

Network Coding-Based Link Failure Recovery over Large Arbitrary Networks

Serhat Nazim Avci and Ender Ayanoglu

Center for Pervasive Communications and Computing
Department of Electrical Engineering and Computer Science
University of California, Irvine
Irvine, CA 92697-2625

Abstract—Network coding-based link failure recovery techniques provide near-hitless recovery and offer high capacity efficiency. Diversity coding is the first technique to incorporate coding in this field and is easy to implement over small arbitrary networks. However, its capacity efficiency is restricted by its systematic coding and high design complexity even though it has lower complexity than the other coding-based recovery techniques. Alternative techniques mitigate some of these limitations, but they are difficult to implement over arbitrary networks. In this paper, we propose a novel non-systematic coding technique and a simple design algorithm to implement the diversity coding-based (or network coding-based) recovery over arbitrary networks. The design framework consists of two parts. An ILP formulation for each part is developed. The simulation results suggest that both the novel coding structure and the novel design algorithm lead to higher capacity efficiency for near-hitless recovery. The new design algorithm is able to achieve optimal results in large arbitrary networks.

I. INTRODUCTION

The information carried by wide area networks is, in general, very important. Yet these networks regularly undergo failures. Detailed statistics about network failures can be found in [1]. This paper focuses on recovery from single link failures since they consist of 70% of all network failures. To minimize the cost of such failures, various restoration and protection techniques are developed. The two main metrics in the design of these techniques are restoration speed and capacity efficiency. Capacity efficiency is measured by the total required capacity, in terms of fiber miles, and restoration speed is measured by the duration between the occurrence of failure and restoration of failed traffic. The goal is to minimize both of these metrics and every technique offers a different tradeoff.

In some recovery techniques, spare resources are shared among different traffic failure scenarios and different connection demands, whereas in others, spare resources are dedicated to connection demands. Dedicated protection techniques are able to offer near-hitless recovery since they do not require the signaling and rerouting of the failed traffic. 1+1 Automatic Protection Switching (APS) is a dedicated protection technique where two link-disjoint paths for each connection demand are employed to transmit the same data to the destination node. In the case of a link failure over the primary path, the destination node switches to the protection path and restores the traffic

nearly instantaneously. However, 1+1 APS requires more than 100% capacity which makes it capacity inefficient. The fact that 1+1 APS is currently employed in today's networks [2] indicates the need for nearly instantaneous link failure recovery despite its low capacity efficiency.

The capacity efficiency of dedicated protection schemes such as 1+1 APS can be improved if the dedicated paths are shared. This can be achieved by employing coding, in particular, erasure coding [3], [4]. The technique introduced in [3], [4], called diversity coding, has two advantages. First, unlike 1+1 APS, it is capacity efficient. Second, unlike rerouting-based restoration schemes, the recovery is nearly instantaneous. References [3], [4] predate network coding, usually considered to be introduced in [5].

In [6], diversity coding is implemented over arbitrary networks using a heuristic algorithm. In [1], optimal algorithms for the diversity coding technique are developed. Diversity coding performs near-hitless recovery while offering competitive capacity efficiency. In [7], a solution of Shared Path Protection (SPP) [8] is converted to a coding-based solution named Coded Path Protection (CPP). Sharing of the spare resources is replaced with the employment of these resources to code different paths. This conversion increases the restoration speed and the transmission integrity, and decreases error signaling complexity. The bidirectional nature of CPP allows efficient encoding and decoding inside the network for unicast demands.

In [9] and [10], network coding-based protection schemes called 1+N protection are proposed in which coding operations are carried out over trees and trails, respectively. The idea is similar to that of diversity coding except the protection is bidirectional. In [11], the cost efficiencies of a network coding-based recovery technique and a simpler version of diversity coding technique are evaluated.

All of the above mentioned techniques implement systematic coding where coding operations are bound to specific protection topologies and primary paths are exempt from coding operations. In addition, they require strict link-disjointness between each primary path and the protection paths. Even though these assumptions make those techniques easier to implement, they restrict the capacity efficiencies of them.

In [12], the primary paths are incorporated into coding operations using a heuristic algorithm for static provisioning. The decodability of the coding structures is preserved by ran-

This work is partially supported by NSF under Grant No. 0917176.

domly adding the connection demands to the existing coding groups one by one. A coding group is a set of connection demands that are coded and protected together. Coding primary paths increases capacity efficiency over conventional diversity coding, as in [12]. Non-systematic coding is implemented in wireless mesh networks for single link failure recovery in [13]. In [14], a general network-coding based approach is presented which employs non-systematic coding and does not explicitly require link-disjointness between primary paths and protection paths. However, this approach is restricted to specific topologies. In addition, it can protect at most two connection demands simultaneously. In [15], the proposed technique lifts the restriction over the number of protected connection demands for bidirected networks. In general, the coding-based recovery techniques in the literature, such as [9], [14], [15], offer promises in terms of capacity efficiency and restoration time. However, they cannot be optimally implemented in real networks due to their high design complexity limitations. The test networks and traffic matrices in those papers are much smaller than real networks.

This paper offers two novel contributions to the field of diversity coding-based (or network coding-based) link failure recovery. First, we introduce an optimal, simple, and modular design algorithm that provisions the static traffic in relatively large networks. The underlying coding structure of this algorithm is arbitrary as long as the destination nodes of the connections are the same, which offers a solution for different techniques under the same framework. Second, we improve the coding structure of simple diversity coding by offering an optimal non-systematic coding structure using an Integer Linear Programming (ILP) formulation. In a non-systematic coding structure, both primary and protection paths are incorporated into the coding groups. The performance of the new proposed coding technique is investigated compared to conventional (systematic) diversity coding using the novel design algorithm. The performance of the new design algorithm is also tested based on a set of simulations over a relatively large U.S. long-distance network.

II. DESIGN ALGORITHM

The link failure recovery problem has two main components, namely an underlying recovery technique and a design algorithm. A recovery technique can achieve its potential performance only with a fast and optimal design algorithm that maps it over the networks of interest. Therefore, some recovery techniques can be theoretically advanced but they may perform poorly on test networks due to the high complexity of the accompanying design algorithm.

We developed a simple, optimal, and modular design algorithm for arbitrary single destination coding-based recovery techniques. The novelty is decomposing the design process into two parts, a pre-processing phase and the main problem solving phase. The design process is depicted in Fig. 1. The pre-processing phase has as its input the network graph and the destination node. In the pre-processing phase, all candidate coding groups are listed and their total cost to route and protect are calculated. These calculations depend on the underlying recovery technique. The size of a coding group is limited by the nodal degree of the destination node. If a

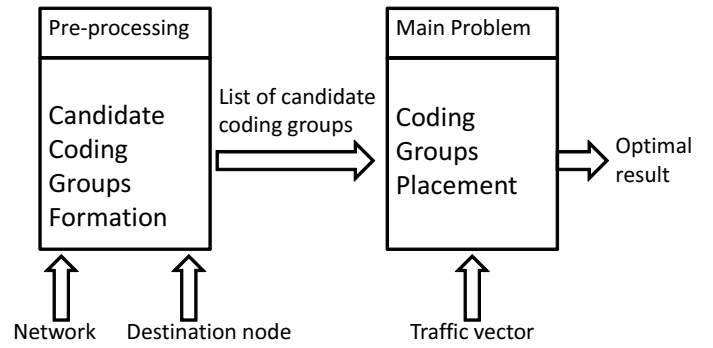


Fig. 1. Two phase design process leads to the optimal result.

coding group is not feasible, it has infinite total cost. The candidate coding group list is given as input to the main problem solving phase. In the main problem solving phase, those coding groups are optimally chosen and placed over the network such that all of the traffic demands are routed and protected through a coding group. The traffic matrix is decomposed into smaller vectors based on the destination nodes of the connections and input to the main problem. The main problem is inspired by the p -cycle algorithm in [16]. Both the coding groups formation and placement operations are carried out by ILP formulations which have dramatically fewer number of variables and constraints than those of [1], [9], and [11].

For example, there is a network with three nodes, S_1 , S_2 , and D . Assume that there are 3 units and 2 units of traffic from S_1 and S_2 to D , respectively. In this scenario, there are three candidate coding groups with source nodes $[(S_1), (S_2), (S_1, S_2)]$. Assume that the total cost vector of these coding groups are $[5, 10, 12]$. The main problem inputs candidate coding groups with their cost vector and tries to minimize the total cost by satisfying the traffic demands. In the optimal solution, two units of coding group (S_1, S_2) and one unit of coding group (S_1) are placed over the network with minimum total cost 29.

In [17], enumeration of p -cycles is argued to be slow. However, the nature of our problem is a better fit to enumeration of the candidate protection structures due to three reasons. First, the number of candidate coding groups is much smaller than the number of candidate p -cycles, when a single destination node is employed. Second, enumeration and placement of p -cycles are a spare capacity placement (SCP) operation and the routing of the primary paths must be handled separately. However, in our case, a coding group both routes and protects the connections, which is joint capacity placement (JCP). Third, the number of constraints in the p -cycle algorithm equals the number of edges, but the number of constraints in our algorithm is equal to the number of nodes.

The total number of candidate coding groups is an important criterion in terms of design complexity. For the typical coding-based recovery techniques, which employ a single destination node, the total number of candidate coding groups is proportional to

$$\binom{|V| - 1}{ND - 1} + \binom{|V| - 1}{ND - 2} + \dots + \binom{|V| - 1}{1}, \quad (1)$$

where ND is the nodal degree and $|V|$ is the number of nodes in the network. $ND - 1$ is the largest size of a coding group and $|V| - 1$ defines the size of the list of source nodes a coding group choose from. ND is the most important parameter defining the complexity of the new algorithm. On the other hand, the size of the traffic matrix is negligible in terms of complexity since traffic demands only take place in the right hand side of the constraints of the ILP formulation of the main problem. They do not affect the number of variables or constraints in the ILP formulation. We want to make the important point that the proposed algorithm is also robust to changes in the traffic matrix. If the traffic matrix changes over time, there is no need to carry out the pre-processing phase again. The right hand side of the constraints in the main problem can be changed to optimize the network in response to changing traffic. The main problem is very fast since it has only $|V| - 1$ constraints.

III. NON-SYSTEMATIC CODING

In this section, we introduce an ILP-based optimal non-systematic diversity coding structure for single link failure recovery.

We assume that the connection demands in the same coding group have a common destination node. Their source nodes can be the same or different. There are N connection demands in a coding group. Each connection demand has two link-disjoint paths carrying the same signal, which is distinct from other connection demands. Some of these paths are combined and coded together and some of them are not combined with any other path. For simplicity, we assume all of the operations are over $GF(2)$, although this assumption can be relaxed, e.g., [3], [4]. The paths in a coding group are assigned to subgroups. The total number of subgroups varies between $N + 1$ and $2N$. The number of paths in a subgroup take values from zero to N . The paths in the same subgroup are assumed to be coded together. In the received vector of the destination node, each connection demand is represented as a variable and each subgroup is represented as an equation. Clearly, if there are smaller than or equal to N subgroups, some data cannot be recovered in some failure scenarios because that leaves $N - 1$ equations for N unknowns. In the opposite extreme, there will be a maximum $2N$ subgroups if each path is transmitted separately, which is the case in 1+1 APS.

In conventional diversity coding against single link failures, there exists N primary paths, one for each connection demand, and a single protection path carrying the modulo-2 sum of the data over primary paths. Each connection demand is delivered to the destination node over two link-disjoint paths. It has a total of $N + 1$ subgroups, N of them are the primary paths and one of them is the combination of protection paths. The protection path topology can be a tree if it is formed by combining paths originating from different source nodes. The coding operations are restricted over the protection path (tree). The common destination node carries out the decoding operation over the received vector. The example in Fig. 2, taken from [12], shows how non-systematic coding can reduce the total capacity for the protection of a coding group. The paths in a non-systematic code are equivalent to each other and therefore cannot be categorized as primary and protection

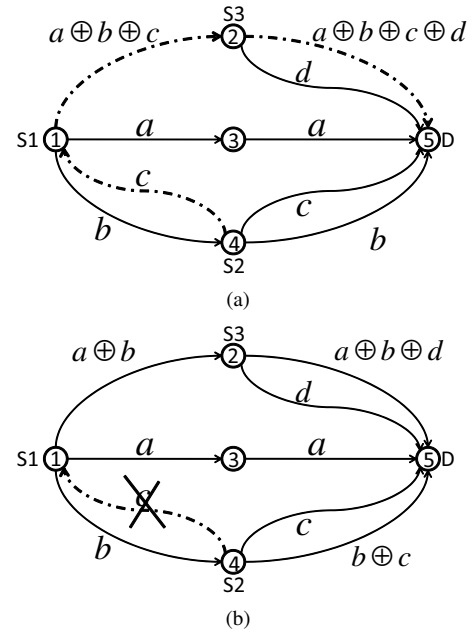


Fig. 2. Capacity efficiency improvement with non-systematic coding, (a) Diversity coding solution, (b) Non-systematic coding solution

paths. There are four connection demands destined to node D . Two of them are originated from $S1$, represented by symbols a and b . The other two are originated from $S2$ and $S3$, represented by symbols c and d , respectively. All four connection demands form a coding group. In Fig. 2(a), a typical diversity coding solution is depicted. The common protection path is shown with dashed lines. In Fig. 2(b), a non-systematic coding solution is depicted. It enables what was once the protection path of c to be coded with what was once the primary path of b over nodes 4 – 5. That coding operation eliminates the need for the link between 4 – 1 carrying c . Therefore, non-systematic coding can improve the capacity efficiency. In the worst case scenario, it performs the same as systematic conventional diversity coding.

A non-systematic code can be built by assigning paths to the subgroups arbitrarily. However, the critical point in the construction of a non-systematic code is the decodability of all N transmitted signals. The N data signal can be decoded under any single link failure scenario as long as any N equations of the received vector are linearly independent. It is clear that any subset of linear equations with size N of the received vector are independent in systematic diversity coding. The received vector of systematic diversity coding for four connection demands is

$$\begin{bmatrix} a \\ b \\ c \\ d \\ a + b + c + d \end{bmatrix}, \quad (2)$$

where a , b , c , and d are transmitted signals by each connection demand. $N + 1$ subgroups are sufficient for systematic diversity coding. In non-systematic coding, the paths in each subgroup must be specified. In [12], connection demands are randomly chosen and paths are assigned one by one to subgroups of the

existing coding groups. However, a general rule is needed to optimally build non-systematic codes. In [13], it is reported by *Lemma 1* that the destination node can recover N data signals from a non-systematic code as long as any subset of the data signals with size k are transmitted over at least $k + 1$ paths. In our technique, *Lemma 1* changes to

Lemma 1. *The non-systematic code will be valid as long as any subset of data signals with size k are members of at least $k + 1$ subgroups in a coding group.*

The proof follows from [13], assuming U_s as the set of connection demand signals and L_s as the set of subgroups in a coding group.

This paper aims to build valid non-systematic codes with the objective of minimizing total capacity. Therefore, we develop an optimization algorithm to find the code that requires lowest total capacity while eliminating the codes that violate *Lemma 1*. The following example shows how an invalid non-systematic code can be detected. Assume we have four connection demands, carrying signals a , b , c , and d in a coding group and each connection demand has two link-disjoint paths. Assume the first three subgroups of this coding group are given as

$$\begin{bmatrix} a + b \\ b + c \\ c + d \end{bmatrix}, \quad (3)$$

which indicate that one path of a and b , b and c , and c and d are coded together. That leads to a coding relationship map shown in Fig. 3(a). In this map, there are two symbols for each connection demand, referring to their two link-disjoint paths. In Fig. 3(b), a bidirectional arrow between two paths means they are in the same subgroup and therefore coded together. If a path of a is coded together with a path of b and a path of b is coded together with a path of c , then connection demand a is indirectly related to connection demand c , which is shown with a dashed arrow in Fig. 3(b). In addition, pairs $a - d$ and $b - d$ are indirectly related as well. If the fourth subgroup consists of $a + d$ then four connection demands are bounded within four subgroups, which is a violation of *Lemma 1*. In Fig. 3(c), the relationship map is updated to include a bidirectional arrow between a path of a and a path of d . As a result, connection demands a and d are coded together and indirectly related at the same time, which causes a circle shown in Fig. 3(d). We call this a coding circle, which is an indication of the violation of *Lemma 1*. Therefore, in the ILP formulation, we seek to prevent coding circles by ensuring two different connection demands can either be coded together or are indirectly related. The resulting non-systematic code will be valid as long as coding circles are prevented.

IV. ILP FORMULATIONS

A. Candidate Coding Groups Formation

An ILP formulation is developed to implement the proposed technique with an objective to minimize the total capacity (cost) of a coding group in arbitrary networks. The ILP formulation finds the optimum non-systematic diversity coding structure by simply going through all possible subgroup assignments for each path and eliminating the ones which violate *Lemma 1*. The input parameters of the ILP are

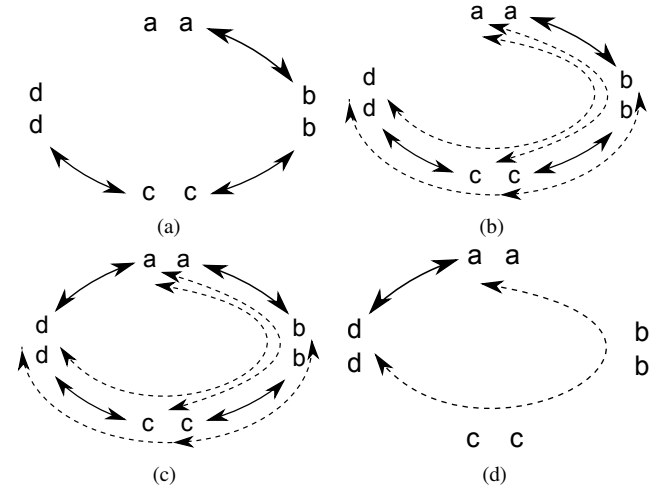


Fig. 3. Formation of a coding circle. A coding circle violates *Lemma 1*.

- $G(V, E)$: Network graph,
- S : The set of spans in the network, there are two opposite directional links in each span,
- N : Enumerated list of all connections,
- P : Enumerated list of all paths, $|P| = 2 \times |N|$,
- a_e : Cost associated with link e ,
- $\Gamma_i(v)$: The set of incoming links of each node v ,
- $\Gamma_o(v)$: The set of outgoing links of each node v ,
- s_i : The source node of path i ,
- d : The destination node.

The variables related to finding two paths for each connection are

- $x_e(i)$: Equals 1 iff the path i passes through link e , 0 otherwise.

The following inequality finds two paths for each connection demand

$$\sum_{e \in \Gamma_i(v)} x_e(i) - \sum_{e \in \Gamma_o(v)} x_e(i) = \begin{cases} -1 & \text{if } v = s_i, \\ 1 & \text{if } v = d, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Note that we require $\text{mod}(i, 2) = 0 \Rightarrow s_i = s_{i-1}$ for $1 \leq i \leq 2N$ since the same demand is carried by two successively indexed paths. The variables related to finding a valid non-systematic code are

- $n(i, s)$: Equals 1 iff path i is in subgroup s , 0 otherwise,
- $m(i, j)$: Equals 1 iff path i and path j are in the same subgroup so are coded together, 0 otherwise,
- $r(i, f)$: Equals 1 iff path i and connection demand f are indirectly related, 0 otherwise.

Each path must be assigned to a single subgroup which is ensured by

$$\sum_{s=1}^{2N} n(i, s) = 1 \quad \forall i, \quad (5)$$

$$n(i, s) + n(i - 1, s) \leq 1 \quad \forall i, s : \text{mod}(i, 2) = 0, \quad (6)$$

$$m(i, j) \geq n(i, s) + n(j, s) - 1 \quad \forall i \neq j, s. \quad (7)$$

Inequality (6) ensures that paths carrying the same data cannot be in the same subgroup. If two paths are in the same

subgroup, then they are assumed to be coded together, which is satisfied by inequality (7).

$$r(i, f) \geq m(i, j) + m(j^*, 2f) + m(j^*, 2f - 1) - m(i, 2f) - m(i, 2f - 1) - 1 \quad \forall i, j, f : i \neq j \quad (8)$$

where $j^* = j - 1$ if $\text{mod}(j, 2) = 0$ and $j^* = j + 1$ otherwise.

$$r(i, f) \geq r(i, g) + m(2g, 2f) + m(2g, 2f - 1) + m(2g - 1, 2f) + m(2g - 1, 2f - 1) - 1 \quad \forall i, f \neq g : i \neq 2f, i \neq 2f - 1, i \neq 2g, i \neq 2g - 1, \quad (9)$$

$$r(2f, g) + r(2f - 1, g) + m(2f, 2g) + m(2f - 1, 2g) + m(2f, 2g - 1) + m(2f - 1, 2g - 1) \leq 1 \quad \forall g, f : g \neq f. \quad (10)$$

In inequality (8), path i becomes indirectly related to demand f if path i is coded with path j and if there exists a path j that is coded with both path i and one of the paths carrying demand f . Moreover, path i must not be coded with either paths of demand f . Inequality (9) ensures that path i becomes indirectly related to demand f if path i is indirectly related to demand g and one of the paths carrying demand g is coded with one the paths carrying demand f . Inequality (10) ensures that only one of the paths carrying demand f can be either coded with one of the paths carrying demand g or be indirectly related to demand g . This inequality ensures the validity of the non-systematic code by preventing coding circles. The final variable of the ILP is

- $t_e(s)$: Equals 1 iff one of the paths in subgroup s traverses over link e , 0 otherwise.

$$t_e(s) \geq x_e(i) + n(i, s) - 1 \quad \forall e, i, s. \quad (11)$$

$$t_e(s_1) + t_e(s_2) + t_f(s_1) + t_f(s_2) \leq 1 \quad \forall e, f \in g, \forall g \in S, \forall s_1, s_2 \quad (12)$$

Inequality (11) finds the topology of each subgroup. The topology of a subgroup is the union of the protection paths of the connections in that subgroup. Inequality (12) ensures that the topologies of two subgroups are link-disjoint.

The objective function is

$$\min \sum_{e \in E} \sum_{s=1}^{2N} t_e(s) \times a_e. \quad (13)$$

B. Coding Groups Placement

We assume that there is a single destination node and the rest are source nodes. There is a single connection demand from each source node to the destination with varying traffic rates. Those connection demands can be split into unit demands and protected via different coding groups. The goal is to select and place the optimal combination of candidate coding groups that will cover the traffic demands with the lowest cost. In addition to the parameters in the previous section, the extra parameters are

- CG : The candidate coding groups list,
- $CG_{i,f}$: Equals 1 iff the candidate coding group i includes connection demand f , 0 otherwise
- c_i : Total cost (capacity) of coding group i ,

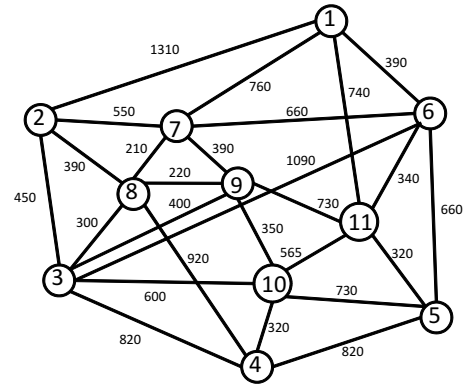


Fig. 4. European COST 239 network.

- t_f : Traffic rate of the connection demand f .

We have only one set of integer variables

- n_i : The number of units of coding group i that are placed.

The objective function is

$$\min \sum_{i=1}^{|CG|} c_i \times n_i \quad (14)$$

subject to

$$\sum_{i=1}^{|CG|} CG_{i,f} \times n_i \geq t_f \quad 1 \leq f \leq |V| - 1, \quad (15)$$

which ensures that the placed coding groups are sufficient to cover the traffic demands. Even though the coding group placement problem may have a large number of variables in large networks, the fact that it only has $|V| - 1$ constraints makes it achieve the optimal results in sub-second time.

V. SIMULATION RESULTS

In this section, we present two different simulations to investigate the performance of the proposed coding technique and the proposed design algorithm differentially. The first test network is COST 239 network, which is depicted in Fig. 4 with 11 nodes and 52 unidirectional edges. The numbers next to the links are the lengths (costs) of those links in km. There are 3 units of uniform traffic between each node pair to be protected against single link failures. The traffic matrix is decomposed into 11 independent traffic vectors, one for each node set up as the destination node. The simulations are repeated as many times as the number of the nodes. In each scenario, one of the nodes is selected as the destination node and we assume there are 3 units of traffic from every other node to the destination node. The performance metric is the spare capacity percentage (SCaP) as defined in [6]. The goal is to measure the decrease in SCaP due to the introduction of non-systematic diversity coding. We also investigate how the new simplified design algorithm enables us to achieve better (optimal) results for systematic diversity coding than a competitive technique. The competitive technique is chosen as diversity coding tree algorithm in [1] because it requires fewer number of variables and constraints than [9] and [11]. CPLEX 12.2 is used for the simulations.

TABLE I
SCaP RESULTS FOR EACH DESTINATION NODE

Destination Node	Diversity Coding Tree		Coding Groups Placement Algorithm			
			Systematic Diversity Coding		Non-systematic Diversity Coding	
	SCaP(%)	Optimality gap	SCaP(%)	Optimality gap	SCaP(%)	Optimality gap
Node 1	84.1	24.9	79.0	0.0	77.0	0.0
Node 2	74.8	23.3	69.5	0.0	68.6	0.0
Node 3	65.6	14.8	64.3	0.0	63.0	0.0
Node 4	88.7	22.8	87.6	0.0	84.1	0.0
Node 5	80.4	25.2	69.2	0.0	69.1	0.0
Node 6	91.5	21.3	84.7	0.0	83.7	0.0
Node 7	95.0	20.8	89.6	0.0	86.5	0.0
Node 8	106.1	19.8	99.9	0.0	99.2	0.0
Node 9	87.8	15.0	85.7	0.0	83.5	0.0
Node 10	116.7	24.9	106.6	0.0	104.5	0.0
Node 11	92.4	18.2	85.7	0.0	77.1	0.0
Average	87.9	21.5	82.1	0.0	80.9	0.0

The SCaP and optimality gap values of three different schemes are shown in Table I in terms of percentiles. Optimality gap is the duality gap in the linear programming context. If an algorithm achieves the optimal result, its optimality gap becomes zero. Systematic diversity coding via the new algorithm is derived by putting the secondary paths to the same subgroup in Section IV-A. Maximum number of candidate coding groups is equal to 3002 when the destination node is set to 3, which has the highest nodal degree of 6.

The simulation results highlight three important points. First, the proposed algorithm can achieve the optimal result in all cases, whereas diversity coding tree algorithm runs into memory limitations and cannot achieve optimal results. Second, even though the same systematic diversity coding is employed in the first and second algorithms, the proposed algorithm achieves better results since it can find the optimal result before the simulation terminates. It is noteworthy that if the optimality gap of the first algorithm is larger, then so is the difference between the SCaP values of the two different algorithms. The difference in terms of SCaP exceeds 10% in some scenarios. This clarifies the importance of the design algorithm for a recovery technique to achieve the promised results. The third point is the improvement in SCaP results due to the improvement in the coding structure. For all cases, non-systematic diversity coding performs better than its systematic counterpart.

The second test network is the U.S. long-distance network, taken from [19], which is shown in Fig. 5. The traffic matrix is created using a gravity-based model [20]. In this model, the number of connection demands between two nodes is proportional to the multiplication of the populations at these nodes. In total, there are 23,204 static unit connection demands. This setup is chosen in order to observe the performance of the new design algorithm in a large realistic network with a dense traffic scenario. The other coding-based recovery design algorithms are too complex to implement in this setup. The SCaP results and the complexity of the algorithm are compared with a p -cycle algorithm in [18, p. 699], which is considered to

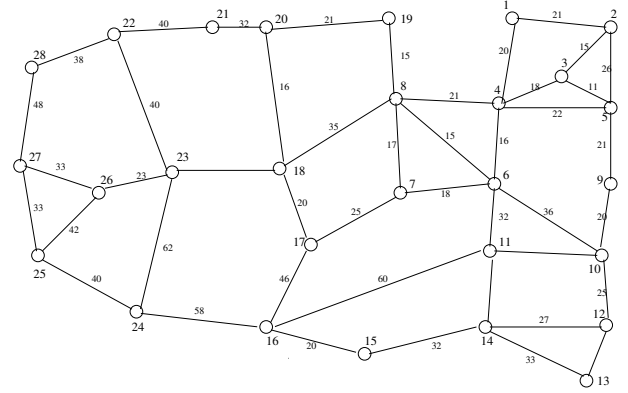


Fig. 5. U.S. long distance network.

be within 5% of the optimal solution. Both of the algorithms have a pre-processing phase where they enumerate all the candidate p -cycles or candidate coding groups. The results are presented in Table II.

As seen from the results, the proposed design algorithm can achieve optimal results with conventional diversity coding even in a large realistic network with a dense traffic scenario. The SCaP result of the new technique is better than that of the p -cycle algorithm. It should be noted that, p -cycle algorithm carries out SCP, whereas the proposed algorithm carries out JCP. One important difference is the design complexity between two techniques, which have a pre-processing phase. In the U.S. long-distance network, the number of candidate cycles is much higher than the number of candidate coding groups because the latter is constrained by the nodal degree of the network. As a future work, we plan to employ column generation technique, as done for p -cycles in [21], to make our algorithm even more scalable in larger networks.

VI. CONCLUSION

In this paper, we introduced an ILP-based non-systematic coding approach and a simple design algorithm to achieve near

TABLE II
COMPARATIVE PERFORMANCE OF THE NEW ALGORITHM IN U.S. LONG-DISTANCE NETWORK

Protection Technique	SCaP	Design complexity	Optimization Type
P -cycle algorithm [18]	107.0%	≥ 1000000 (p -cycles)	SCP
Coding groups placement for systematic diversity coding	105.6%	31464 (coding groups)	JCP

instantaneous recovery with higher capacity efficiency. Non-systematic coding allows any path in the coding group to be coded with other paths without compromising the decodability at the destination node. The code is developed with the objective of minimum capacity. These two advanced techniques combined achieve results with higher capacity efficiency. The advantages of both techniques are shown with examples and simulation results.

The new design framework consists of two parts, a pre-processing phase where the candidate coding groups are formed and the main problem solving phase where the optimal coding groups are placed over the network. We have developed an ILP formulation for each of these steps. In the pre-processing phase, coding groups are formed under the optimal non-systematic diversity coding principles. The main problem consists of only $|V| - 1$ constraints. It finds and places the optimal coding group combinations to match the traffic demands, which takes sub-second time to run. The new algorithm can be implemented over networks with arbitrary topology and it can achieve optimal results in large networks for arbitrary traffic scenarios.

We ran two sets of simulations. Non-systematic diversity coding has better capacity efficiency than conventional systematic diversity coding. In addition, we observe the significance of the simplicity of the design algorithm to achieve better results. In the later simulations, coding group placement algorithm is compared to a similar algorithm employing p -cycle protection over realistic U.S. long-distance network. The proposed algorithm achieves the optimal result.

REFERENCES

- [1] S. N. Avci and E. Ayanoglu, "Optimal algorithms for near-hitless network restoration via diversity coding," in *Proc. IEEE GLOBECOM*, December 2012, pp. 1–7.
- [2] I. B. Barla, F. Rambach, D. A. Schupke, and G. Carle, "Efficient protection in single-domain networks using network coding," in *Proc. IEEE GLOBECOM*, December 2010, pp. 1–5.
- [3] E. Ayanoglu, C.-L. I. R. D. Gitlin, and J. E. Mazo, "Diversity coding: Using error control for self-healing in communication networks," in *Proc. IEEE INFOCOM '90*, vol. 1, June 1990, pp. 95–104.
- [4] —, "Diversity coding for transparent self-healing and fault-tolerant communication networks," *IEEE Trans. Commun.*, vol. 41, pp. 1677–1686, November 1993.
- [5] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Trans. Inf. Theory*, vol. 46, pp. 1204–1216, July 2000.
- [6] S. N. Avci, X. Hu, and E. Ayanoglu, "Recovery from link failures in networks with arbitrary topology via diversity coding," in *Proc. IEEE GLOBECOM*, December 2011, pp. 1–6.
- [7] S. N. Avci and E. Ayanoglu, "Coded path protection: Efficient conversion of sharing to coding," in *Proc. IEEE ICC (to appear)*, June 2012.
- [8] S. Ramamurthy, L. Sahasrabudhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightwave Technol.*, vol. 21, no. 4, pp. 870–883, April 2003.
- [9] A. E. Kamal and O. M. Al-Kofahi, "Efficient and agile 1+N protection," *IEEE Trans. Comm.*, vol. 59, no. 1, pp. 169–180, January 2011.
- [10] A. E. Kamal, A. Ramamoorthy, L. Long, and S. Li, "Overlay protection against link failures using network coding," *IEEE/ACM Trans. Netw.*, vol. 19, no. 4, pp. 1071–1084, Aug. 2011.
- [11] H. Øverby, G. Bóizck, P. Barabarczy, and J. Tapolcai, "Cost comparison of 1+1 path protection schemes: A case for coding," in *Proc. IEEE ICC*, June 2012.
- [12] S. N. Avci and E. Ayanoglu, "Extended diversity coding: Coding protection and primary paths for network restoration," in *Proc. of the International Symposium on Network Coding*, June 2012, pp. 1–6.
- [13] O. M. Al-Kofahi and A. E. Kamal, "Network coding-based protection of many-to-one wireless flows," *IEEE J. Sel. Areas in Commun.*, vol. 27, no. 5, pp. 787–813, June 2011.
- [14] S. E. Rouayheb, A. Sprinston, and C. Georgiades, "Robust network codes for unicast connections: A case study," *IEEE/ACM Trans. Netw.*, vol. 19, no. 3, pp. 644–656, June 2011.
- [15] A. Sprinston, S. E. Rouayheb, and C. Georgiades, "Robust network coding for bidirected networks," in *Proc. of the Information Theory and Applications Workshop*, Jan.-Feb. 2007, pp. 378–383.
- [16] W. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. ICC '98*, vol. 1, 1998, pp. 537–543.
- [17] B. Wu, K. L. Yeung, and P.-H. Ho, "ILP formulations for p -cycle design without candidate cycle enumeration," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 284–295, February 2010.
- [18] W. D. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice-Hall PTR, 2004.
- [19] Y. Xiong and L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Trans. Netw.*, vol. 7, pp. 98–110, February 1999.
- [20] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proc. ACM SIGMETRICS*, June 2003.
- [21] T. Stidsen and T. Thomadsen, "Joint routing and protection using p -cycles," Informatics and Mathematical Modelling, Technical University of Denmark, DTU, Kgs. Lyngby, Denmark, Tech. Rep., May 2005.