Optimal Algorithms for Near-Hitless Network Restoration via Diversity Coding

Serhat Nazim Avci, *Student Member, IEEE*, and Ender Ayanoglu, *Fellow, IEEE*

***Abstract*—Coding-based restoration techniques have proactive restoration which results in time savings over other state-of-the-art restoration techniques. Diversity coding is a coding-based recovery technique which offers near-hitless restoration with a competitive spare capacity requirement with respect to other techniques. In this paper, we show that diversity coding can achieve sub-ms restoration time. In addition, we develop two optimal algorithms for pre-provisioning of the static traffic and one for the dynamic provisioning of the traffic on-demand. There is one algorithm for systematic and one for non-systematic diversity coding in pre-provisioning. An MIP formulation and an ILP formulation are developed for systematic and non-systematic cases, respectively. The MIP formulation of the systematic diversity coding requires much fewer integer variables and constraints than similar optimal coding-based formulations. In dynamic provisioning, an ILP-based algorithm covers both of the systematic and non-systematic diversity coding. In all scenarios, diversity coding results in smaller restoration time, higher transmission integrity, and much reduced signaling complexity than the existing techniques in the literature. Simulation results indicate that diversity coding has significantly higher restoration speed than Shared Path Protection (SPP) and *p*-cycle techniques from the literature as well as Synchronous Optical Network (SONET) rings, which are commonly deployed by service providers today. In terms of capacity efficiency, it outperforms SONET rings and $1+1$ APS, whereas it may require more extra capacity than the *p*-cycle technique and SPP. Diversity coding offers a preferable tradeoff which offers two orders of magnitude increase in restoration speed at the expense of less than 26% extra spare capacity.**

***Index Terms*—Network fault-tolerance, network coding, linear programming.**

I. Introduction

CABLE cuts in wide area networks are common. They happen approximately 4.39 times a year per 1000 sheath miles [1]. In this paper, we focus on recovery from single link failures which consist of 70% of all the failures [2], although our techniques can be generalized to multiple link failures. The prominent restoration techniques can be listed as ring-based restoration, mesh-based restoration, and the *p*-cycle technique [3], [4], [5]. Each technique offers a different tradeoff in terms of capacity efficiency and restoration speed. For voice traffic, telephone network industry set the restoration time goal to 50 ms, which is the lower bound of the failure perception time by users. For IP traffic, it is always desirable to decrease the restoration time to even smaller values due to the complexities introduced by the different layers of the networking hierarchy. In networking industry, hitless switching is considered to be the ultimate restoration technique, in which end nodes do not experience the failure even as a bit [3].

Mesh-based restoration techniques are grouped into two, namely path-based restoration and link-based restoration. The simplest form of path-based restoration techniques are $1+1$ and $1:1$ Automatic Protection Switching (APS). They reserve a link-disjoint dedicated backup path for every primary path. In this paper, link-disjointness actually refers to span-disjointness, where each span consists of two opposite directional links with arbitrary capacity. In $1+1$ APS, the backup path transmits the same data in the primary path at all times. The $1:1$ APS scheme can be extended to $N:M$ APS, which requires M link-disjoint backup paths to protect the N link-disjoint primary paths from any M link failures [3]. Mesh-based protection schemes employ sharing of the spare capacity among different primary paths in order to offer high capacity efficiency at the expense of lower restoration speed and higher signaling complexity. Shared-path protection (SPP) is a path-based restoration technique that has been studied extensively in both all-optical [6] and opaque [7] networks.

The accompanying restoration technique of the standard Synchronous Optical Network (SONET) is based on protection switching over reserved capacity of multi-node ring structures, known as self-healing rings or SONET rings. More than 100% capacity, in terms of fiber miles, is deployed over self-healing rings to match and protect all of the affected traffic over the failed links. However, due to geographical deployment of self-healing rings, required redundant capacity in fiber miles exceeds 100%.

A technique that combines the speed advantage of SONET rings and the capacity efficiency of mesh-based restoration is known as *p*-cycle protection [8]. *P*-cycle protection is faster than mesh-based techniques because it eliminates most of the cross-connect configurations that are required to reroute the traffic after the failure. The same idea is used in mesh-based techniques by [9] and [10], named as "hot-standby" and "pre-cross-connected trials" (PXT), respectively.

Coding based link failure recovery was introduced in [11], [12] and was called diversity coding. In single failure diversity coding, N primary links are protected using a separate $N+1^{st}$ protection link which carries the modulo-2 sum of the data

Manuscript received October 28, 2012; revised March 20 and June 4, 2013. The editor coordinating the review of this paper and approving it for publication was P. Popovski.

The authors are with the Center for Pervasive Communications and Computing, Department of Electrical Engineering and Computer Science, University of California, Irvine (e-mail: {savci, ayanoglu}@uci.edu).

This work was partially supported by the National Science Foundation under Grant No. 0917176. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the National Science Foundation.

This work was presented in part during the IEEE Global Communications Conference, Anaheim, USA, December 2012.

Digital Object Identifier 10.1109/TCOMM.2013.071013.120817

signals in each of the primary links. Assuming all $N+1$ links are disjoint, in other words physically diverse, then any single link failure can impair only one of them and the failed data can be extracted by applying modulo-2 sum to the received data. The biggest advantage of this technique is the fast automatic recovery from single link failures by eliminating the complex and time-consuming signaling and rerouting operations. The fact that a single protection link carries the coded data of N primary links leads to capacity savings. As a result, both the restoration speed and the capacity efficiency goals can be achieved, albeit within certain limits. Its proactive restoration simplifies the network management, minimizes the signaling overhead, and eliminates the instability threat that can be caused by the dynamic configuration of the optical cross-connects [13]. Diversity coding, like APS, can be generalized to multiple link failures by deploying M protection links to protect N primary links from M link failures. In [4], diversity coding is applied to arbitrary network topologies, using a heuristic algorithm. There, it is shown that diversity coding is much faster than a typical SPP technique known as source rerouting, and the p-cycle technique. In [14], both primary paths and protection paths are incorporated into coding operations which results in further capacity savings over typical diversity coding structures. The optimal algorithms of diversity coding for both pre-provisioning and dynamic provisioning are presented in [15]. It is shown that, diversity coding can achieve sub-ms restoration time if proper synchronization and buffering are implemented.

We introduced a technique that converts any sharing-based solution of SPP into a coding-based solution in [16], called as Coded Path Protection (CPP). The conversion makes the restoration automatic, faster, and simpler with some slight extra capacity. In that paper, it is shown that, coding-based restoration techniques preserve the transmission integrity after a link failure. CPP provides, in addition to encoding inside the network, decoding inside the network as has been sought for within the context of network coding.

We would like to note that although the publication of [11] and [12] predate the topic of network coding, diversity coding is a form of network coding. For the purposes of this paper, it has the goal of minimizing a distance metric, in addition to optimum erasure coding in a network.

Designing a survivable network against single link failures consists of two main steps, namely a restoration technique and a capacity placement algorithm. The performance of the design depends on both of these structures and it is evaluated by a number of different criteria. The restoration speed and the capacity efficiency are the most important metrics. Since the goal is to implement the restoration techniques in arbitrary networks, the complexity of the capacity placement algorithm plays a vital role for design purposes especially in big networks and dense traffic scenarios. Therefore, the challenge is to develop a sufficiently simple and efficient restoration technique jointly with a design algorithm that can optimally provision the traffic with low complexity. Both static and dynamic traffic consist of multiple unicasts between different nodes. A small improvement in the restoration technique can cause an exponential increase in the complexity of the optimal capacity placement algorithm. A theoretically superior restoration technique can result in inferior results if the accompanying design algorithm is not simple enough to find the near-optimum solutions with the limited resources. In this paper, optimal design techniques with low complexity are presented for diversity coding under static and dynamic traffic scenarios, respectively.

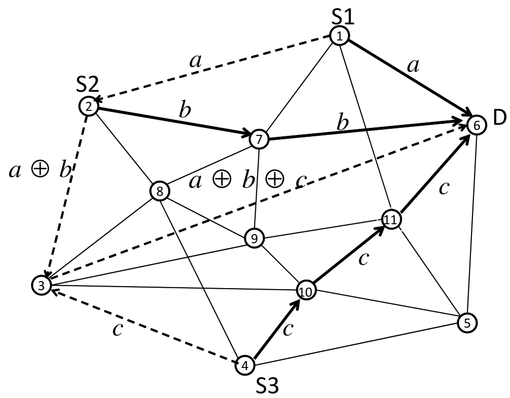
Coding-based recovery techniques have higher restoration speed than the rerouting-based techniques since they are proactive. In this paper, for the first time, the restoration speed of a coding-based recovery technique is quantified within sub-ms. In addition, the synchronization mechanism required for this operation is simpler than the competitive techniques in [15] and [17] since it only requires $N-1$ buffers for each coding group with N connection demands. Second, in this paper, we present an optimal design algorithm for diversity coding that can achieve competitive capacity efficiency compared to state-of-the-art recovery techniques. The optimal algorithm is based on Mixed Integer Programming (MIP) and is much simpler than the coding-based optimal algorithms in the literature e.g., [15], [17], and [18]. Third, we present the first coding-based dynamic provisioning algorithm for single link failure recovery. This algorithm is also optimal under a set of assumptions.

This paper consists of two parts. The first part addresses the pre-provisioning problem of the static traffic whereas the second part deals with the dynamic traffic.

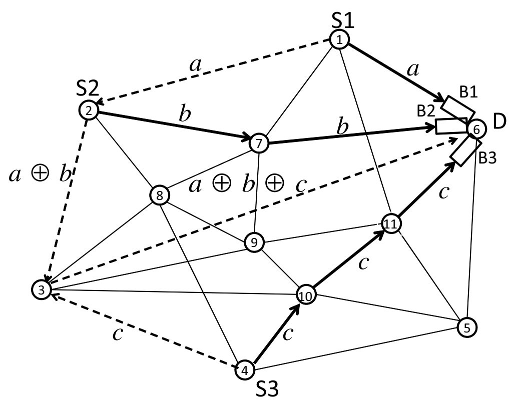
II. Diversity Coding Tree

In this section, we present how to achieve recovery within sub-ms in the case of single link failures and how to pre-provision the static traffic with an optimal design algorithm. The recovery technique we adopt is a form of diversity coding in which the coding operations are carried out among the connections with the same destination node. In this paper, there are three observations that leverage the simplicity of this coding structure. First, this coding structure achieves sub-ms restoration time in optical networks with a simple synchronization structure. Besides the restoration time and synchronization complexity, it also simplifies the signaling complexity. Second, it enables decomposition of the traffic matrix into smaller groups which decreases the design algorithm complexity without loss of optimality. The partitioned traffic subgroups can be input to parallel MIP formulations. Third, the nature of the coding structure helps eliminate some of the variables and constraints in the MIP formulation. This enables application of diversity coding on realistic networks. The optimal design algorithm is realized with an MIP formulation and called *Diversity Coding Tree* algorithm. In this algorithm, a primary tree structure serves as the primary paths and a protection tree serves as the protection paths of a set of connections. Moreover, the diversity coding tree inputs unidirectional connections. This flexibility adds strength to diversity coding tree technique in offering solutions for nonsymmetric traffic scenarios.

An example is provided in Fig. 1(a). In this figure, nodes $S1$, $S2$, and $S3$ transmit their data, given as a, b, and c, respectively, to the common destination node D. The primary tree is shown with solid black lines and arrows. The protection



(a)



(b)

Fig. 1. An example of the diversity coding tree structure (a) There are three link-disjoint primary paths spanned by the primary tree and there is a link-disjoint protection tree, (b) The synchronization buffers for the tree structure.

tree is disjoint to all these links used by the primary tree and it encodes the data to be protected on the tree structure, shown in this case with dashed black lines and arrows. In the sequel, we call the set of connections protected by the same set of primary and protection trees a *Coding Group.*

A. Synchronization and Buffering

In order to achieve near-hitless recovery, the arrival instances of the signals in the same coding group must be the same. If so, restoration time will be upper-bounded by a very small interval which includes failure detection, protection switching, and a single XOR operation. This is ensured via synchronization and buffering. We present a simpler synchronization mechanism than the one in [15]. In addition, the buffering delays are reduced. The example Fig. 1(a) is replicated in Fig. 1(b) with some additional features. The boxes $B1$, $B2$, and $B3$ are buffers that synchronize the primary tree and the protection tree for near-hitless switching. These buffers equalize the arrival instances of the primary signals and the protection (parity) signal. The delay values of the buffers can be calculated with the help of a variable $d^i_{x,y,z,v}$ which is equal to total time when signal i traverses from node x to node v over intermediate nodes y and z. Then buffer delays are calculated according to the example in Fig. 1(b), assuming $d^a_{1,2,3,6}$ is the longest path in the coding group, $d^b_{2,3,6} \geq d^b_{2,7,6}$, and $d^c_{4,3,6} \geq d^c_{4,10,11,6}$

$$B1 = d^a_{1,2,3,6} - d^a_{1,6} \quad (1)$$

$$B2 = d^b_{2,3,6} - d^b_{2,7,6} \quad (2)$$

$$B3 = d^c_{4,3,6} - d^c_{4,10,11,6}. \quad (3)$$

Total number of buffers is $N-1$, reduced by N compared to the synchronization mechanism in [15]. They are placed at the incoming links of the destination node except the link which carries the latest arriving signal of the decoding operation. The buffers at the intermediate nodes in [15] are eliminated. Assuming the protection tree is the longest path in the coding group, each signal on the primary tree is delayed as long as the propagation delay difference of the same signal between the primary and protection tree. The buffering delays are reduced by eliminating the synchronization mechanism over the protection tree, which does not compromise the decoding structure.

To achieve fine synchronization, either pointer processing or the Global Positioning System (GPS) can be used. In a simpler implementation, in packet networks, packet headers with sequence numbers can be employed for the same purpose. Besides the synchronization and buffering requirements, the nodes should be able to carry out XOR operations in high speed. One way is to carry out those operations in the optical domain as shown in [19]. The second alternative is to convert those signals into electrical domain and carry out these operations in the electrical domain. The destination node also has a small memory requirement for decoding purposes.

A frame structure enables the correction of bit errors via a two-dimensional mechanism. When one employs CRC checks on the bits of the primary paths, then one knows the existence of errors on those paths, then one can treat the frame with the bad CRC check as "erased" and recover it using the data in the protection path and the primary paths. Assuming N is the number of primary paths, there is one protection path, p is the raw Bit Error Rate (BER), and L is the size of a frame, the probability of the frame being in error is $1-(1-p)^L$. The calculation of the probabilities of error is complicated, but they are dominated by a polynomial in the form of $(1-p)^N$. Then, the probability of error after this mechanism is dominated by NLp^2, which is much smaller than the raw BER p for practical values of N, L, and p.

B. Diversity Coding Tree

The design algorithm for systematic diversity coding is given in this section. We call this algorithm when employed for pre-provisioning, diversity coding tree. In systematic diversity coding, there are link-disjoint primary paths and coded protection paths which are link-disjoint to primary paths. The optimal diversity coding tree algorithm uses ideas from a p-cycle approach that is based on a cycle exclusion technique [20]. The diversity coding tree algorithm is also simpler than the optimal algorithms of similar coding-based recovery techniques in [15], [17], and [18].

One of the novelties in the diversity coding tree algorithm is the way the primary and protection paths are formed. Instead of building the primary and protection paths of each connection with a separate variable in the MIP [17], the diversity coding tree algorithm builds trees that replace primary and protection paths for the connections that are protected together. Therefore, it carries out the same task with a significantly smaller number of variables and constraints. More details about the complexity analysis can be found in Section II-B6. There are two separate trees forming the primary and protection paths of the connections, respectively. In primary tree, there is a link-disjoint path from each source node to the destination node, defining the primary path of that source node. The primary tree consists of multiple link-disjoint branches originating from the source nodes that merge at the common destination node. Even though these branches do not share a link, the resulting structure is a tree in the context of graph theory. The protection tree serves as the common protection path for all of the connections protected by the same tree. The branches of this tree can merge until they reach the root of the tree, which is the destination node. The primary tree and the protection tree of the same diversity coding tree structure are link-disjoint.

The optimal MIP formulation of diversity coding tree technique is provided below. The input parameters are

- $G(V, E)$: Network graph,
- S : The set of spans in the network, a span consists of two links in opposite directions,
- N : Enumerated list of all unit-demand connections which have the destination node d,
- a_e : Cost associated with link e,
- T : Maximum number of diversity coding groups allowed, about half of the number of connections in each subproblem,
- $\Gamma_i(v)$: The set of incoming links of each node v,
- $\Gamma_o(v)$: The set of outgoing links of each node v,
- s_i : Source node of the connection demand i,
- d : The common destination node,
- ND_d : The nodal degree of the destination node d,
- α : A constant employed in the algorithm where $\frac{1}{|V|} \geq \alpha \geq 0$,
- β : A constant employed in the algorithm, $\beta \geq 2 \times \max(|V|, \max_i(ND_i))$.

Next we provide the variables. Except the last two, they are binary and take the value of 0 or 1.

- $n(i,t)$: Equals 1 iff connection i is routed and protected by the diversity coding group t,
- $d_e(t)$: Equals 1 iff the primary tree of coding group t passes through link e,
- $c_e(t)$: Equals 1 iff the protection tree of coding group t passes through link e,
- $p_v(t)$: A continuous variable between 0 and 1, resulting in an MIP formulation. It keeps the "voltage" value of node v in the protection tree of t. It is possible to set this variable as an integer larger than 0 but that makes the simulation run slower,
- $g_v(t)$: Same as $p_v(t)$ except it is used for the primary tree of t.

The objective function is to minimize the total cost incurred by primary and protection paths of each coding group

$$\min \sum_{e \in E} \sum_{t=1}^{T} a_e \times (d_e(t) + c_e(t)). \tag{4}$$

1) Coding Group Formation:

$$\sum_{t=1}^{T} n(i,t) = 1 \quad \forall i, \tag{5}$$

$$\sum_{i=1}^{N} n(i,t) \leq ND_d - 1 \quad \forall t. \tag{6}$$

The first constraint ensures that a connection can be routed and protected by only one diversity coding group. In a coding group with N connections, there are N link-disjoint paths inside the primary tree and at least 1 link-disjoint path as the protection tree. The required number of link-disjoint paths are at least $N+1$ for a coding group with size N. However, the nodal degree of the destination node is limited. Therefore, the maximum size of a coding group is limited by $ND_d - 1$.

2) Building Primary Trees:

$$\sum_{e \in \Gamma_o(v)} d_e(t) = \sum_{i=1, s_i=v}^{N} n(i,t) + \sum_{e \in \Gamma_i(v)} d_e(t) \quad \forall v \neq d, \forall t. \tag{7}$$

$$\sum_{e \in \Gamma_i(v), v=d} d_e(t) = \sum_{i=1}^{N} n(i,t) \quad \forall t, \tag{8}$$

$$\sum_{e \in \Gamma_o(v), v=d} d_e(t) = 0 \quad \forall t, \tag{9}$$

The constraints above define the structure of the primary trees depending on the node they traverse. A primary tree must have a link-disjoint path from each source node of the protected connections to the destination node. If node v is not a destination node, it can be an intermediate node or a source node or both. However, the behavior of the primary tree is the same on these nodes. There are two rules to consider while building primary trees on non-destination nodes. First, there must be a branch of the primary tree t on the outgoing links of a non-destination node for each protected connection originating from this node. Second, a non-destination node must forward the primary tree branches that are input using its outgoing links. In equation (7), on a non-destination node, the number of outgoing branches belonging to the primary tree t is equal to the number of incoming links carrying a branch of primary tree t plus the number of connections protected by coding group t that are originated at that node. Equation (7) satisfies both of the rules. If node v is a destination node, then the total number of incoming links carrying the primary tree of t must be equal to the number of connections protected by that coding group t as mathematically stated by equation (8). In equation (9), it is ensured that there is no primary tree on the outgoing links of the destination node since they are supposed to terminate at that node.

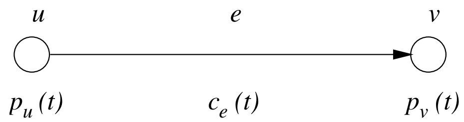


Fig. 2. A typical link in the protection tree t.

3) Building Protection Trees:

$$\sum_{e\in\Gamma_o(v)} c_e(t) \geq \frac{\sum\limits_{i=1,s_i=v}^{N} n(i,t)}{\beta} + \frac{\sum\limits_{e\in\Gamma_i(v)} c_e(t)}{\beta} \qquad \forall v \neq d, \forall t. \tag{10}$$

$$\sum_{e\in\Gamma_i(v),v=d} c_e(t) \geq \frac{\sum\limits_{i=1}^{N} n(i,t)}{\beta} \qquad \forall t, \tag{11}$$

$$\sum_{e\in\Gamma_o(v),v=d} c_e(t) \leq 0 \qquad \forall t. \tag{12}$$

The structure of the protection tree also depends on the nature of the node it is traversing over. Inequality (10) is similar to equation (7) except with one fundamental difference. Primary paths are link-disjoint and are not coded so they cannot merge. On the other hand, protection paths, which are link-disjoint to primary paths, are coded when they merge at the encoding nodes. Inequality (10) makes sure that if a node inputs one or more protection signals then it encodes those signals and uses at least one of its outgoing links to transmit the encoded signal to the destination node. The objective function makes sure that the encoded signals are transmitted over only a single outgoing link of the encoding node. The constraints regarding the destination node are different than the constraints regarding other nodes. In inequality (11), if coding group t protects at least one connection then there must be at least one branch of the protection tree of t carrying the protection signals on the incoming links of the destination node. In some cases, some of the branches of the protection tree, carrying different signals, may not merge and input to the destination node separately. This does not compromise the decoding structure. Inequality (12) ensures that there are no branches of the protection tree on the outgoing links of the destination node since they are supposed the terminate at that node.

4) Link-Disjointness:

$$d_e(t) + d_f(t) + c_e(t) + c_f(t) \leq 1 \qquad \forall e, f \in g, \forall g \in S, \forall t. \tag{13}$$

The primary tree and the protection tree must be link-disjoint which is satisfied by (13). The variables e and f are the links of span g in the opposite directions because a failure over this span affects both of these links at the same time.

The link-disjointness criterion between the primary paths of the connections in the same coding group is ensured implicitly while the MIP formulation builds the primary trees.

5) Cycle Prevention: In order to prevent getting cyclic (or loop) structures inside the trees, we choose to assign two "voltage" values to each node in the tree, as in [20], for the primary and protection trees, respectively. These are not actual voltage values, instead they are each a metric used as a variable in the formulation. We would like to emphasize that this voltage value is only used in the sense of a resemblance to the familiar Kirchoff's voltage law. It is an assigned variable to prevent loops.

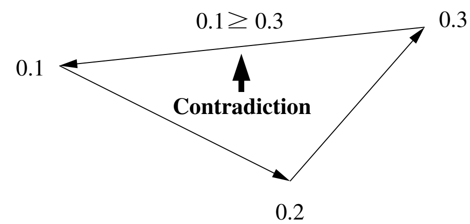


Fig. 3. Voltage value contradiction in a loop structure.

$$g_v(t) - g_u(t) \geq \alpha \cdot d_e(t) - (1 - d_e(t)) \qquad \forall e = u \to v, \forall t. \tag{14}$$

$$p_v(t) - p_u(t) \geq \alpha \cdot c_e(t) - (1 - c_e(t)) \qquad \forall e = u \to v, \forall t. \tag{15}$$

In inequalities (14) and (15), the voltage value at the head node should be higher than the voltage value at the tail node of the links which are part of the primary or protection trees, respectively. Fig. 2 shows a typical link in the network. The voltage value of node v must be higher than the voltage value of node u. This voltage relationship prevents the cyclic structures to be a part of the diversity coding trees such as in Fig. 3. These variables are crucial to ensure that the inequalities (7)-(12) produce valid primary and protection trees.

As an example, in Fig. 4(a), assume that there are 8 connections originated from S_is to D. The nodal degree of the destination node is 5 which means the maximum size of a coding group is 4. Therefore, the connections are partitioned into two different coding groups, the first group includes $S_1 - D$, $S_2 - D$, $S_3 - D$, and $S_4 - D$. The other group includes the rest of the connections.

In Fig 4(b), there is an example of the primary tree belonging to the first coding group formed in Fig. 4(a). In that example, there are four connections originating from nodes S_1, S_2, S_3, and S_4 to node D carrying signals a, b, c, and d , respectively. The primary tree of this coding group is depicted in Fig. 4(b) with thick straight arrows. As seen on node D, there are four branches of the primary tree on the incoming links of the destination node, each carrying the signal of a different connection. Nodes 1, 2, and 4 are pure source nodes of the connections protected by this coding group. It is seen that a new primary tree branch originates from each source node. Node 10 is both an intermediate and a source node. When equation (7) is applied to this node, there are two branches of primary tree on its outgoing links. One of the primary tree branches carry the signal of the connection $S_4 - D$ and the other branch is formed by forwarding the tree branch on the incoming link $4 \to 10$ to outgoing link $10 \to 11$. As seen in Fig. 4(b), equations (7)-(9) ensure that there is a link-disjoint primary path in the primary tree for each protected connection.

The protection tree of the example in Fig. 4 is depicted in Fig. 4(c). Nodes 1, 4, and 10 are pure source nodes whereas node 3 is a pure intermediate node and node 2 is both. There is only a single branch of protection tree over the incoming

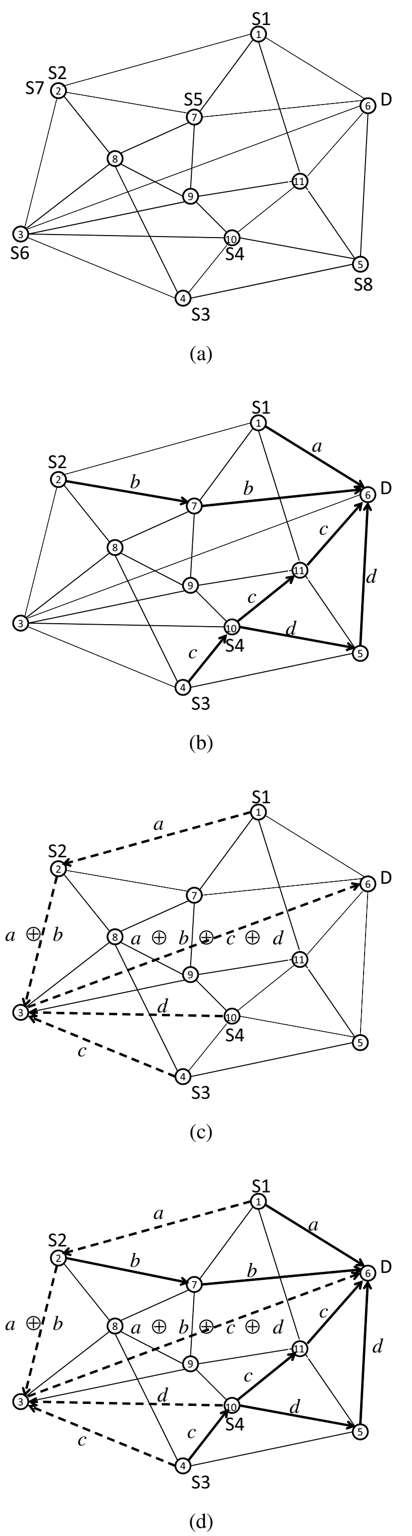


Fig. 4. A diversity coding tree example with 8 connection demands, (a) The source nodes of the 8 connections and the common destination node, (b) In the primary tree, on each non-destination node, the number of incoming signals is equal to that of outgoing signals, (c) In the protection tree, a non-destination node merges the incoming signal flows into a single outgoing link, (d) In a coding group, the primary and the protection trees are link-disjoint.

links of the destination node. At node 2, the signals a and b are coded since node 2 inputs two signals and merges them over a single branch of the protection tree. At node 3, three incoming branches of the protection tree are merged into a single outgoing branch. The signals of those branches are coded as well. As seen in Fig. 4(c), inequalities (10)-(12) ensure that the protection tree originates from the source nodes and its branches merge until they arrive at the destination (root) node of the tree.

In Fig. 4(d), it is shown that the primary and protection trees of a coding group are link-disjoint to ensure decodability.

6) Design Complexity Comparison: We leveraged the nature of the adopted diversity coding technique to simplify the design algorithm in two ways. First, the novel diversity coding tree algorithm requires dramatically fewer number of variables and constraints than the similar optimal algorithms of coding-based recovery techniques e.g., [17]. Second, the traffic matrix can be partitioned into smaller groups and each subgroup can be input to a parallel simulation without loss of optimality.

The diversity coding tree algorithm replaces the individual primary and protection paths with trees which leads to savings in the number of integer variables in the MIP formulation. Table I compares the complexities of different LP formulations of optimal coding-based algorithms. The techniques in [15], [17], and [18] are compared with the novel technique in terms of the total number of integer variables and constraints. We assume $T = |N|/2$.

In MIP, the complexity incurred by the continuous variables are negligible compared to that of the integer variables. As seen in Table I, the novel algorithm requires significantly fewer number of integer variables and constraints compared to the other techniques including the preceding work in [15]. It is also more scalable than the other techniques with larger network size and larger traffic matrix.

The adopted diversity coding structure implements coding on connections with the same destination node. In Fig. 5(a), there is an example of a typical traffic matrix of a network with 11 nodes. The rows are source nodes and the columns are destination nodes. The indices are the total connections between two nodes, which makes $|N| = 176$. However, in our algorithm, the connections with different destination nodes cannot interact with each other. Therefore, it is possible to partition that matrix into vectors of connections depending on their destination nodes. In Fig. 5(b), the set of connections with destination node equal to 1 are encircled, which makes $|N| = 12$ for this group. This observation leads to significant simplification of design complexity since $|N|$ has a significant effect on the complexity as shown in Table I. The whole traffic matrix can be partitioned into vectors which can be input to parallel simulations without loss of optimality. On the other hand, in general diversity coding, any connection can be encoded with other connections in the traffic matrix. Therefore, it is impossible to partition the traffic matrix into smaller groups without loss of optimality. The only insight is to partition connections which have closer destination nodes [4]. Even in that case, the size of the non-optimal partitions in general diversity coding cannot be as small as the size of the optimal partitions in the adopted diversity coding.

C. Non-systematic Coding

In this section, we present the optimal design algorithm for non-systematic diversity coding, where each connection has

TABLE I
COMPLEXITY COMPARISONS OF THE LP FORMULATIONS OF DIFFERENT TECHNIQUES

Technique	Number of integer variables	Number of constraints
Diversity Coding Tree	$\|N\|\|E\| + \|N\|^2/2$	$3\|N\|\|E\|/2 + \|N\|\|V\| + 7\|N\|/2$
Diversity Coding Tree in [15]	$3\|N\|\|E\|/2 + \|N\|^2/2$	$\|N\|^3\|E\|/2 + \|N\|^2\|E\|/2 + ...$
ILP formulation in [17]	$\|N\|^2/2(\|E\| + 1) + 3\|N\|\|E\|$	$\|N\|^4/8 + ...$
ILP formulation in [18]	$\|N\|\|E\|(\|V\| + 2) + \|N\|(\|N\| + 2\|V\|) + ...$	$\|N\|\|V\|(3\|E\| + \|N\| + \|V\|) + ...$

Nodes	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	1	3	1	1	1	1	1
2	1	0	10	3	2	8	5	4	1	1	3
3	1	10	0	5	2	11	5	7	1	1	6
4	1	3	5	0	3	9	2	2	1	1	6
5	1	2	2	3	0	9	1	1	1	1	3
6	3	8	11	9	9	0	8	6	2	2	11
7	1	5	5	2	1	8	0	4	1	1	3
8	1	4	6	2	1	6	4	0	1	1	6
9	1	1	1	1	1	2	1	1	0	1	1
10	1	1	1	1	1	2	1	1	1	0	1
11	1	3	6	6	3	3	11	3	1	1	0

(a)

Nodes	1	2	3	4	5	6	7	8	9	10	11
1	0	1	1	1	1	3	1	1	1	1	1
2	1	0	10	3	2	8	5	4	1	1	3
3	1	10	0	5	2	11	5	7	1	1	6
4	1	3	5	0	3	9	2	2	1	1	6
5	1	2	2	3	0	9	1	1	1	1	3
6	3	8	11	9	9	0	8	6	2	2	11
7	1	5	5	2	1	8	0	4	1	1	3
8	1	4	6	2	1	6	4	0	1	1	6
9	1	1	1	1	1	2	1	1	0	1	1
10	1	1	1	1	1	2	1	1	1	0	1
11	1	3	6	6	3	3	11	3	1	1	0

(b)

Fig. 5. (a) An exampler traffic matrix. The rows are source nodes and the columns are destination nodes. The indices are total number of unit-demand connections between these nodes, (b) The connections are partitioned depending on their destination node. The connections with destination node 1 are encircled as a subgroup and so on.

two paths and each path can be coded with others under some rules. 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. The paths that are to be coded together are assigned to the same subgroup of a coding group. The total number of subgroups varies between $N+1$ and $2N$. The number of paths in a subgroup takes values from zero to N. In the received vector of the destination node, each connection demand is represented as a variable and each subgroup is represented as an equation. 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 maximum $2N$ subgroups if each path is transmitted separately, which is the case in 1+1 APS. For example, in systematic diversity coding, there are a total of $N+1$ subgroups, N of them are the primary paths and one of them is the combination of protection paths. The common destination node carries out the decoding operation over the received vector.

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 signals 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 and $N+1$ subgroups are sufficient in systematic diversity coding.

In non-systematic coding, the paths in each subgroup must be specified. In [14], connection demands are randomly chosen and paths are assigned to subgroups of the existing coding group one by one. However, a general rule is needed to optimally build non-systematic codes. In [21], 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* can be paraphrased as

1) 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 can be followed from [21] by assuming U_s as the set of connection demand signals and L_s as the set of subgroups in a coding group.

We 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 exemplifies 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

$$\begin{bmatrix} a+b \\ b+c \\ c+d \end{bmatrix}, \tag{16}$$

which indicates 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. 6(a). In this map, there are two symbols for each connection demand, referring to their two link-disjoint paths. In Fig. 6(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 c, then connection demand a is indirectly related to connection demand c, which is shown with a dashed arrow in Fig. 6(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. 6(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. 6(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 indirectly related. The resulting non-systematic code will be valid as long as coding circles are prevented.

An ILP formulation is developed to implement the proposed technique with an objective to minimize total capacity on arbitrary networks. The ILP uses the same parameter as in Section II-B.

The variables related to non-systematic diversity coding problem are given below. All are binary and take the value

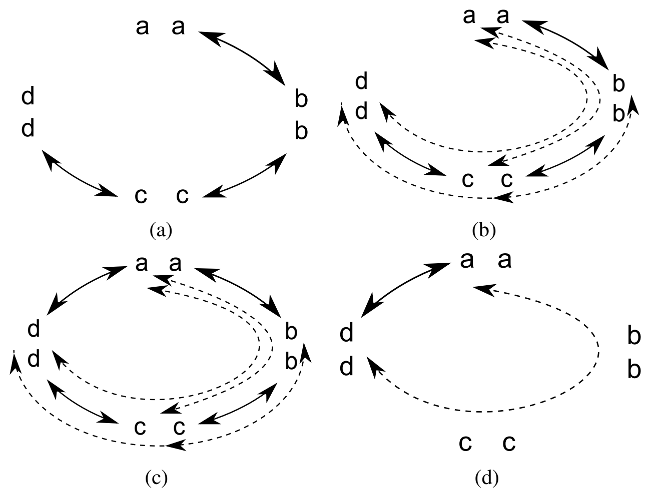


Fig. 6. Formation of a coding circle. A coding circle violates *Lemma 1*.

of 0 or 1.

- $x_e(i)$: Equals 1 iff the path i passes through link e,
- $n(i,t,s)$: Equals 1 iff path i is in subgroup s of coding group t,
- $m(i,j)$: Equals 1 iff path i and path j are coded together,
- $r(i,f)$: Equals 1 iff path i and connection demand f are indirectly related, 0 otherwise,
- $\theta_e(t,s)$: Equals 1 iff the topology of subgroup s of coding group t includes link e.

Note that $1 \leq i,j,s \leq 2N$, $1 \leq f \leq N$, and $1 \leq t \leq T$. The objective function is

$$\min \sum_{t=1}^{T} \sum_{s=1}^{2N} \sum_{e \in E} a_e \times \theta_e(t,s). \tag{17}$$

The following inequality finds two paths for each connection demand

$$\sum_{e \in \Gamma_i(v)} x_e(i) - \sum_{e \in \Gamma_o(v)} x_e(i) = \begin{cases} \text{-1} & \text{if } v = s_i, \\ 1 & \text{if } v = d, \\ 0 & \text{otherwise.} \end{cases} \tag{18}$$

Note that we require $\mathrm{mod}(i,2) = 0 \Rightarrow s_i = s_{i-1}$ for $1 \leq i \leq 2N$.

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

$$\sum_{t=1}^{T} \sum_{s=1}^{2N} n(i,t,s) = 1 \qquad \forall i, \tag{19}$$

$$n(i,t,s) + n(i-1,t,s) \leq 1 \qquad \forall i,s,t : \mathrm{mod}\,(i,2) = 0, \tag{20}$$

$$\sum_{s=1}^{2N} n(i,t,s) = \sum_{s=1}^{2N} n(i-1,t,s) \qquad \forall i,t : \mathrm{mod}(i,2) = 0, \tag{21}$$

$$m(i,j) \geq n(i,t,s) + n(j,t,s) - 1 \qquad \forall i \neq j, s, t. \tag{22}$$

Inequality (20) ensures that paths of the same connection cannot be in the same subgroup. However, equation (21) ensures that they must be in the same coding group. If two paths are in the same subgroup then they are assumed to be coded together, which is satisfied by inequality (22).

$$r(i,f) \geq m(i,j) + m(j^*,2f) + m(j^*,2f-1)$$
$$-m(i,2f) - m(i,2f-1) - 1 \qquad \forall i,j,f, : i \neq j \tag{23}$$

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

$$r(i,f) \geq r(i,g) + m(2g,2f) + m(2g,2f-1)$$
$$+m(2g-1,2f) + m(2g-1,2f-1) - 1 \;\; \forall i, f \neq g :$$
$$i \neq 2f, i \neq 2f-1, i \neq 2g, i \neq 2g-1. \tag{24}$$

In inequality (23), if path i becomes indirectly related to demand f 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 (24) ensures that path i becomes related to demand f if path i is related to demand g and one of the paths carrying demand g is coded with one of the paths carrying demand f. Inequality (25) 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 the coding circles.

$$r(2f,g) + r(2f-1,g) + m(2f,2g) + m(2f-1,2g)$$
$$+m(2f,2g-1) + m(2f-1,2g-1) \leq 1 \;\; \forall g, f : g \neq f, \tag{25}$$

$$\theta_e(t,s) \geq x_e(i) + n(i,t,s) - 1 \quad \forall e,i,s,t \tag{26}$$

$$\theta_e(t,s_1) + \theta_e(t,s_2) + \theta_f(t,s_1) + \theta_f(t,s_2)$$
$$\leq 1 \qquad \forall e, f \in g, \forall g \in S, \forall t, s_1, s_2 \tag{27}$$

Inequality (26) finds the topologies of the subgroups. The topology of a subgroup is the union of the protection paths of the connections in that subgroup. Inequality (27) ensures that two subgroups have link-disjoint topologies.

In Section II-D and III-D, we will present our simulation results. Although our techniques are applicable to electrical and optical networks, in our simulations we focused on optical networks.

D. Pre-Provisioning Results

In this section, we present the simulation results of systematic and non-systematic diversity coding on test networks. Those results are compared with the simulation results of SPP, *p*-cycle protection, SONET rings, and $1+1$ APS in terms of capacity efficiency and restoration speed. Capacity efficiency is measured as the total capacity to route and protect connection demands and restoration speed is measured as the worst-case restoration time.

We have three test networks to analyze the comparative performance of these five techniques. These networks are the COST 239 network [3], the NSFNET network [22] and the Smallnet network [20]. Their topologies are given in Fig. 7(a), Fig. 7(b) and Fig. 7(c), respectively. In Fig. 7(a)-7(c), the numbers next to the nodes are node indices, whereas in Fig. 7(a)-7(b), the numbers next to the links are costs (lengths) of using that link. In the Smallnet network, link lengths are set to 100 kilometers. The traffic matrix of the COST 239 network is taken from [20].The COST 239 network characterizes the network between major European metropolises. The traffic matrix of the NSFNET network consists of 300 random unit-sized demands, which are chosen using a realistic

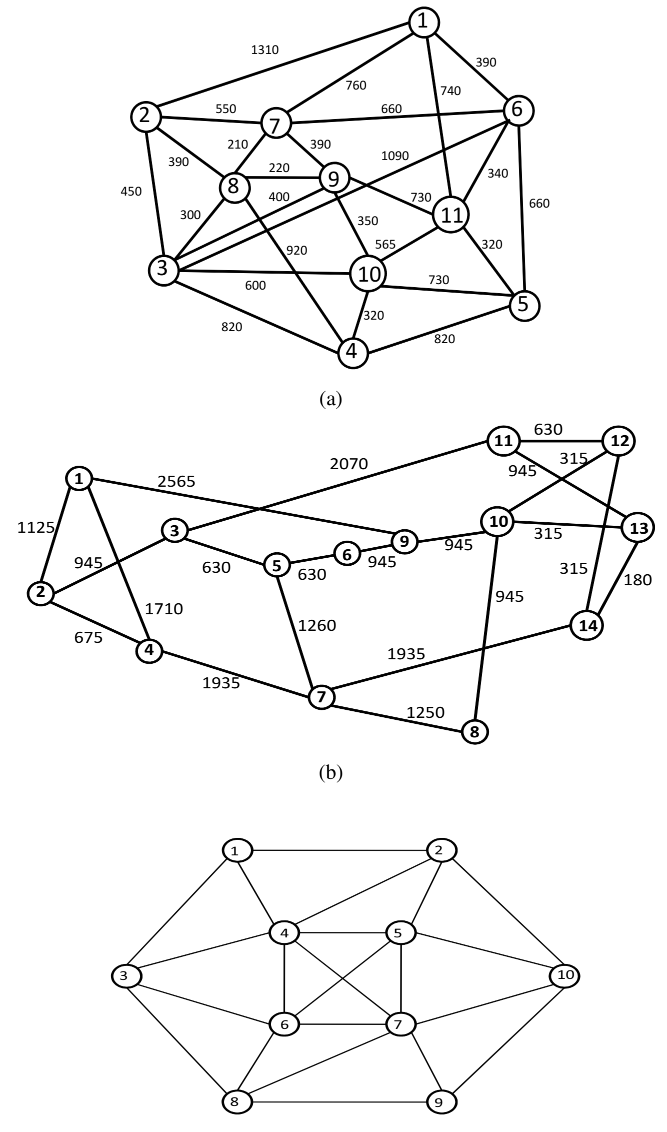


Fig. 7. Test networks, (a) COST 239 network, (b) NSFNET network, (c) Smallnet network.

gravity-based model [23]. Each node in the NSFNET network represents a U.S. metropolitan area and their population is proportional to the weight of each node in the connection demand selection process. The traffic matrix of the Smallnet network consists of randomly selected 250 random unit-sized demands according to the gravity-based model. It is assumed that node 3 and node 10 have 5 units of population and rest of the nodes have 1 unit population to highlight the comparative performance of diversity coding in a traffic scenario similar to U.S. coast-to-coast traffic.

For p-cycle protection, optimal cycle-exclusion based ILP for joint capacity placement (JCP) and spare capacity placement (SCP) algorithms from [20] are employed depending on the network scenario. The SCP design algorithm for SPP is taken from [3, p. 406]. The algorithm of SPP is suboptimal due to the extremely high complexity of the optimal ILP formulation of SPP. The design algorithm for SONET 4-fiber bidirectional line switched rings is developed using the design algorithm of p-cycle protection. The design algorithm for $1+1$ APS is simply finding the shortest pair of disjoint primary and protection paths for each connection. It can be implemented using different, relatively simple design algorithms. The design algorithms for SPP, p-cycle protection, and SONET rings are developed with the assumption of symmetric traffic. Diversity coding tree and $1+1$ APS treat nonsymmetric traffic the same as the symmetric traffic. Therefore, they can maintain their performance under nonsymmetric traffic whereas the other techniques are expected to behave worse.

In COST 239 and Smallnet networks, JCP is carried out with the exception of the design algorithm of SPP. In the NSFNET network, since both the network size and the traffic matrix are relatively large, SCP is carried out instead of JCP for all. It will become clear from simulation results that this change does not affect our overall conclusion in this paper. In SCP, the primary paths are routed using the shortest paths and the algorithms minimize the total spare capacity. The traffic matrix is partitioned into smallest number of groups when the computational resources are not able to compute the problem for the whole traffic matrix at once. Simulation results for the COST 239 network, the Smallnet network, and the NSFNET network are given in Table II, Table III, and Table IV, respectively. TC means total capacity to route and protect the traffic and RT is the worst-case restoration time. The restoration times of systematic and non-systematic diversity coding are simply

$$RT_{SDC} = F+M+S, \quad RT_{N-SDC} = 2\times F+2\times M+S+T,$$

where F is the failure detection time, M is the node processing time, and S is the node switching time. The maximum values for F, M, and S are taken as 10μs [16]. RT_{N-SDC} is larger than RT_{SDC} due to added complexity in the decoding structure. In non-systematic coding, assume two primary paths carrying data a and b are encoded and $a+b$ is formed. If the link carrying data a fails, the receiver will still treat the encoded data as $a+b$, not $0+b$, unless the intermediate node detects this failure and issues a feedforward error signal. In order to guarantee correct decoding, there is an extra detection and node processing operation in the intermediate node so that the receiver can adjust its decoding structure. T represents the transmission time of the feedforward signal in non-systematic diversity coding, which is also taken as 10μs. There is also additional detection and node processing times in RT_{N-SDC}. In reality, these values depend on the nature of the network, the adopted network protocol, and the transmission technology. However, in today's networks, these numbers are in the sub-ms range. The propagation delay difference in the restoration time formulation of diversity coding is eliminated because of the buffers, which equalize the arrival time to the destination node of each path in the same coding group. The restoration time formulation of SPP is taken from [24] as

$$RT_{\text{SPP}} = F + 2\times d_{sd} + (h_{si}+1)\times M + X + (h_b+1)\times M.$$

The symbol X refers to the configuration time of an optical cross-connect (OXC), h_b is the number of hops in the protection path between destination node d and source node s, h_{is} is number of nodes between node i, which detects the failure, and

TABLE II
SIMULATION RESULTS OF COST 239 NETWORK

COST 239 Network, 11 nodes, 26 spans					
Scheme	TC	RT for different X values (ms)			
		0.5ms	1ms	5ms	10ms
N-S. Div. Cod.	217560	$2 \times F + 2 \times M + S + T$ (60μs)			
S. Div. Cod.	226200	$F + M + S$ (30μs)			
$1 + 1$ APS	355530	$F + S$ (20μs)			
SPP	216630	25.11	25.61	29.61	34.61
p-cycle	215190	31.26	31.76	35.76	40.76
SONET	288140	26.2	26.7	30.7	35.7

TABLE III
SIMULATION RESULTS OF SMALLNET NETWORK

Smallnet Network, 10 nodes, 22 spans					
Scheme	TC	RT for different X values (ms)			
		0.5ms	1ms	5ms	10ms
N-S. Div. Cod.	77900	$2 \times F + 2 \times M + S + T$ (60μs)			
S. Div. Cod.	78000	$F + M + S$ (30μs)			
$1 + 1$ APS	103700	$F + S$ (20μs)			
SPP	65000	5.59	6.09	10.09	15.09
p-cycle	67300	5.11	5.61	9.61	14.61
SONET	92600	4.6	5.1	9.1	14.1

TABLE IV
SIMULATION RESULTS OF NSFNET NETWORK

NSFNET Network, 14 nodes, 21 spans					
Scheme	TC	RT for different X values (ms)			
		0.5ms	1ms	5ms	10ms
N-S. Div. Cod.	1562880	$2 \times F + 2 \times M + S + T$ (60μs)			
S. Div. Cod.	1581570	$F + M + S$ (30μs)			
$1 + 1$ APS	1881880	$F + S$ (20μs)			
SPP	1264865	82.87	83.37	87.37	92.37
p-cycle	1440435	74.41	74.91	78.91	83.91
SONET	1595840	68.74	69.24	73.24	78.24

node s. Finally, d_{sd} represents the propagation delay between node s and node d. It is optimistically assumed that the OXC configurations over the protection path can be carried out simultaneously, which is opposed by some researchers in [8], [10], and [13]. The restoration time formulation of p-cycle protection is taken from [16], which is

$$RT_{p-\text{cycle}} = F + X + h \times M + d,$$

where the parameter d is the longest propagation delay between any two nodes in a p-cycle and h is the number of nodes in a cycle. The restoration time formulation of SONET rings is the same as p-cycle protection except that, in simulations, the longest rings are usually shorter than longest p-cycles resulting in shorter restoration time. In $1{+}1$ APS, restoration is basically detecting the failure and switching the traffic from the primary path to the protection path assuming they are synchronized. Then,

$$RT_{1+1} = F + S.$$

As stated earlier, we conservatively assume that F, M, and S have values about 10μs each. This makes RT approximately 30μs for systematic diversity coding, 60μs for non-systematic diversity coding and 20μs for $1{+}1$ APS in Tables II-IV, where "S." and "N-S." mean systematic and non-systematic, respectively. It is noted that the restoration time results are evaluated based on the assumptions above and the simulation results. In this paper, CPLEX 12.2 is used to run LP formulations.

As seen from the results, diversity coding is much faster than both SPP and the p-cycle technique in each network in each configuration. In all networks, non-systematic diversity coding is more capacity efficient than the systematic version as expected. However, systematic diversity coding is faster than the non-systematic version. For pre-provisioning of the static traffic, they are less capacity efficient than SPP in each network. The restoration speed increases more than hundred times over SPP. Therefore, our algorithms offer a tradeoff for network designers where the speed increase is at least two orders of magnitude at the expense of less than 26% extra capacity. We believe with the existence of abundant fiber on today's networks, our techniques offer a desirable tradeoff. The restoration time of SPP increases as the expected time of OXC configuration increases. Realistically, in some cases OXC configuration may take seconds [9]. It should be noted that, if the nodes in the backup paths are not able to carry out dynamic OXC configurations simultaneously, then the restoration time of SPP increases significantly. The p-cycle technique also results in higher capacity efficiency than diversity coding in each network. The capacity efficiency of the diversity coding gets closer to the capacity efficiencies of the p-cycle technique and the SPP, when the JCP simulations are carried out in the COST 239 network. On the other hand, diversity coding is also more than hundred times faster than the p-cycle technique. It is important to realize that, the restoration time of the p-cycle technique may increase if a link is protected via multiple p-cycles. In this case, end nodes of the failed link have to configure multiple OXCs simultaneously. Some nodes may not be able to carry out these configurations in parallel. Therefore, restoration time of p-cycles can significantly increase in some cases. Furthermore, it is observed that capacity efficiency of the p-cycle technique vanishes while going towards more sparse networks.

On the other hand, both versions of diversity coding are significantly more capacity efficient than $1 + 1$ APS and SONET rings in each network. They are also much faster than the SONET rings, which is slightly faster than the p-cycle technique. SONET rings can offer less than 50 ms restoration time if there is a length limit over the candidate rings with the expense of higher total capacity. The fact that $1 + 1$ APS is faster than diversity coding becomes negligible considering the extra delay coming from imperfections inside the optical network.

In diversity coding, the maximum synchronization delay of the primary paths in the COST 239 network is 8.17 ms. This value is equal to 22.27 ms and 2 ms for the NSFNET and the Smallnet networks, respectively.

As a breakdown of the diversity coding results, the capacity efficiency is investigated depending on each destination node for three networks in Table V. The spare capacity percentage

(SCaP) is calculated as

$$SCaP = \frac{Total\ Capacity - Shortest\ Working\ Capacity}{Shortest\ Working\ Capacity}.$$

$Shortest\ Working\ Capacity$ is the total capacity when there is no protection and the traffic is routed over the shortest paths.

As it is seen from the results, diversity coding results in lower SCaP when the destination node has a higher nodal degree or it is closer to the edge of the network. The SCaP results of non-systematic diversity coding are lower than or equal to that of systematic diversity coding as expected.

III. Dynamic Provisioning

The static traffic assumptions are not always valid. In some cases, the future network demands cannot be known or predicted. The connections can be set up by the bandwidth-on-demand paradigm. Therefore, there is a need for a dynamic provisioning technique that will dynamically provision the new connection demands without any future knowledge of the traffic. To our knowledge, this paper presents the first coding-based dynamic provisioning technique except its preceding works in [15] and [25].

Some of the challenges in the dynamic provisioning problem are the tight timing constraints, the lack of knowledge about the future traffic and preservation of the integrity of the existing connections. In this paper, those challenges are mitigated with a simple and optimal design algorithm. The simplicity of the design algorithm is due to the fact that provisioning of each connection is done one-by-one instead of optimizing the whole set of connections at once. One-by-one provisioning of connection demands was used for coding in [14] and [26] as heuristic algorithms for static traffic. The optimality of the design algorithm depends on the assumptions listed as

1. The existing connections cannot be rearranged due to QoS requirements,
2. At the beginning, the demand matrix is an empty set,
3. Centralized information about the state of the network is updated and conveyed to the nodes every time there is a change,
4. Every node is able to run the algorithm and calculate the routes,
5. Connections can be set up when demands appear and terminated when they no longer exist,
6. The objective function is to minimize the total cost.

The adopted diversity coding schemes are the same as the previous section. We developed ILP formulations for both systematic and non-systematic diversity coding with single destination node. These schemes result in lower complexity, lower restoration time, lower signaling, and higher coding flexibility. A novelty of this paper is to show that non-systematic coding can be implemented optimally without adding any extra complexity to that of systematic coding. The only difference between the two schemes is the way the cost parameters are defined in each ILP formulation.

In this section, we will first present the algorithm for non-systematic diversity coding. Later, systematic diversity coding will be explained as a special case of non-systematic coding.

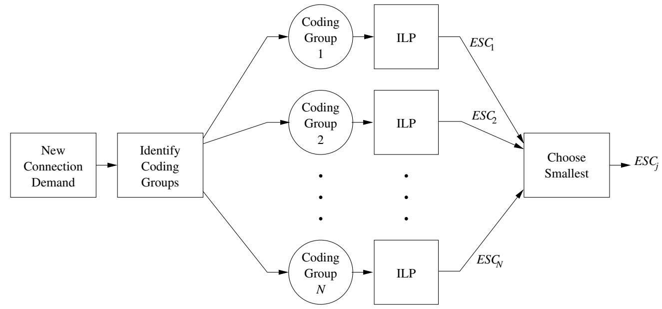


Fig. 8. Extra spare capacity is calculated for each coding scenario and the minimum is chosen.

The dynamic provisioning algorithm is based on adding new connection demands to the established coding groups such that the new coding group preserves the decodability under any single link failure. When a new demand arrives, its destination node is detected. Then, the existing coding groups which share the same destination node with the new connection are listed. One of the coding groups is an empty set which allows the new demand to establish a new coding group. After that, the new connection is hypothetically added to each listed coding group. We developed an ILP formulation that does the adding operations with minimum extra capacity. It finds link-disjoint primary and protection paths for the new demand depending on each coding scenario. Finally, the cost of each addition scenario is evaluated and the new connection demand is added to the group requiring the lowest extra capacity. The algorithm is depicted in Fig. 8. In this figure, ESC_i means total extra capacity required to add a new connection demand to coding group i. The cost vector of the links is adjusted for every coding scenario, depending on the topology of the coding group. Once the best available coding group is selected, the coding group topology is updated with the new connection and the algorithm is ready to incorporate a new connection demand.

Before attempting to optimize the extra cost, the decodability of the coding groups must be preserved due to the addition of a new demand. In our algorithm, we defined a set of rules to preserve the validity of coding groups after adding a new connection demand. These rules are

1. One of the paths of the new connection must be link-disjoint to any path in the coding group,
2. The other path of the new connection must be coded with only one path in the coding group,
3. No path in the coding group can diverge after any node.

The decodability of the augmented coding groups can be proven using induction if the rules above are followed. In non-systematic diversity coding, the received vector at the destination node of a coding group looks like in Fig. 9(a), where a, b, c, and d are the coded signals and x_{ij} are the binary coding parameters, which can take value of 0 or 1. In reality, there may be more than $N+1$ incoming paths but some of the paths can be encoded and merged just before they enter the destination node so that the structure above is established. It is assumed that the signals can be extracted from this coding vector under any single link failure scenario. Therefore, the received matrix $\mathbf{X} = [x_{ij}]_{5\times 4}$ is full rank even if any one of its rows is deleted. Deleting a row is the analog

TABLE V
SCaP RESULTS FOR EACH DESTINATION NODE

Dest. Node	Non-systematic Diversity Coding			Systematic Diversity Coding		
	Smallnet	COST 239	NSFNET	Smallnet	COST 239	NSFNET
Node 1	107.4	74.6	87.6	107.4	80.8	87.6
Node 2	76.0	71.4	109.1	76.0	74.8	109.1
Node 3	54.0	65.3	146.0	54.0	69.2	146.0
Node 4	61.9	79.3	81.2	61.9	81.5	81.2
Node 5	82.6	68.6	122.8	82.6	75.8	122.8
Node 6	82.3	66.6	145.4	82.3	77.9	145.4
Node 7	63.3	88.3	98.7	66.6	95.3	109.1
Node 8	94.7	96.4	194.4	94.7	108.2	194.4
Node 9	114.2	83.9	209.4	114.2	88.2	209.4
Node 10	55.7	101.3	85.4	55.7	111.6	108.4
Node 11	NA	77.6	117.3	NA	88.0	117.3
Node 12	NA	NA	96.5	NA	NA	96.5
Node 13	NA	NA	89.9	NA	NA	89.9
Node 14	NA	NA	102.9	NA	NA	102.9
Average	68.6	77.4	104.3	68.8	84.5	106.7

$$\begin{bmatrix} x_{11}a+x_{12}b+x_{13}c+x_{14}d \\ x_{21}a+x_{22}b+x_{23}c+x_{24}d \\ x_{31}a+x_{32}b+x_{33}c+x_{34}d \\ x_{41}a+x_{42}b+x_{43}c+x_{44}d \\ x_{51}a+x_{52}b+x_{53}c+x_{54}d \end{bmatrix}$$

(a)

$$\begin{bmatrix} x_{11}a+x_{12}b+x_{13}c+x_{14}d+0e \\ x_{21}a+x_{22}b+x_{23}c+x_{24}d+0e \\ x_{31}a+x_{32}b+x_{33}c+x_{34}d+0e \\ x_{41}a+x_{42}b+x_{43}c+x_{44}d+0e \\ x_{51}a+x_{52}b+x_{53}c+x_{54}d+e \\ 0a+0b+0c+0d+e \end{bmatrix}$$

(b)

$$\begin{bmatrix} 0a+0b+0c+0d+0e \\ x_{21}a+x_{22}b+x_{23}c+x_{24}d+0e \\ x_{31}a+x_{32}b+x_{33}c+x_{34}d+0e \\ x_{41}a+x_{42}b+x_{43}c+x_{44}d+0e \\ x_{51}a+x_{52}b+x_{53}c+x_{54}d+e \\ 0a+0b+0c+0d+e \end{bmatrix}$$

(c)

$$\begin{bmatrix} x_{11}a+x_{12}b+x_{13}c+x_{14}d+0e \\ x_{21}a+x_{22}b+x_{23}c+x_{24}d+0e \\ x_{31}a+x_{32}b+x_{33}c+x_{34}d+0e \\ x_{41}a+x_{42}b+x_{43}c+x_{44}d+0e \\ x_{51}a+x_{52}b+x_{53}c+x_{54}d+e \\ 0a+0b+0c+0d+0e \end{bmatrix}$$

(d)

Fig. 9. The received vector of a nonsystematic diversity coding, (a) $\mathbf{X} = [x_{ij}]_{5\times 4}$ is a "full rank + 1" matrix, (b) Addition of a new connection demand carrying e, (c) Signals are decodable if the first path fails, (d) Signals are decodable if the sixth path fails.

of a single link failure whereas the full rank property assures the decodability of the signals. We call these matrices to have "full rank + 1" property. The goal is to preserve the "full rank + 1" property of the augmented coding matrices after adding a new demand. Assume that we have a new connection demand that shares the same destination node with the connections in matrix $\mathbf{X}$, carrying signal e. Following the rules, the received matrix can be transformed to the format in Fig. 9(b). The link disjoint path of the new connection is represented as the sixth row of the new coding matrix. The other copy of signal e is coded with other signals in the coding group as arbitrarily denoted in the fifth row. In reality, the decision to choose the coding row is made by the ILP formulation depending on the underlying topology with an objective of minimum extra capacity. For decodability, the new signal should be coded with at most one path in the coding group. The "full rank + 1" property of the new coding matrix can be shown by checking the decodability of the new coding vector under any single link failure. For the no link failure case, we can derive a, b, c, and d by solving first four rows and we can derive e using the last row. If we delete one of the first four rows, say the first row, then the received matrix is depicted in Fig. 9(c). We derive e from the last row and subtract it from the fifth row. Then the matrix generated from rows 2 to 5 that multiplies the vector $(a, b, c, d)^T$ has full rank. Therefore, all of a, b, c, and d can be extracted. In Fig. 9(d), the sixth row is deleted. Then, the first four rows can be used to extract signals a, b, c, and d. These signals can then be used to find the value of e from the fifth row.

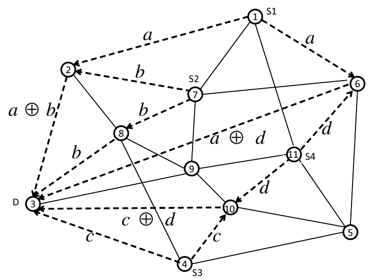
To preserve the "full rank + 1" property of the coding matrix, none of the paths is allowed to diverge after any node. If a new connection is merged with a path in the existing coding group, they will stay together until the destination node is reached. Otherwise, a path may span multiple rows in the coding matrix, which may impair decodability of the new coding group.

If the new connection demand cannot be added to any of the possible coding groups following the mentioned rules, then a new coding group is established by the new connection demand itself. The new coding group will consist of two link-disjoint paths belonging to the new connection. The received vector of the new group becomes

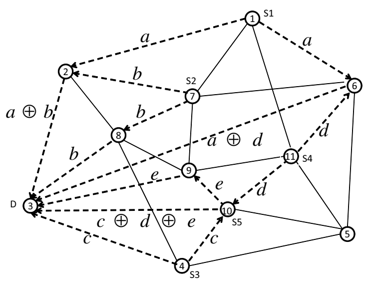
$$\begin{bmatrix} f \\ f \end{bmatrix} \tag{28}$$

where f is the signal carried by the new connection. The proof of decodability of coding groups via induction is completed since the initial state of coding matrix $\mathbf{X}$ satisfies the "full rank + 1" property.

Coding matrices with "full rank + 1" property can be built with coefficients other than 0 and 1. However, we opt to choose this approach because of its simplicity and the fact that it should result in minimum overall capacity.



(a)



(b)

Fig. 10. Addition of a new connection to an existing coding group (a) The existing coding group, (b) The new augmented coding group after the addition.

An example is provided to highlight how a new connection demand is added to an existing coding group. The coding group is shown in Fig. 10(a). There are four connection demands from $S1$, $S2$, $S3$, and $S4$ to D whose signals are represented as a, b, c, and d, respectively. The dashed links belong to the coding group. There is a new connection demand from $S5$ to D to be added to the existing coding group, denoted by e. As it is seen from Fig. 10(a), there exists a path from $S5$ to D over $10 \rightarrow 9 \rightarrow 3$, which is link-disjoint to the coding group topology. The other copy of the new signal will be coded with one of the paths of the existing coding group. After the addition algorithm, the new augmented coding group topology is shown in Fig. 10(b). As it is seen, the secondary path of the new connection demand incurs no extra cost since it is coded over the already established portions of the coding group. The transformation in the $\mathbf{X}$ matrix is depicted in Fig. 11(a). The dashed rectangle shows the coding matrix before addition. The corresponding transformation of the received coding vector is shown in Fig. 11(b). The $\mathbf{X}$ matrix preserves the "full rank + 1" property after the transformation.

Systematic diversity coding is a special case of the non-systematic diversity coding. In this case, there is a specific distinction between primary and protection paths of each connection demand. When a new connection demand arrives, its protection path can only be encoded with the protection paths of other connections in the existing coding group. Therefore, we need to redefine the second rule of addition as

2. The other (protection) path of the new connection must be coded with only protection paths in the coding group.

$$X = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \longrightarrow \begin{bmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

(a)

$$\begin{bmatrix} a \oplus b \\ b \\ a \oplus d \\ c \oplus d \\ c \end{bmatrix} \longrightarrow \begin{bmatrix} a \oplus b \\ b \\ a \oplus d \\ c \oplus d \oplus e \\ c \\ e \end{bmatrix}$$

(b)

Fig. 11. Transformation of the $\mathbf{X}$ matrix after adding a new connection demand.

The received vector of the a coding group looks like

$$\left[\begin{array}{ccccc} a+b+c+d & a & b & c & d \end{array} \right]^T \tag{29}$$

where the encoded paths span only a single row. The first row is the union of protection paths and the rows $2-5$ are primary paths. Assume a new connection demand is to be added to this coding group. The primary path of the new demand has to be link-disjoint to the paths in the coding group. The protection path of it can only be encoded with other protection paths placed in the first row of the coding vector. The augmented coding vector looks like

$$\left[\begin{array}{cccccc} a+b+c+d+e & a & b & c & d & e \end{array} \right]^T \tag{30}$$

We developed an integer linear programming (ILP) based algorithm to establish and enlarge coding groups as new demands arrive. The algorithm is optimal given the assumptions mentioned above. It maps both diversity coding structures into arbitrary topologies in order to protect the connections against single link failures in a cost efficient way. The ILP core of the algorithm inputs the new demand and a coding group. It searches for possible routing and coding scenarios with lowest cost. The cost of the links are adjusted regarding the topology of the coding group. When the new demand is coded over an existing link of the coding group, it incurs no extra capacity. The ILP formulation leverages the underlying coding group topology to find a pair of paths for the new connection at lowest extra cost.

The parameters of the ILP formulation to find a pair of link disjoint primary and secondary paths are as follows.

- $G(V, E)$: Network graph,
- S : The set of spans in the network, a span consists of two links in the opposite directions,
- N : Enumerated list of all connections,
- ${a_e}^1$: Cost associated with link e for the primary path, same for both non-systematic and systematic diversity coding,
- ${a_e}^2$: Cost associated with link e for the secondary path, depends on the nature of diversity coding,

- $\Gamma_i(v)$: The set of incoming links of each node v,
- $\Gamma_o(v)$: The set of outgoing links of each node v.

The binary ILP variables which take values 0 or 1 are

- x_e : Equals 1 iff the primary path of the new connection passes through link e,
- y_e : Equals 1 iff the secondary path of the new connection passes through link e.

The objective function is

$$\min \sum_{e \in E} x_e \cdot {a_e}^1 + y_e \cdot {a_e}^2. \quad (31)$$

$$\sum_{e\in\Gamma_i(v)} x_e - \sum_{e\in\Gamma_o(v)} x_e = \sum_{e\in\Gamma_i(v)} y_e - \sum_{e\in\Gamma_o(v)} y_e = \begin{cases} \text{-1} & \text{if } v = s, \\ 1 & \text{if } v = d, \\ 0 & \text{otherwise}, \end{cases} \quad \forall v, \quad (32)$$

$$x_e + x_f + y_e + y_f \leq 1 \quad \forall e, f \in g, g \in S. \quad (33)$$

The origination, flow, and termination of the primary path (x_e) and the secondary path (y_e) are determined by equation (32), where s and d are the source and destination nodes of the new connection, respectively. The link disjointness between the primary and secondary paths is satisfied by inequality (33).

A. Cost Adjustment Example

In Fig. 12(a), there are two unidirectional links between each node and the numbers next to them are their costs. As an example, we have a coding group that is shown in Fig. 12(b). There are two source nodes $S1$ and $S2$ and one destination node D. The signals transmitted from $S1$ and $S2$ are a and b respectively. The third path also carries the coded version of these signals to the destination node. The dashed lines show links that are incorporated in the coding group. Assume that we have a new connection request from any arbitrary node (except node 3) to node 3. We want to calculate the required spare capacity to add this connection to the coding group. When we run the ILP formulation, there will be two different topologies in terms of the costs of the links. The topology for the primary path differs from the topology for the secondary path depending on the existing coding group. The topology for the primary path in both non-systematic and systematic coding are the same. However, the topology for the protection path depends on the coding scenario. We consider the existing coding group shown in Fig. 12(b). Consider rules 1-3 described earlier. This fact is visualized in Fig. 12(c) and Fig. 12(d) for the primary and secondary paths of non-systematic coding, respectively. Following rule 1, in Fig. 12(c), the links which carry the signals in the existing coding group are removed from the network. Following rule 2, in Fig. 12(d), the links which belong to the existing coding groups have zero cost. On the other hand, the cost of the links for the secondary path of systematic coding is depicted in Fig. 12(e) following the modified rule 2. Only the links belonging to secondary paths of the existing coding group have zero cost. Rest of the links in that coding group are removed. The links which are not associated with the coding group have the same regular cost for both of the paths as shown in Fig. 12(c), 12(d), and 12(e).

B. Limited Capacity Case

We assumed that there is no limit on the capacity of the links. Therefore, any new demand can be routed and protected. In the opposite case, some of the new demands may not have sufficient capacity over the trail of their candidate paths. In limited capacity case, the links which have zero capacity are removed from the network that is input to the ILP formulation. If a new demand cannot be routed and protected within the existing topology, then it is blocked.

C. Connection Teardown

We assume that connections leave the network after a duration. The teardown process of these leaving connections are

- First, the connection is dropped from its coding group and the topology of that coding group is updated. This is done by subtracting the links which purely carry the signal of interest from that coding group topology. The links that carry the coded version of this signal are kept in the coding group topology.
- In the second step, the received matrix of the coding group is updated by excluding the signal associated with the leaving connection.
- In the last step, the capacity of the links that are subtracted from the respective coding group are increased by 1. It should be noted that the last step is required only if the capacity of the links are limited.

D. Dynamic Provisioning Results

In this section, the performance of the ILP-based dynamic provisioning algorithm is compared against a *p*-cycle protection algorithm given in [27], the optimal $1+1$ APS algorithm, and an ILP-based algorithm for diverse routing given in [28], which is another form of SPP. Both systematic and non-systematic coding are employed. Unlike the simulation setup in [15], the capacities of links are not limited. We present the simulation results on test networks consisting of worst-case restoration time and total capacity required to route and protect connections. The same NSFNET and Smallnet networks are used for simulations. In [27], the Most-Free-Routing algorithm is chosen for *p*-cycle protection among similar techniques due to its superior performance. In the simulations, connection demands are provisioned randomly one-by-one from a finite traffic demand matrix, without any future information about the connection demands. The traffic matrices are the same as in the pre-provisioning simulations. The objective is to minimize total capacity without changing the paths of the existing connection demands. The recovery time formulations of the techniques are the same as the case in static pre-provisioning.

The comparative results in terms of worst-case restoration time and total capacity are presented in Table VI and Table VII. Evaluation of the simulation results of the dynamic provisioning confirms that the restoration time and capacity efficiency analysis of different techniques do not change much with the design algorithm and the nature of traffic. It is because the nature of these techniques is preserved in both static and dynamic provisioning. Therefore, both versions of diversity coding are faster than *p*-cycle protection and diverse routing in dynamic provisioning by three orders of magnitude. The

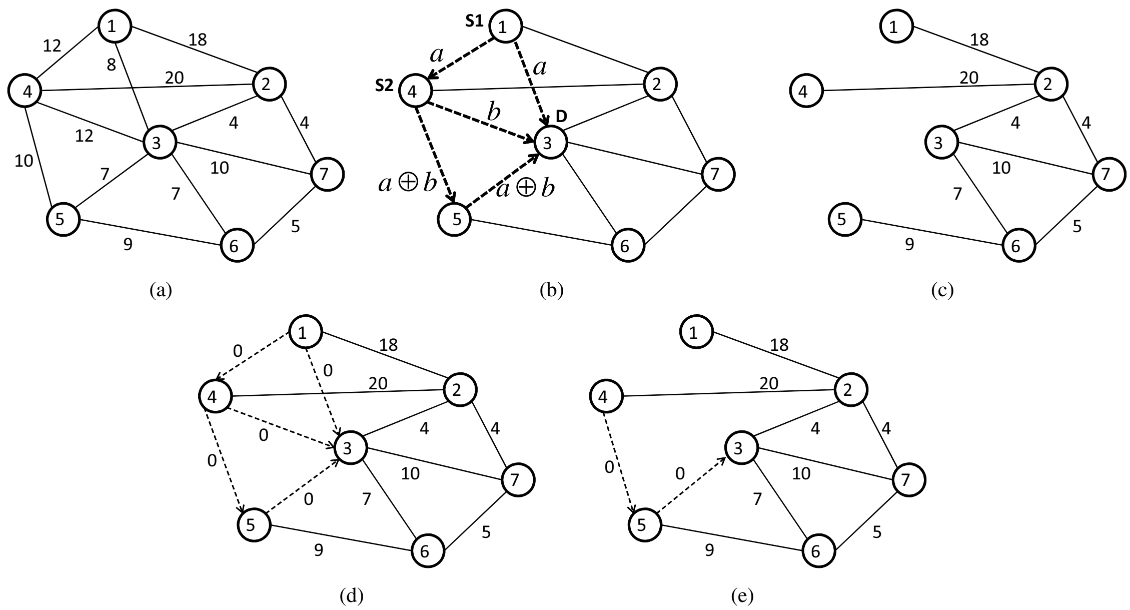


Fig. 12. (a) The topology of the network associated with the regular link costs, (b) An existing coding group, (c) The network seen by the primary path, (d) The network seen by the secondary path in non-systematic coding, (e) The network seen by the secondary path in systematic coding.

TABLE VI
SIMULATION RESULTS OF SMALLNET NETWORK

Smallnet Network, 10 nodes, 22 spans					
Scheme	TC	RT for different X values (ms)			
		0.5ms	1ms	5ms	10ms
N-S. Div. Cod.	85800	$2 \times F + 2 \times M + S + T$ (60μs)			
S. Div. Cod.	88500	$F + M + S$ (30μs)			
$1 + 1$ APS	103700	$F + S$ (20μs)			
SPP	75500	4.59	5.09	9.09	14.09
p-cycle	70900	5.11	5.61	9.61	14.61

TABLE VII
SIMULATION RESULTS OF NSFNET NETWORK

NSFNET Network, 14 nodes, 21 spans					
Scheme	TC	RT for different X values (ms)			
		0.5ms	1ms	5ms	10ms
N-S. Div. Cod.	1704855	$2 \times F + 2 \times M + S + T$ (60μs)			
S. Div. Cod.	1754460	$F + M + S$ (30μs)			
$1 + 1$ APS	1881880	$F + S$ (20μs)			
SPP	1580040	52.61	53.11	57.11	62.11
p-cycle	1609055	76.21	76.71	80.71	85.71

requirement of feedforward signaling when both primary and protection paths are coded marginally increases the restoration time of nonsystematic diversity coding compared to the case of systematic coding. On the other hand, both versions of diversity coding are more capacity efficient than $1 + 1$ APS. Both versions of diversity coding have slightly lower capacity efficiency than diverse routing and p-cycle protection in both networks. However, the difference in terms of capacity efficiency may become negligible compared to the speed advantage of diversity coding for the network designers in pursuit of higher restoration speed without strict restrictions on the fiber capacity. It is observed that increased coding flexibility in non-systematic diversity coding creates a bigger advantage in highly connected networks than the relatively sparse networks.

IV. CONCLUSION

This paper presents optimal design algorithms of diversity coding for pre-provisioning and dynamic provisioning against single link failures. The adopted diversity coding technique has two variations, employing non-systematic and systematic coding, respectively. Only the connections with the same destination node are encoded together in both of the variations of diversity coding. We were able to achieve sub-ms restoration time with this scheme. In dynamic provisioning, optimality of the design algorithm is supported with a set of assumptions.

We developed a novel MIP formulation that routes and protects a set of static traffic demands optimally. The significance of this algorithm is lower complexity compared to the similar techniques in the literature. The MIP formulation forms the coding groups, first. Then, it creates a primary tree structure for each coding group which serves as the primary paths of the connections in that coding group. The primary tree is accompanied by a link-disjoint protection tree which replaces the protection paths in that coding group. The coding operations in the protection tree require no extra variable. As a result, the new formulation proves to be simpler in terms of the number of integer variables and constraints. Evaluation of the simulation results indicate that both versions of diversity coding can achieve sub-ms restoration time. Diversity coding is much faster than the other techniques except $1 + 1$ APS in each scenario. On the other hand, SPP and the p-cycle protection are more capacity efficient than diversity coding for these scenarios. However, it offers a desirable tradeoff where one can achieve a speed gain of more than hundred times with less than 26% extra capacity over SPP and the p-cycle protection.

In the second part of the paper, an ILP-based dynamic provisioning algorithm is developed for dynamic traffic. New connections are routed and protected one-by-one by using both versions of diversity coding. The design algorithm is both optimal and fast enough to provision the quickly changing traffic. The idea behind the algorithm is to add new connection demands to the existing suitable coding groups without impairing the integrity of the existing connections. A new demand is added to the coding group which requires the lowest extra capacity. The evaluations of the simulation results are consistent with the results in static pre-provisioning. Both versions of diversity coding are still significantly faster than p-cycle protection and diverse routing. However, the capacity efficiency of diversity coding gets closer to those of p-cycle protection and diverse routing compared to the static traffic scenario.

V. Acknowledgements

The authors would like to thank the anonymous reviewers whose comments improved the quality of the presentation in the paper.

References

[1] M. To and P. Neusy, "Unavailability analysis of long-haul networks," *IEEE J. Sel. Areas Commun.*, vol. 11, pp. 100–109, Jan. 1994.

[2] M. Menth, M. Duelli, and J. Milbrandt, "Resilience analysis of packet-switched communication networks," *IEEE/ACM Trans. Netw.*, vol. 17, no. 6, p. 1, Dec. 2009.

[3] W. D. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice-Hall PTR, 2004.

[4] S. Avci, X. Hu, and E. Ayanoglu, "Recovery from link failures in networks with arbitrary topology via diversity coding," in *Proc. 2011 IEEE GLOBECOM*, pp. 1–6.

[5] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Elsevier, 2004.

[6] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightwave Technol.*, vol. 21, no. 4, pp. 870–883, Apr. 2003.

[7] R. Iraschko, M. H. MacGregor, and W. D. Grover, "Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks," *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 325–336, June 1998.

[8] W. Grover and D. Stamatelakis, "Cycle-oriented distributed preconfiguration: ring-like speed with mesh-like capacity for self-planning network restoration," in *Proc. 1998 ICC*, vol. 1, pp. 537–543.

[9] G. Li, A. Chiu, and J. Strand, "Resilience design in all-optical ultralong-haul networks," *J. Opt. Netw.*, vol. 5, no. 8, pp. 625–636, July 2006.

[10] T. Chow, F. Chudak, and A. Ffrench, "Fast optical layer mesh protection using pre-cross-connected trails," *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 539–548, June 2004.

[11] 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. 1990 IEEE INFOCOM*, vol. 1, pp. 95–104.

[12] ——, "Diversity coding for transparent self-healing and fault-tolerant communication networks," *IEEE Trans. Commun.*, vol. 41, pp. 1677–1686, Nov. 1993.

[13] A. Kodian and W. D. Grover, "Failure-independent path protecting p-cycles; efficient and simple fully pre-connected optical path protection," *J. Lightwave Technol.*, vol. 23, no. 10, pp. 3241–3259, Oct. 2005.

[14] S. Avci and E. Ayanoglu, "Extended diversity coding: coding protection and primary paths for network restoration," in *Proc. 2012 International Symposium on Network Coding*, pp. 1–6.

[15] ——, "Optimal algorithms for near-hitless network restoration via diversity coding," in *Proc. 2012 IEEE GLOBECOM*, pp. 1–7.

[16] ——, "Coded path protection: efficient conversion of sharing to coding," in *Proc. 2012 IEEE ICC*.

[17] A. E. Kamal and O. Al-Kofahi, "Efficient and agile 1+N protection," *IEEE Trans. Commun.*, vol. 59, no. 1, pp. 169–180, Jan. 2011.

[18] H. Øverby, G. Bóizck, P. Barbarczi, and J. Tapolcai, "Cost comparison of 1+1 path protection schemes: a case for coding," in *Proc. 2012 IEEE ICC*.

[19] Z. Liu, M. Li, L. Lu, C.-K. Chan, S.-C. Liew, and L.-K. Chen, "Optical physical-layer network coding," *IEEE Photon. Technol. Lett.*, vol. 24, no. 16, pp. 1424–1427, Aug. 2012.

[20] 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, Feb. 2010.

[21] O. M. Al-Kofahi and A. E. Kamal, "Network coding-based protection of many-to-one wireless flows," *IEEE J. Sel. Areas Commun.*, vol. 27, no. 5, pp. 787–813, June 2011.

[22] B. Mukherjee, S. Ramamurthy, D. Banerjee, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Trans. Netw.*, vol. 4, no. 5, pp. 684–706, Oct. 1996.

[23] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proc. 2003 ACM SIGMETRICS*.

[24] C. Ou, H. Zang, N. K. Singhal, K. Zhu, L. H. Sahasrabuddhe, R. A. Macdonald, and B. Mukheerjee, "Subpath protection for scalability and fast recovery in optical WDM mesh networks," *IEEE J. Sel. Area Commun.*, vol. 22, no. 9, p. 1859-1875, Nov. 2004.

[25] S. Avci and E. Ayanoglu, "Design algorithms for fast network restoration via diversity coding," in *Proc. 2012 UCSD Information Theory and Applications Workshop*, pp. 1–7.

[26] M. Mohandespour and A. E. Kamal, "1+N protection in polynomial time: a heuristic approach," in *Proc. 2010 IEEE GLOBECOM*, pp. 1–5.

[27] W. He, J. Fang, and A. Somani, "A p-cycle based survivable design for dynamic traffic in WDM networks," in *Proc. 2005 IEEE GLOBECOM*, vol. 4, pp. 1869–1873.

[28] P. H. Ho, J. Tapolcai, and H. T. Mouftah, "On achieving optimal survivable routing for shared protection in survivable next-generation Internet," *IEEE Trans. Rel.*, vol. 53, no. 2, pp. 216–225, June 2004.

Serhat Nazim Avci (S'12) received his B.S. degree in Electrical and Electronics Engineering from Bilkent University, Ankara, Turkey in June 2009. During the same year, he was accepted into the Department of Electrical Engineering and Computer Science of the University of California, Irvine with a departmental fellowship. He received his M.S. degree from this department in December 2011. Currently, he is a Ph.D. candidate in the same department. His research interests are in applications of coding theory into problems in networking, in variations of linear programming, especially in the generation of fast algorithms for linear programming applications such as cutting stock or column generation solutions.

Ender Ayanoglu (S'82-M'85-SM'90-F'98) received the B.S. degree from the Middle East Technical University, Ankara, Turkey, in 1980, and the M.S. and Ph.D. degrees from Stanford University, Stanford, CA, in 1982 and 1986, respectively, all in electrical engineering. He was with the Communications Systems Research Laboratory, AT&T Bell Laboratories, Holmdel, NJ (Bell Labs, Lucent Technologies after 1996) until 1999 and was with Cisco Systems until 2002. Since 2002, he has been a Professor in the Department of Electrical Engineering and Computer Science, University of California, Irvine, where he served as the Director of the Center for Pervasive Communications and Computing and held the Conexant-Broadcom Endowed Chair during 2002-2010. Dr. Ayanoglu is the recipient of the IEEE Communications Society Stephen O. Rice Prize Paper Award in 1995 and the IEEE Communications Society Best Tutorial Paper Award in 1997. Since 1993, he has been an Editor of the IEEE Transactions on Communications and served as its Editor-in-Chief from 2004 to 2008. From 1990 to 2002, he served on the Executive Committee of the IEEE Communications Society Communication Theory Committee, and from 1999 to 2001, was its Chair.