IEEE Fourth Workshop on Network Coding Theory and Applications (NetCod), Hong Kong, Jan. 3-4, 2008.

Robust Network Coding Using Diversity through Backup Flows

Hossein Bahramgiri and Farshad Lahouti
Wireless Multimedia Communications Laboratory
School of ECE, College of Engineering
University of Tehran
h.bahramgiri@ece.ut.ac.ir, lahouti@ut.ac.ir

***Abstract*- We introduce algorithms to design robust network codes in the presence of link failures for multicast in a directed acyclic network. Robustness is achieved through diversity provided by the network links and flows, while the maximum multicast rate due to max-flow min-cut bound is maintained. The proposed scheme is a receiver-based robust network coding, which exploits the diversity due to the possible gap of the specific receivers min-cut with respect to the network multicast capacity. An improved version of this scheme guarantees multicast capacity for a certain level of failures. In a multicast session, failure of a flow may not necessarily reduce the capacity of the network as other useful branches within the network could still facilitate back up routes (flows) from the source to the sinks. We introduce a scheme to employ backup flows in addition to the main flows to multicast data at maximum rate *h*, when possible. In a limiting case, the scheme guarantees the rate *h*, for all link failure patterns, which do not decrease the maximum rate below *h*. Here, the number of link failures may in general exceed the refined singleton bound.**

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

I. INTRODUCTION

Network coding is a field of much recent interest in information theory aiming at designing codes for network nodes in order to improve the performance by devising an enhanced communication systems network layer. In particular, network coding has been initially motivated for improving network throughput in multicast applications. This article deals with the design of robust network codes in the presence of link failures, while achieving the multicast rate bound due to max-flow min-cut theorem. The proposed scheme can be viewed as a joint network and channel coding approach in a directed acyclic network.

Ahlswede *et al*. introduced a new target for multicast in a network by evaluating the capacity and showing that it can be achieved only by network coding [1]. Next, Li *et al*. showed that the capacity can be achieved even by linear coding and finite alphabet size [2]. Koetter and Medard demonstrated an algebraic scheme for network coding and studied information flows [3]. Ho *et al*. proposed a simple random scheme for network coding [4], in which the probability of success is enhanced by increasing finite field size. The approach is decentralized and consequently more proper for networks with unknown or time variant topologies. Jaggi *et al*. in [5] introduced a centralized algorithm to find the coding solution in polynomial time. They introduced deterministic linear network coding (DLIF), a completely deterministic algorithm, and random linear network coding (RLIF), which is random in the middle stages. Fragouli *et al*. in [6], develop a distributed and deterministic method for network code design by translating the problem as a graph coloring problem. More recently, Jabbarihagh and Lahouti presented a decentralized approach based on learning to design network codes [7]. The scheme may be viewed as a smart random search which facilitates the code design especially for large networks and small field sizes.

In the perspective of separate network and channel coding, ergodic errors in different links are combated by channel coding techniques in the data link layer. Some joint network-channel coding schemes have been presented in the literature for robust network code design. These schemes comprise two categories.

In the first category, link failure is considered. Koetter and Medard in [3], showed that there is a network code which can support a determined rate, *k*, for all failure patterns that do not reduce the capacity below this value. In [5], Jaggi *et al*. introduced a network code for this case, in which all such failure patterns are to be recognized and checked during code design. Thus, the scheme is complicated for practical situations. Ho *et al*. in [8], showed the benefits of random network coding for robustness in the presence of network changes and link failures. Chou *et al*. in [9] suggested a practical network coding algorithm, based on the random scheme of [4], presenting the concept of data packet generations, and aiming to increase the robustness to packet loss, delay and variations in network topology. In [10], receiver-based and network-wide recovery schemes are analyzed and network management is quantified by some results. Also, it is shown that there is not a single coding scheme for all recoverable single-link failures and at least the receivers must react to the failure model. In [11], El Rouayheb *et al*. investigate network codes that facilitate instantaneous recovery from single edge failure for unicast connections. In [12], using a trade off of multicast rate and diversity, a coding scheme referred to as robust network coding, RNC(h,k), is proposed where for a network with multicast capacity, h, a multicast rate of k is guaranteed in presence of at most h-k flow failures. The scheme is flow-based and thus the design complexity is manageable.

The second category is initiated by the work of Cai and Yeung [13], in which network error correcting codes are presented. The target is to correct (or detect) a determined number of edge errors. The work is elaborated in [14][15]. The Hamming, Singleton and Gilbert-Varshamov bounds for network codes are studied. Zhang in [16] and Yang *et al.* in [17] independently present a refined singleton bound. This bound is sink-dependent acknowledging the possible difference in the min-cut value of different sinks. Other works in this category includes [18]-[21]. In a recent work, Koetter and Kschischang study error correction in random network codes [22]. The code is designed at the transmitter, based on vector spaces which are preserved through the network, and the structure of the network does not affect the design.

This work falls in the first category. In this paper, we present algorithms to design robust network codes in the presence of link failures, using intrinsic diversity provided by the network branches and flows. The diversity exploited does not reduce the achievable multicast rate below max-flow min-cut bound. We introduce a receiver-based robust network coding scheme, RNC1(h+,h), which exploits the diversity due to the possible gap of the specific receivers unicast capacities (receiver min-cut) with respect to the network multicast capacity. An improved version of RNC1, RNC2(h+,h), is also presented to guarantee the transmission rate h, for a certain level of failures within the network. To design the local encoding vectors, we employ the scheme of [5]. The idea of using the diversity through the beyond-multicast-capacity-flows, is similar to the refinement in singleton bound in the works of the [16] and [17]. The proposed algorithms could be viewed as exploiting a multi-user diversity within the network for robustness in the presence of link failure.

Failure of a route (flow) may not necessarily reduce the multicast capacity of the network due to existence of other useful branches within the network. Once carefully considered in the network code design process, such branches in fact could facilitate back up flows from the source to the sinks in face of link failures. We introduce RNC3, to employ backup flows in addition to the main flows to transmit the data at rate h, when possible. In a limiting case, the scheme guarantees the transmission rate h, for all link failure patterns, which do not decrease the maximum transmission rate below h.

As evident from the above discussion, in the first category and the proposed scheme, the design objective is to facilitate sufficient intact flows to maintain the capacity following a failure, as opposed to the number of edge errors and erasures and the refined singleton bound, which is the criterion in the second category. Note that, for the proposed scheme, the number of failures in active links of receiver t, with the min-cut value of h_t may in general, be much more than h_t-h; which is the ultimate number of tolerable failures for network error correcting codes based on achieving refined singleton bound.

The rest of this paper is organized as follows: In section II, we describe the system in detail and present our motivations for robustness through diversity within the network. In the next three sections, RNC1, RNC2 and RNC3 are introduced, respectively, and their complexity and field size are analyzed. Finally the schemes are compared and discussed in section VI.

II. Motivations

A. System Description

The network can be represented by a graph G=(V,E), which is assumed to be directed and acyclic, with V and E representing the set of nodes and the set of links or edges, respectively. All links have unit capacity; thus a link with a positive integer capacity of c, is shown by c parallel links. The link e=(u,v,i) is the i-th link from u=*tail*(e) to v=*head*(e), which may be shown by e=(u,v) where i=1. A node $v \in V$ has a set of inputs links, denoted by $\Gamma_I(v)$. The sequence of links $f^t = [(s,b_1),(a_2,b_2),...,(a_{k-1},b_{k-1}),(a_k,t)]$ is a flow of the receiver t if for i=1,…,k, $b_{i-1} = a_i$. Here, we assume the network is synchronous. We consider multicast session in the network with a transmitter, $s \in V$ and a set of receivers, $T \subset V$. The network encounters link failures, in which the failed links can not transmit any data in the failure time. The objective is to increase the robustness of the network against link failures.

B. Robustness through Diversity of Redundant Flows

Consider a multicast session in a network, modeled by graph G. The capacity of the session is computed, applying max-flow min-cut theorem, as follows [1]

$$c_{multicast} = \underset{t \in T}{Min}\{Maxflow(t)\}. \qquad (1)$$

In an error-free network, one can design a network code to achieve the transmission rate $h = c_{multicast}$. In this case, it is enough to find h flows from the transmitter to each receiver. The rest of the links, which are not present in these flows are not required for this session. As we shall demonstrate, it is possible to employ the rest of the links for diversity. Thus, the robustness of the transmission is increased by employing what we refer to as the *redundant flows* in a multicast session. Such flows may be identified by considering the max-flow of each receiver denoted by h_t, for receiver t, which is greater than or at least equal to h.

Figure 1 shows a simple example. The node s multicasts data to the nodes y and z. We can define the following flows for the receivers:

$$f_1^y = [(s,t),(t,y)]$$

$$f_2^y = [(s,u),(u,w),(w,x),(x,y)]$$

$$f_1^z = [(s,t),(t,w),(w,x),(x,z)]$$

$$f_2^z = [(s,u),(u,z)]$$

$$f_3^z = [(s,v),(v,z)]$$

We can see that Maxflow(y)=2, and Maxflow(z)=3. The multicast capacity of the session is equal to the minimum max-flow, or 2. Thus, the sender can send two linear combinations of the two symbols in each time unit. But there is an extra path from the source s to z, which can be employed to increase robustness. The source can send another linear combination of the two symbols in the third path, such that any two of the three linear combinations are independent. Thus, receiving two combinations is enough to extract the data. The degree of diversity is equal to $h_t - h$ for the receiver t. In the example of figure 1, h=2. We can send two combinations of 2 source symbols to y, and three combinations to z. Based on the concept of redundant flows, two robust network coding schemes are presented in sections III and IV.

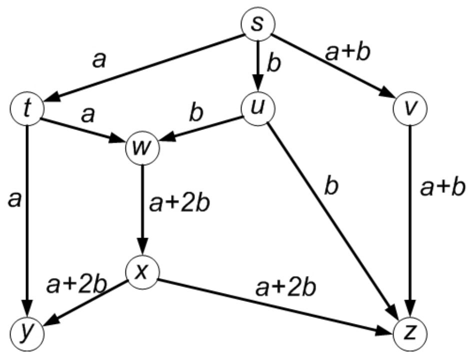

Figure 1- A simple example for multicast: The node s multicasts data to y and z.

C. Robustness through Backup Flows

Consider a network with multicast capacity of h. In the classic network coding, one could find h flows from the source to each receiver and design the code for the determined set of flows. The other links of the network are not used in the transmission scheme. Here, we wish to employ these links, as backup links, for increased robustness. It is possible that the original set of flows fail to transmit the rate h in the event of link failure, but one can still find at least h flows from the intact links and backup links, which can support the transmission rate of h. For a receiver t, it is possible that more than h_t-h original flows fail, but one can find h or more flows from s to t. In the example network of figure 1, failure of the edges (s,t) and (u,z) fails the flows f_1^z and f_2^z, respectively. But there are still two flows from the node s to the node y:

$f = [(s,t),(t,w),(w,x),(x,y)]$ and $f_3^{\,z}$. The flow f is present, but not considered, in the error-free network.
Thus, the second approach to robust network code design, elaborated in section V, is to use backup links for constructing *backup flows* to replace those that possibly fail within the network.

III. ROBUST NETWORK CODING THROUGH DIVERSITY OF REDUNDANT FLOWS

In this section, we introduce a robust network code, RNC1(h+,h), designed to utilize redundant flows for improved performance in the presence of link failure as described in section II.B. The algorithm is based on a centralized approach, and is inspired by the work of Jaggi *et al*. in [5]. However, the ideas and techniques presented for code design could also be cast into other design algorithms.
The scheme is centralized and consists of two steps. The first step is to find the flows for each of the receivers. Thus, we find $\Phi^t = \{f_1^t, ..., f_{h_t}^t\}$, the set of h_t flows from the source to the receiver t. It is possible that the flows of two receivers have common edges, but the flows of one receiver are edge-disjoint. An edge that is present in at least one flow is referred to as an *active edge*.
In the second step, we construct a scheme to transmit the rate h from the source to each of the receivers and employ redundant flows. The idea is to transmit h_t linear combination of h independent source symbols in h_t flows of the receiver t, such that every h symbol subset is enough to extract the source symbols. The source s generates h independent symbols, $X = \{x_1, ..., x_h\}$, which is to be transmitted to the receivers in T. Every active edge carries a symbol, which is a linear combination of the elements of X. The global encoding vector, **b**(e), is a vector of length h, which consists of the combination coefficients of the symbol in the edge e. The global encoding vector of an active edge, e, is a linear combination of the global encoding vectors of the edges in the set $\Gamma_I(tail(e))$. The combination coefficients are determined by local encoding vector, $\mathbf{m}_e$, as follows

$$\mathbf{b}(e) = \sum_{e' \in \Gamma_I(tail(e))} \mathbf{m}_e(e')\mathbf{b}(e') \,. \qquad (2)$$

The set C_t is defined for receiver t, consisting of the last considered edges in h_t edge-disjoint flows, transmitting h_t 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 h-element subsets forms an independent set of vectors. We refer to this property as *h-independence*. To initialize the algorithm, we add a new node s' to the network and connect it to s with $h_{\max} = \max_{t \in T}(h_t)$ directed edges from s' to s. Therefore, we have the new graph $G' = (V', E')$ with $V' = V \cup \{s'\}$, and $E' = E \cup \{(s',s,1),...,(s',s,h_{\max})\}$. The set C_t is initialized by the set of added edges, $\{(s',s,1),...,(s',s,h_t)\}$. 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 will be 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. Also, $\Gamma_{\leftarrow}^t(e)$ is the previous edge of the edge e in the flow of t. After replacement, h-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 2. Next, the resulting B_t is verified for h-independence for each $t \in Q(e)$. The following theorem shows the condition for the presence of at least one valid local encoding vector.

Theorem 1: A local encoding vector for each and every network edge exists in RNC1(h+,h), if the size of the employed finite field F satisfies the following condition

$$|F| \geq |T| \cdot \binom{h_{\max}-1}{h-1} \qquad (3)$$

Proof- We show that in each edge, local encoding vector, $\mathbf{m}_e$, can be computed, or equivalently there is a linear combination of the global encoding vectors of the active edges in $\Gamma_I(tail(e))$ such that h-independence is satisfied after replacing $\Gamma_{\leftarrow}^t(e)$ by e in C_t, for all of the receivers in $Q(e)$. For the h-independence verification of the receiver t, $\binom{h_t-1}{h-1}$ independence tests are done. Thus, the number of independence tests, denoted by N, is at most $|Q(e)| \cdot \binom{h_{\max}-1}{h-1}$. We prove the theorem by induction on the number of independence tests denoted by n. The vector $\mathbf{m}_e$ is of size $|Q(e)|$. We order randomly the receivers in $Q(e)$ as $t_1, ..., t_{|Q(e)|}$. For n=1, consider the receiver $t_1 \in Q(e)$ and $C_{t_1}^{(1)}$, an h element subset of C_{t_1}, which includes the edge e. Consider $\mathbf{e}_i$ as a $|Q(e)|$ element vector of zeros with only a 1 at position i. If $\mathbf{m}_{e,1}$=$\mathbf{e}_1$, we have $\mathbf{b}(e,1)$=$\mathbf{b}(\Gamma_{\leftarrow}^{t_1}(e))$, which satisfies the independence of the updated $B_{t_1}^{(1)}$, related to $C_{t_1}^{(1)}$. Now, assume that $\mathbf{m}_{e,n}$ satisfies the independence for n tests. Assume that the (n+1)-th test is one of the tests of the h-independence of the receiver t_k for the subset $C_{t_1}^{(n+1)}$. In this step, the independence of $B_{t_k}^{(n+1)}$ is tested. If $\mathbf{m}_{e,n}$ satisfies independence of $B_{t_k}^{(n+1)}$, then $\mathbf{m}_{e,n+1}$ = $\mathbf{m}_{e,n}$; Otherwise, the resulted $\mathbf{b}_{e,n+1}$ is in the subspace of $B_{t_k}^{(n+1)} \setminus \mathbf{b}(\Gamma_{\leftarrow}^{t_k}(e))$. Thus, we consider $\mathbf{m}_{e,n+1}$ = $a.\mathbf{m}_{e,n}$ + $\mathbf{e}_k$, which satisfies independence for the (n+1)-th test, for all $a \in F$. We now check the independence for each of the previous tests. Each test fails for at most one value of a. Since, if the test fails for two combinations of u= a_1. $\mathbf{m}_{e,n}$ + $\mathbf{e}_{k(n+1)}$ and v= a_2. $\mathbf{m}_{e,n}$ + $\mathbf{e}_k$, the test fails for u-v=(a_1- a_2). $\mathbf{m}_{e,n}$, which contradicts the induction hypothesis. Thus, if the parameter a is selected from a field of size $|F| \geq n+1$, then there exists a linear combination that passes all the tests. As we know, $N \leq |T| \cdot \binom{h_{\max}-1}{h-1}$, thus the equation 3 is obtained.

□

In this scheme, $\mathbf{m}_e$ is selected randomly. The following theorem determines a bound for the probability of success in designing $\mathbf{m}_e$ in each random trial as a function of the field size.

Theorem 2: In RNC1(h+,h), if the coefficients of $\mathbf{m}_e$ are selected randomly from the finite field F, the probability of h-independence of the sets B_t, $t \in Q(e)$, is given by

$$p \geq 1-\binom{h_{\max}-1}{h-1}\frac{|T|}{|F|}, \quad (4)$$

where $h_{\max}=\max_{t\in T}(h_t)$ and $\binom{a}{b}=\frac{a!}{b!(a-b)!}$.

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, $\Gamma^t_{\leftarrow}(e)$ is replaced by *e*. The vector $\mathbf{m}_e$ is selected randomly. Using equation 2, we examine a candidate of **b**(*e*) for *h*-independence for the updated B_t. Specifically, the vector **b**(*e*) with every *h*-1 element subset of the global encoding vectors of h_t-1 other edges in C_t must construct an independent set. The probability that **b**(*e*) fails each test of *h*-independence is $p_1=1/|F|$, which is resulted using Lemma 4 in [5]. The number of ($h-1$)-element subsets under consideration is $\binom{h_t-1}{h-1}$. Applying union bound, the probability that $\mathbf{m}_e$ fails at least one of the tests for *h*-independence is $p_2 \leq \binom{h_t-1}{h-1}\frac{1}{|F|}$. Considering the receivers in $Q(e)$, the probability that $\mathbf{m}_e$ fails at least one of the verifications for *h*-independence is $p_3 \leq \sum_{t\in Q(e)}\binom{h_t-1}{h-1}\frac{1}{|F|}$. Therefore, the probability of *h*-independence of C_t, $t \in Q(e)$, is given by $p=1-p_3 \geq 1-\sum_{t\in Q(e)}\binom{h_t-1}{h-1}\frac{1}{|F|}$. As we know, $|Q(e)| \leq |T|$ and $h_t \leq h_{\max}$. Thus $p \geq 1-\binom{h_{\max}-1}{h-1}\frac{|T|}{|F|}$.

□

Theorem 3: The expected number of operations of RNC1(*h*+,*h*) is $O(|E|.\max\{1,|T|^2 h^2 \binom{h_{\max}-1}{h-1}^2 . \frac{1}{|F|}\})$.

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 $|Q(e)| \leq |T|$ verification of *h*-independence is to be performed, each of which consists of at most $\binom{h_{\max}-1}{h-1}$ tests. The expected number of operations of each test is $O(h^2)$. Thus, the expected number of operations for each test of local encoding vector is $O(|T|k^2\binom{h_{\max}-1}{h-1})$. The probability of failure of a test of the local encoding vector is $q \leq \binom{h_{\max}-1}{h-1}\frac{|T|}{|F|}$, due to theorem 2. Thus, the expected number of operations of the scheme to find proper local encoding vector in an edge is $O(\max\{1,|T|k^2\binom{h_{\max}-1}{h-1}.q\})$, and the expected number of operations of the scheme is $O(|E|.\max\{1,|T|^2 k^2\binom{h_{\max}-1}{h-1}^2.\frac{1}{|F|}\})$.

□

Note that by increasing the field size, the expected number of operations decreases, but in contrast the complexity of each operation is increased. In the next section, we introduce an extension of RNC2 to guarantee the rate *h* for h_t-h flow failures for each receiver *t*.

IV. Rate-Guaranteed Robust Network Coding

In RNC1(*h*+,*h*), receivers can extract all data by receiving symbols of each *h* flows of *h* edge-disjoint flows. A flow fails if at least one link of the flow fails. Using RNC1(*h*+,*h*), if there is only one receiver *t*, failing any set of h_t-*h* flows, the receiver can extract all the data using the remaining flows. However, in the scenario with more than one receiver, receiving *h* 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.

In figure 1, we can see that f_1^z has common edges with f_1^y and f_2^y. Failure of link (*s, t*) destroys f_1^y and f_1^z, but does not directly fails any other flow. However, this still affects the flow f_2^y, in the sense that it may lead to collapsing the *h*-independence of the B_t at receiver *y* (decoding failure), since the flow f_1^z joins into f_2^y, another flow of *y*. Therefore, to guarantee *h*-independence in the presence of h_t-*h* 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(*h*+,*h*), to overcome joint-failures. In RNC2(*h*+,*h*), in each edge *e*, the local encoding vector, $\mathbf{m}_e$, is verified for *h*-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. For each edge *e*, the node *v*=tail(*e*) sends zero, if all the flows passing the edge *e* fail, or otherwise, send a packet based on the computed local encoding vector. Also, the receivers react to failure patterns, by changing the decoding matrix.

Now, assume n_t flows of h_t flows of the receiver *t* are joint-flows. R_t denotes the set of all joint-failure patterns for the flows of receiver *t* that do not reduce the capacity below *h*. $R_{t,i}$ 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

$$|R_t|=\sum_{i=0}^{h_t-h}\binom{n_t}{i}. \quad (5)$$

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

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

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

Theorem 4: A local encoding vector for each and every network edge exists in RNC2(h+,h), if the size of the finite field F satisfies the following

$$|F| \ge |T| \cdot |R| \binom{h_{\max}-1}{h-1}. \qquad (7)$$

Proof- The argument follows from the proof of theorem 1. Here, the number of independence tests, is at most $|Q(e)| \cdot |R| \cdot \binom{h_{\max}-1}{h-1}$, since the tests are repeated for each of $|R|$ joint-failure patterns. Also, for each pattern, the elements of $\mathbf{m}_e$ related to the edges of the failed flows are set to zero. □

Theorem 5: In RNC2(h+,h), if the coefficients of $\mathbf{m}_e$ are selected randomly from the finite field F, the probability that the set B_t, $\forall t \in Q(e)$, is h-independent for all the specified failure patterns is as follows,

$$p \ge 1 - \binom{h_{\max}-1}{h-1} \frac{|T|}{|F|} \cdot |R|. \qquad (8)$$

Proof- The argument follows directly from the proof of theorem 2. The details are omitted for brevity. □

Theorem 6: The expected number of operations of RNC2(h+,h) is $O(|E|.\max\{1, |T|^2 h^2 |R| . \binom{h_{\max}-1}{h-1}^2 . \frac{1}{|F|}\})$.

Proof- In each edge e, h-independence is to be verified for all the specified failure patterns. Thus, the number of tests is multiplied by $|R|$. The argument follows from the proof of theorem 3. The details are omitted for brevity. □

V. Robust Network Coding Via Backup Flows

In this section, we introduce RNC3, in which backup flows are employed when some main flows fail to transmit data. An example was presented in section II.C. The scheme is not limited to one set of flows, as we design one set of local encoding vectors for all flow selections. In the limiting case the scheme guarantees the transmission rate h for every failure pattern, which does not reduce the capacity below h.

We consider Ψ_t as the set of all flows from s to t. The flows may have common edges. Each h-element subset of the set Ψ_t, which consists of h edge-disjoint flows, can provide a multicast rate h. The set Ω_t has all such subsets as elements. If at least one of these subsets is completely safe, the max-flow from the node s to the node t is equal to h. A complete-flow-set $\theta = \{\omega_1, ..., \omega_{|T|}\}$ is a set of subsets $\omega_t \in \Omega_t$ for all $t \in T$. The set of all possible complete-flow-sets is denoted by Θ. It can be easily shown that $|\Theta| = \prod_{t \in T} |\Omega_t|$.

After finding all subsets composed of h disjoint flows for all $t \in T$, we design the coding scheme for all edges. To initialize the scheme, we add the node s' and h edges from s' to the node s, and find h independent linear combination of h source symbols, as the global encoding vectors of these edges. Here, an active edge is an edge, which is present in at least one of the h disjoint-flow subsets. The algorithm considers the active edges, one by one. The edge e is considered, only after all active edges in $\Gamma_I(tail(e))$ have been processed. The edge e replaces $\Gamma^t_{\leftarrow}(e)$ in $C_{t,\omega}$ for each receiver $t \in Q(e)$ and each subset $\omega \in \Omega_t$, for which the edge e is in one of their flows. After replacement, independence is verified for each updated $B_{t,\omega}$. The algorithm concludes when the edges ending at the receivers are processed.

Theorem 7: A local encoding vector for each and every network edge exists in RNC3, if the size of the finite field F satisfies the following condition

$$|F| \ge |T| \cdot \prod_{t \in T} |\Omega_t|. \qquad (9)$$

Proof- In this case, the local encoding vector is tested for all complete-flow-sets. Thus, the number of independence tests is equal to $N = \sum_{c \in C} |Q_c(e)|$. We know that $N \le |T| \cdot |\Theta| = |T| \cdot \prod_{t \in T} |\Omega_t|$. Here, the size of $\mathbf{m}_e$ is equal to the number of input active edges of the node $tail(e)$. For each complete-flow-set, the elements of $\mathbf{m}_e$ related to the edges that are not present in the complete-flow-set, is set to zero. The argument follows from the proof of theorem 1. □

Theorem 8: In RNC3, if the coefficients of $\mathbf{m}_e$ are selected randomly from the field F, the probability that the sets $B_{t,\psi}$ of all h-disjoint flows $\omega \in \Omega_t$ for all $t \in Q(e)$ are independent, is as follows,

$$p \ge 1 - \frac{|T|}{|F|} \prod_{t \in T} |\Omega_t|. \qquad (10)$$

Proof- The argument follows from the proof of theorem 2. The details are omitted for brevity. □

Theorem 9: The expected number of operations of RNC3 to guarantee the rate h for all failure patterns that do not reduce the capacity below h is $O(|E|.\max\{1, |T|^2 h^2 .[\prod_{t \in T} |\Omega_t|]^2 . \frac{1}{|F|}\})$.

Proof- In each edge e, h-independence is to be verified for all complete-flow-sets. Thus, the number of tests is multiplied by $|\Theta| = \prod_{t \in T} |\Omega_t|$. The details are omitted for brevity □

Usually, there are more than one set of h disjoint flows for each receiver. We select the set randomly. The other remaining flows do not participate in the transmission scheme. The optimization of the set selection is not considered here, and remains open to new ideas. The pseudo-code of RNC3 is presented in figure 2.

For each edge e, the node v=tail(e) sends zero, if all the flows passing the edge e fail, or otherwise, send a packet based on the computed local encoding vector. Also, the receivers react to failure patterns, by changing the decoding matrix.

It is possible to consider only some h disjoint flow subsets of the set Ψ_t for each t, and find the coding scheme for them. For example, there may be some sensitive flows in the network, which can be supported by some backup flows. The scheme finds a unique coding scheme for the failure-free case and the cases in which some backup flows replaces some main flows. Note that, the theorems 7, 8 and 9 are resulted for the limiting case, where all the *h*-disjoint-flow subsets of the set Ψ_t for each *t* are considered in the scheme.

RNC3:
h_t is calculated from Max-flow Min-cut theorem for all $t \in T$
F is a finite field satisfying equation 9

for each $t \in T$ **do**
find Ψ_t :the set of all flows from *s* to *t*
find Ω_t : the set of all *h*-disjoint-flows of the receiver *t*

find Θ : the set of all complete-flow-sets

for each $\theta \in \Theta$ **do**
for each $t \in T$ **do**
initialize $C_{t,\omega_{t,\theta}}$ and $B_{t,\omega_{t,\theta}}$

for each active edge *e* in topological order **do**
for each $\theta \in \Theta$ **do**
for each $t \in Q^{\theta}(e)$
$C_{t,\omega_{t,\theta}} \leftarrow C_{t,\omega_{t,\theta}} \setminus \{\Gamma_{\leftarrow}^{t,\theta}(e)\} \cup \{e\}$
select $\mathbf{m}_e$ randomly
for each $\theta \in \Theta$ **do**
if $e \in E_\theta$ (E_θ is the set of active edges of the complete-flow-set θ)

$$\mathbf{b}_\theta(e) = \sum_{e' \in \Gamma_I^{\theta}(tail(e))} \mathbf{m}_e(e')\mathbf{b}_\theta(e')$$

$$B'_{\theta,t} = B_{\theta,t} \setminus b_\theta\{\Gamma_{\leftarrow}^{t,\theta}(e)\} \cup \{\mathbf{b}_\theta(e)\}$$

if $B'_{\theta,t}$ is not *h*-independent
Go to **select**

$B_{\theta,t} = B'_{\theta,t}$

Figure 2- The pseudo-code for RNC3

VI. CONCLUDING REMARKS

We presented three flow-based network coding schemes to increase the robustness of the network against link failure. The RNC1(*h*+,*h*) is the simplest scheme, in which diversity and hence robustness is achieved through redundant flows. The scheme can be categorized as receiver-based [3],[12]. The RNC2(*h*+,*h*) guarantees the rate *h* for the failure of $h_t - h$ failures for each receiver *t* of the determined flows. The advantage is provided at the cost of increased design complexity by a factor of $|R|$, and the need to inform the flow failure to the nodes in the failed flows. In RNC2(*h*+,*h*), for each edge *e*, the node *v*=tail(*e*) sends zero, if all the flows passing the edge *e* fail, or otherwise, send a packet based on the computed local encoding vector. The flow failure is announced to the nodes in the failed flows, which requires feedback. It is possible that the flows fail to transmit the rate *h*, due to failure of more than $h_t - h$ flows for the receiver *t*, but there are still other flows present that can substitute the failed flows. Thus, we presented RNC3 scheme to employ backup flows. In a limiting case, the scheme can guarantee the rate *h* for all failure patterns, which do not decrease the capacity below *h*. The RNC3 can be applied to increase the robustness by compensating the sensitive points of the network via backup flows. In RNC3, in each state, we can select one *h*-disjoint set for each receiver and remove all other flows to decrease the global cost of the scheme. The benefits are achieved at the cost of the need to identify the safe flows via feedback.

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*, Oct. 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)*, July 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, June 2005.

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

[7] M. Jabbarihagh and F. Lahouti, "A decentralized approach to network coding based on learning," *Proc. IEEE Information Theory Workshop* (*ITW*), Norway, July 2007.

[8] 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.

[9] 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.

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

[11] 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.

[12] H. Bahramgiri, M. Jabbarihagh, F. Lahouti, "Robust Network Coding in the Presence of Link Failure Using a Multicast Rate-Diversity Trade off," *Proc. IEEE Information Theory Workshop* (*ITW*), Norway, July 2007.

[13] N. Cai and R. W. Yeung, "Network coding and error correction," in *Proc. IEEE ITW'02*, 2002.

[14] 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.

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

[16] Z. Zhang, “Network error correction coding in packetized networks,” *Proc. IEEE Information Theory Workshop* (*ITW*), Chengdu, China, pp.433–437, Oct. 22–26, 2006.

[17] S. Yang, C.K. Ngai, and R.W. Yeung, “Construction of linear network codes that achieve a refined Singleton bound,” *in Proc. ISIT 2007*, Nice, France, June 24–29, 2007.

[18] S. Yang and R. W. Yeung, "Characterizations of network error correction/detection and erasure correction," *in Proc. NetCod'07*, Jan. 2007.

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

[20] Z. Zhang, X. Yan and H. Balli, "Some key problems in network error correction coding theory," *Proc. IEEE Information Theory Workshop* (*ITW*), Norway, July 2007.

[21] H. Balli, X. Yan and Z. Zhang, "On failure Probability of random network codes," *Proc. IEEE Int. Symp. Information Theory*, Nice, France, June 2007.

[22] R. Koetter and F. R. Kschischang, "Coding for errors and erasures in random network coding," *Proc. IEEE Int. Symp. Information Theory*, Nice, France, June 2007.