e$ is selected randomly and **b**(*e*) is computed using equation 1. Next, the resulting B_t is verified for *k*-independence for each $t \in Q(e)$.

The following theorem determines a bound for the probability of success in designing $\mathbf{m}_e$ as a function of the field size.

Theorem 1: In *RNC1*(*h*,*k*), if the coefficients of $\mathbf{m}_e$ are selected randomly from the finite field *F*, the probability of *k*-independence of the sets C_t, $t \in Q(e)$, is given by

$$p \geq 1 - \binom{h-1}{k-1}\frac{|T|}{|F|} , \qquad (3)$$

where $\binom{a}{b} = \frac{a!}{b!(a-b)!}$.

Proof- The flows of a receiver are disjoint, thus $|Q(e)| \leq |T|$. Therefore, the coefficient candidates for $\mathbf{m}_e$ are at most checked for $|T|$ receivers. Consider the receiver $t \in Q(e)$, and C_t and B_t that are to be updated for the edge *e*. In C_t, $\Gamma_{\leftarrow}^t(e)$ is replaced by *e*. The vector m_e is selected randomly. Using equation 1, we examine a candidate of **b**(*e*) for *k*-independence for the updated B_t. Specifically, the vector **b**(*e*) with every *k*-1 element subset of the global encoding vectors of *h*-1 other edges in C_t must construct an independent set. The probability that

$\mathbf{b}(e)$ fails each test for k-independence is $p_1 = 1/|F|$, which is resulted using Lemma 4 in [5]. The number of k-1 element subsets under consideration is $\binom{h-1}{k-1}$. Applying union bound, the probability that $\mathbf{m}_e$ fails at least one of the tests for k-independence is $p_2 \le \binom{h-1}{k-1}\frac{1}{|F|}$. Considering the receivers in $Q(e)$, which counts to at most $|T|$, the probability that m_e fails at least one of the verifications for k-independence is $p_3 \le \binom{h-1}{k-1}\frac{|T|}{|F|}$. Therefore, the probability of k-independence of C_t, $t \in Q(e)$, is given by $p = 1 - p_3 \ge 1 - \binom{h-1}{k-1}\frac{|T|}{|F|}$.

□

If we set the success probability of the design of m_e for edge e as $p \ge 1 - \binom{h-1}{k-1}\frac{|T|}{|F|} \ge 1/2$, then we have

$$|F| \ge \binom{h-1}{k-1}.2|T|. \qquad (4)$$

Theorem 2: The expected running time of *RNC1*(h,k) is $O(|E||T|k^2\binom{h-1}{k-1})$.

Proof- The algorithm is executed edge by edge. There are at most $|E|$ active edges. Every edge is at most in $|T|$ flows. For edge e, at most $|Q(e)| \le |T|$ verification of k-independence is to be performed, each of which consists of $\binom{h-1}{k-1}$ tests. The expected running time of each test is $O(k^2)$. Thus, the expected running time for each edge is $O(|T|k^2\binom{h-1}{k-1})$ and the expected running time of the algorithm is $O(|E||T|k^2\binom{h-1}{k-1})$. □

RNC1 is independent of link failure patterns and increases the overall resistance of the network against link failure. As discussed in theorems 1 and 2, both the complexity and the field size of *RNC1(h,k)* grow linearly with $\binom{h-1}{k-1}$.

Figure 1 demonstrates an example. We define a multicast session from the node a, to the two nodes m and n. Assuming all the edges have unit capacities, the capacity of the network is 3. We apply *RNC1*(*3*,*2*) to increase the robustness of the network against link failure. Due to equation 4, we can consider $|F| = 11$, for example. The global encoding vectors are denoted on each edge. In this example $\binom{h-1}{k-1} = 2$.

Table 1 presents the performance (rate of successful decoding) of *RNC1*(*8*,*k*) in the presence of failure patterns, which reduce the capacity down to k. The results are obtained by evaluating a sufficiently large number of multi-layer networks with capacity 8 in the presence of a sufficiently large number of random failure patterns. We observe that the rate of successful decoding approaches 100% by increasing the field size.

In the next section, we introduce an algorithm which guarantees maximum throughput for h-k flow failures for each receiver.

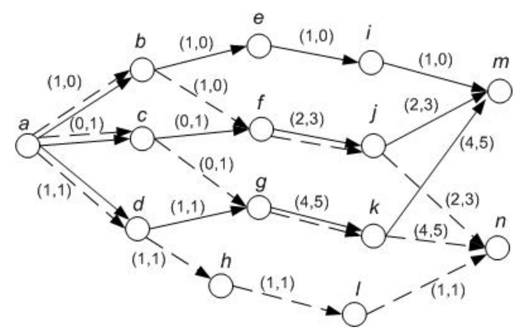

Figure 1- A network in which data is multicast from node a to nodes m and n. The multicast capacity is 3. Three flows of the nodes m and n are shown by directed solid and dashed lines, respectively. Global encoding vectors of the edges for *RNC1*(*3*,*2*) with $|F| = 11$ are denoted.

Field Size	k=3	k=4	K=5	k=6	k=7
$|F| = \left\lceil \binom{h-1}{k-1}.2|T| \right\rceil_p$	97.8	99.4	99.5	98.7	91.5
$|F| = \left\lceil \binom{h-1}{k-1}.8|T| \right\rceil_p$	99.5	99.9	99.9	99.7	98.4

Table 1- The performance (rate of successful decoding (%)) of *RNC1*(*8*,*k*) for failure patterns that reduce the capacity down to k for various k. $\lceil a \rceil_p$ denotes the smallest prime number greater than a.

IV. Improved Robust Network Coding

In *RNC1*(h,k), the receivers can extract all data by receiving the symbols of any k flows out of all h edge-disjoint flows. A flow fails if at least one link of the flow fails. Using *RNC1*, if there is only one receiver, failing any set of h-k flows, the receiver can extract all the data using the remaining flows. However, in the scenario with more than one receiver, receiving k flows for each receiver is not sufficient for extracting all the data, since the failed flows of other receivers may have affected the contents of the received symbols. For example, in figure 1, the flow $f_1^n = [(a,b),(b,f),(f,j),(j,n)]$ joins into $f_2^m = [(a,c),(c,f),(f,j),(j,m)]$ in one edge. Failure of link (b, f) destroys f_1^n, but does not directly fails any flows of m. However, this still affects the flow f_2^m, in the sense that it may lead to collapsing the k-independence of the B_t at receiver m (decoding failure). Therefore, to guarantee k-independence in the presence of h-k flow failures for each receiver, we must examine these joint flows. A *joint-flow* is a flow that has at least a common edge with at least one flow of another receiver. A *joint-failure* is a failure pattern in which at least one joint-flow fails. Here, we present an improvement to *RNC1*, referred to as *RNC2*, to overcome joint-failures. In *RNC2*, in each edge e, the local encoding vector, $\mathbf{m}_e$, is verified for k-independence for all $t \in Q(e)$ for failure-free and all joint-failure patterns that do not reduce the capacity below *k*. Therefore, the constructed code resists all such joint-failure patterns.

Now, assume n_t flows of *h* flows of the receiver *t* are joint-flows. R_t^k denotes the set of all joint-failure patterns for the flows of receiver *t* that do not reduce the capacity below *k*. $R_{t,i}$ denotes the set of joint-failure patterns for receiver *t* in which *i* joint-flows fail. We have $R_{t,i} = \binom{n_t}{i}$, and

$$\left|R_t^k\right| = \sum_{i=0}^{h-k} \binom{n_t}{i}. \qquad (5)$$

Note that $\binom{a}{b} = 0$ if $b > a$. R^k is the set of all joint-failure patterns that do not reduce the capacity below *k*. Thus,

$$\left|R^k\right| = \prod_{t\in T} \left|R_t^k\right| = \prod_{t\in T} \left[\sum_{i=0}^{h-k} \binom{n_t}{i}\right]. \qquad (6)$$

In *RNC2*, the local encoding vector of an edge is examined for all joint-failure patterns one by one.

Theorem 3: In *RNC2(h,k)*, if the coefficients of $\mathbf{m}_e$ are selected randomly from the finite field *F*, the probability that the set C_t, $\forall t \in Q(e)$, is *k*-independent for all the specified failure patterns is as follows,

$$p \geq 1 - \binom{h-1}{k-1} \frac{|T|}{|F|} \cdot \left|R^k\right|. \qquad (7)$$

Proof- The argument follows from the proof of theorem 1. The details are omitted for brevity. □

If we set the success probability for each edge as $p \geq 1 - \binom{h-1}{k-1} \frac{|T|}{|F|} \cdot \left|R^k\right| \geq 1/2$, then we have,

$$|F| \geq \binom{h-1}{k-1} . 2|T| \cdot \left|R^k\right| \qquad (8)$$

Theorem 4: The expected running time of *RNC2*(*h*,*k*) is $O(|E||T|k^2 . \left|R^k\right| . \binom{h-1}{k-1})$.

Proof- In each edge *e*, *k*-independence is to be verified for all the specified failure patterns. Thus, the number of tests is multiplied by $\left|R^k\right|$. The argument follows from the proof of theorem 2. The details are omitted for brevity. □

A pseudo-code for *RNC2* is presented in figure 2. The proposed *RNC2* algorithm using a field of size given in equation 8, with the expected running time presented in theorem 4, resists against joint-failure patterns and thus guarantees the resistance against *h-k* flow failures for each receiver. Our simulations validated the mentioned design objective.

Comparing to the *RNC1* algorithm, the expected running time and the field size of *RNC2* algorithm increases by a factor of $\left|R^k\right|$. We can compute an upper bound for $\left|R^k\right|$ as follows,

$$\left|R^k\right| \leq \prod_{t\in T} \left[\sum_{i=0}^{h-k} \binom{h}{i}\right] = \left[\sum_{i=0}^{h-k} \binom{h}{i}\right]^{|T|} \qquad (9)$$

Note that the algorithm introduced in [5], guarantees the rate *k* for all link failure patterns, which do not reduce the capacity below *k*. This advantage is achieved at the cost of a substantial increase in the complexity and field size. Specifically, the complexity is $O(|E||T|h^2|H|)$ and the field size is $|F| \geq 2|T||H|$, where *H* is the set of failure patterns considered. In a common network, $|H|$ is a large number and grows exponentially with the number of edges. In the example of figure 1, $n_1 = 3$ and $n_2 = 3$, and $\left|R^k\right| = 16$. Therefore, the complexity of *RNC1* and *RNC2* is proportional to $\binom{h-1}{k-1} = 2$ and $\binom{h-1}{k-1} . \left|R^k\right| = 32$, respectively, which is much less than the number of all failure patterns that do not reduce the capacity below 2, $|H| = 1761$. *RNC2* outperforms *RNC1* in terms of robustness; while *RNC1* already provides acceptable performance as depicted in table I.

RNC2(*h*,*k*):
h is calculated from Max-flow Min-cut theorem
F is a finite field satisfying equation 4

for each $t \in T$ **do**
  find Φ^t :the set of *h* disjoint flows from *s* to *t*
  find Φ^t_{joint} : the set of joint flows of the receiver *t*

find R^k : the set of all joint-failure patterns that do not reduce the capacity below *k*

for each t $\in T$ **do**
  initialize C_t
  for each $f \in R^k$ **do**
    initialize $B_{f,t}$

for each edge *e* in topological order **do**
  for each $t \in Q(e)$
    $C_t \leftarrow C_t \setminus \{\Gamma^t_{\leftarrow}(e)\} \cup \{e\}$
  select $\mathbf{m}_e$ randomly

  for each $f \in R^k$ **do**
    ($Q^f(e)$: the set of receivers using the edge e in the joint-failure pattern *f*)
    ($\Gamma_I^f(v)$: the set of active input edge of the node *v* in the failure pattern *f*
    for all $t \in Q^f(e)$
      $\mathbf{b}_f(e) = \sum_{e' \in \Gamma_I^f(tail(e))} \mathbf{m}_e(e') \mathbf{b}_f(e')$
      $B'_{f,t} = B_{f,t} \setminus b_f \{\Gamma^t_{\leftarrow}(e)\} \cup \{\mathbf{b}_f(e)\}$
      if $B'_{f,t}$ is not *k*-independent
        Go to **select**
  $B_{f,t} = B'_{f,t}$

Figure 2- The pseudo-code for *RNC2*(*h*,*k*)

References

[1] R. Ahlswede, N. Cai, S. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, July 2000.

[2] S-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Trans. Inform. Theory*, 49:371–381, Feb. 2003.

[3] R. Koetter and M. Medard, "An algebraic approach to network coding," *IEEE/ACM Trans. Networking*, 2003.

[4] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," *International Symposium on Information Theory (ISIT)*, 2003.

[5] S. Jaggi, P. Sanders, P. A. Chou, M. Effros, S. Egner, and L. Tolhuizen, "Polynomial time algorithms for multicast network code construction," *IEEE Trans. Inf. Theory*, Vol. 51, No.6, 2005.

[6] C. Fragouli, E. Soljanin, and A. Shokrollahi, "Network Coding as a Coloring Problem," *Proceeding of CISS*, 2004.

[7] P. A. Chou, Y. Wu, and K. Jain. "Practical network coding," *In Proc. 41st Allerton Conf. Communication*, Control and Computing, Monticello, IL, Oct. 2003. Invited paper.

[8] R. W. Yeung and N. Cai, "Network error correction, part I: Basic concepts and upper bounds," *Comm. in Inform. and Systems*, vol. 6, no. 1, pp. 19–36, 2006.

[9] R. W. Yeung and N. Cai, "Network error correction, part II: Lower bounds," *Comm. in Inform. and Systems*, vol. 6, no. 1, pp. 37–54, 2006.

[10] R. Matsumoto, "Construction Algorithm for Network Error-Correcting Codes Attaining the Singleton Bound," *IEICE Trans. Fundamentals*, VOL.E90–A, NO.9 Sept. 2007.

[11] T. Ho, M. Medard, R. Koetter, D. R. Karger, M. Effros, J. Shi and B. Leong, " A random linear network coding approach to multicast," *IEEE Trans. Inf. Theory*, Vol. 52, No. 10, pp. 4413- 4430, 2006.

[12] T. Ho, M. Medard and R. Koetter, "An Information-Theoretic View of Network Management," *IEEE Trans. Inf. Theory*, Vol. 51, No. 4, 2005.

[13] S. Y. El Rouayheb, A. Sprintson, and C. Georghiades. "Simple Network Codes for Instantaneous Recovery from Edge Failures in Unicast Connections," Technical Report WCL-TR-06-101, Texas A&M University, College Staion, Texas, January 2006.