

Coded Path Protection: Efficient Conversion of Sharing to Coding

Serhat Nazim Avci and Ender Ayanoglu

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

Abstract—Link failures in wide area networks are common and cause significant data losses. Mesh-based protection schemes offer high capacity efficiency but they are slow and require complex signaling. Additionally, real-time reconfiguration of a cross-connect threatens their transmission integrity. On the other hand, coding-based protection schemes are proactive. Therefore, they have higher restoration speed, lower signaling complexity, and higher transmission integrity. This paper introduces a coding-based protection scheme, named Coded Path Protection (CPP). In CPP, a backup copy of the primary data is encoded with other data streams, resulting in capacity savings. This paper presents an optimal and simple capacity placement and coding group formation algorithm. The algorithm converts the sharing structure of any solution of a Shared Path Protection (SPP) technique into a coding structure with minimum extra capacity. We conducted quantitative and qualitative comparisons of our technique with the SPP and, another technique, known as p -cycle protection. Simulation results confirm that the CPP is significantly faster than the SPP and p -cycle techniques. CPP incurs marginal extra capacity on top of SPP. Its capacity efficiency is lower than the p -cycle technique for dense networks but can be higher for sparse networks. In addition, unlike p -cycle protection, CPP is inherently suitable for the wavelength continuity constraint in optical networks.

I. INTRODUCTION

Studies show that reasons of failure in networks can be widespread. According to [1], cable cut rate per 1000 sheath miles per year is 4.39. That means on average a cable-cut occurs every three days per 30,000 fiber miles. These numbers are consistent with the data published by FCC. As stated in [2], 70% percent of the unplanned network failures affect only single links. For this reason, in this paper, we focused on single link failure recovery.

1+1 and 1:1 automatic protection switching were early attempts of path-based protection mechanisms but were dropped due to low capacity efficiency. Mesh-based protection schemes attracted attention due to their high capacity efficiency but suffered from low speed. SPP [3] is a widely recognized mesh-based path protection technique. It specifies two link-disjoint paths for each connection and reroutes the traffic over the protection path if the primary path fails.

Reference [4] introduced the concept of a p -cycle in order to achieve both fast restoration and low spare capacity percentage. Fundamentally, a p -cycle is a mixture of ring-type protection and link-based protection. Its performance is similar to SPP in terms of resource utilization and similar to link-based protection in terms of restoration time. In the case of a failure

in a link protected by the cycle, the affected traffic is rerouted over the spare capacity in the healthy parts of the p -cycle.

The p -cycle approach achieves higher restoration speed by simply minimizing the number of optical cross-connect (OXC) configurations after failure. “Hot-standby” [5] and “pre-cross-connected trials” (PXT) [6], which are extensions of SPP, are developed based on the same idea. We offer a novel proactive protection scheme called Coded Path Protection (CPP). It is faster and more stable than rerouting based schemes because it eliminates the real-time OXC configurations after failure. The capacity placement algorithm of CPP is based on converting the sharing operation of SPP into coding and decoding operations with a slight extra cost. Integer linear programming (ILP) is incorporated to carry out the optimal conversion with minimum total capacity. In the next sections, comparisons between our schemes and aforementioned conventional techniques are performed. Simulations over realistic network scenarios using ILP formulations are carried out.

II. RELATED WORK

The idea of incorporating network coding into link failure protection dates back to 1990 [7] and 1993 [8], prior to the first papers on network coding [9]. The technique is called *diversity coding*, and in it, N primary links are protected using a separate $N + 1^{st}$ protection link which carries the modulo-2 sum of the data signals in each of the primary links. If all of the $N + 1$ links were disjoint or physically diverse, then any single link failure could be recovered from by applying the modulo-2 sum over the received links. Assume that data on the primary links are $b_1, b_2, b_3, \dots, b_N$ and the checksum of the primary data is

$$c_1 = b_1 \oplus b_2 \oplus \dots \oplus b_N = \bigoplus_{j=1}^N b_j.$$

In the receiver side of the operation, if a failure is detected, the decoder applies modulo-2 sum to the rest of the N links and extract the failed data as

$$c_1 \oplus \bigoplus_{\substack{j=1 \\ j \neq i}}^N b_j = b_i \oplus \bigoplus_{\substack{j=1 \\ j \neq i}}^N (b_j \oplus b_j) = b_i$$

where we assumed b_i is the failed link. This operation is fundamentally different than rerouting-based protection schemes since it does not need any feedback signaling.

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

This idea was revisited by the authors of this paper in [10] and a coding structure for an arbitrary network topology was developed. This scheme may require extra links from the destination nodes to decoding nodes to be able to decode signals. It has been shown in [10] that diversity coding can achieve higher capacity efficiency than the SPP and the p -cycle techniques.

In [11], a bidirectional protection scheme that uses network coding over p -cycle topologies on mesh networks was introduced and called as 1+N protection. The idea presented in [11] is to form circular protection paths in both directions that traverse over the source and destination nodes of the group of flows that are to be protected. In [12], a new tree-based protection scheme was introduced instead of a p -cycle based scheme and called Generalized 1+N protection (G1+N). In [12], same data from both end-nodes are sent on a parity link. Symmetric transmission is broken only for the connection affected from the failure. The capacity efficiency of G1+N is basically unknown. However, it clearly lacks the speed of diversity coding since the distance between the destination nodes and the decoding node is much longer than for diversity coding.

III. CODED PATH PROTECTION

In this paper, we propose a novel coding technique, which we call Coded Path Protection (CPP). We present a simple strategy to find the optimal coding structure without much complexity. Coded path protection is faster, has less signaling complexity, and has higher transmission integrity than any of the rerouting-based protection techniques. Spare capacity percentage (SCP) of coded path protection is slightly larger than the SCP of shared path protection. Our contribution in this paper consists of two parts, namely a novel coding structure and a simple but optimal coding group formation algorithm.

In this paper, we extend the basic idea in [12] to transform a SPP solution into a CPP solution so that sharing is replaced with coding. CPP is a proactive protection technique since two copies of the same data are sent through the primary and protection paths at a non-failure state so that protection is immediately available when a failure occurs on the primary path. We offer a slightly different coding structure than G1+N coding. Symmetric transmission is key in coding and decoding operations. This is illustrated in Fig. 1(a) in an example with two connections. Thick straight lines are primary paths and dotted lines are protection paths. For the time being, synchronization and timing are not considered. Assume that $S1$ transmits s_1 to $D1$ and $D1$ transmits d_1 to $S1$ using the primary path at time t_0 . After a delay of τ , these signals are received by the reciprocal nodes at the same time and both end-nodes form the summation of these two signals, mathematically $c_1 = s_1 \oplus d_1$. At time $t_0 + \tau$, the same c_1 symbols are sent from the corresponding end-nodes of the protection path of $S1 - D1$. It is similar for $S2 - D2$. As it is seen at Fig. 1(a), c_1 and c_2 are coded over the link $A - B$. A is the node where c_1 and c_2 are coded and node B is responsible for decoding. Node B extracts c_1 using $c_1 \oplus c_2$ and c_2 , and extracts c_2 using $c_1 \oplus c_2$ and c_1 . Therefore coding and decoding of signals c_1 and c_2 are successfully completed and node B transmits them to $D1$ and $D2$ over links $B - D1$

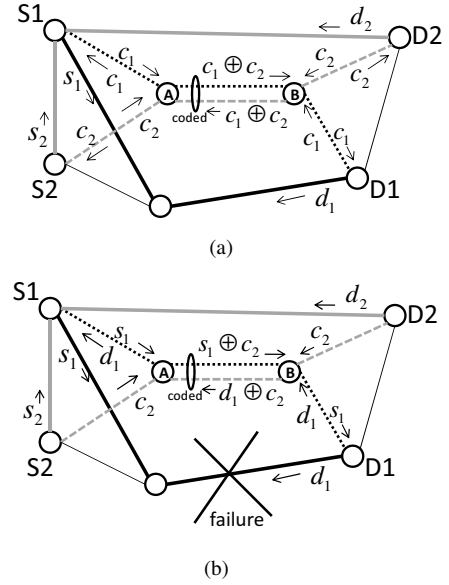


Fig. 1. Coding and decoding operations for coded path protection, (a) In normal state, (b) Reaction to single link failure

and $B - D2$ respectively. Fig. 1(b) gives an example of single link failure on the primary path of $S1 - D1$. Due to failure, $S1$ receives 0 instead of d_1 and transmits $s_1 \oplus 0 = s_1$ at time $t_0 + \tau$ over the protection path. Similarly, $D1$ receives 0 instead of s_1 , so it transmits $0 \oplus d_1 = d_1$. These signals are coded with c_2 over link $A - B$ and they are decoded at nodes B and A respectively. At node B , s_1 is extracted by summing $s_1 \oplus c_2 \oplus c_2 = s_1$ and forwarded over the link $B - D1$. At node A , d_1 is acquired by summing $d_1 \oplus c_1 \oplus c_1 = d_1$ and forwarded over the link $A - S1$. Reciprocity over the protection path of $S2 - D2$ enables perfect coding and decoding of the data signals of the failed connection.

“Poison-antidote” analogy [13] is useful in understanding the general coding structure. When two signals are coded together they “poison” each other. At the decoding node, “antidote” data are needed to extract the signals from each other. For the general two connection case in Fig. 2, same signals traverse the reverse directions over the protection path. Straight lines are protection paths of a and b in one direction. Dotted lines are protection paths of same a and b in the reverse direction. At node A , straight paths are antidotes of dotted paths. At node B , dotted paths are antidotes of straight paths. In the single link failure case, if connections have link-disjoint primary paths, at most one of them is affected from the failure. The other connection can preserve reciprocity and the poison-antidote structure to help recover the affected connection. If the primary path of a and the protection path of b , or vice versa, shares a common link and it fails, reciprocity is broken for both of them. In this case, the connection whose protection path fails will stop transmission over the protection path to stop “poisoning” since there is no “antidote.”

In order to generalize this coding structure for any network, first we should define the concept of a coding group. Assume that $S1 - D1$ and $S2 - D2$ are coded over some link E . Then, they are considered to be in the same coding group. This group gets bigger if either $S1 - D1$ or $S2 - D2$ is coded

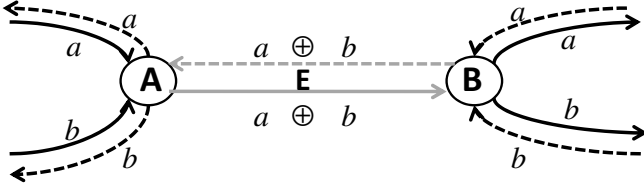


Fig. 2. Coding at an arbitrary link and decoding at an arbitrary node.

with some other connection. For example, if $S3-D3$ is coded with $S2-D2$, it is also considered in the same coding group with both $S2-D2$ and $S1-D1$. In a coding group, coding structure can recover from a single link failure on one of the primary paths if the reciprocity property is preserved for the other connections. To guarantee this property, two protection paths can be coded together as long as

- 1) Their primary paths are link-disjoint,
- 2) Their primary paths are also link-disjoint with the primary paths of the connections in the same coding group.

If there are more than two connections of interest, the same rules are applied on any couple of connections to guarantee perfect coding and decoding. In a realistic network scenario, buffering and synchronization mechanisms are required. We exclude this discussion here due to space limitations.

The scheme in [12] is similar to CPP with two differences. First, its capacity efficiency is not known. Second, there does not exist a simple and optimal algorithm to implement it for an arbitrary topology. However, CPP is suitable to convert a typical solution of SPP with low complexity because a typical solution of SPP must obey the first rule above. The rest of the work to convert an SPP solution to CPP is to form coding groups that satisfy the second rule.

We assume that for a given topology and a given set of connections, there is a pre-calculated solution of SPP. Given the solution, primary and protection paths of the connections, wavelength assignments, and maximum required spare capacity on each link will be known. Referring to Fig. 3(a), thick straight lines represent the primary paths of end-to-end connections, whereas protection paths are stated by dotted lines. In Fig. 3(a), numbers associated with edges are index values of edges. Some of the protection capacity is shared by multiple protection paths. There is a limited freedom in terms of choosing the group of connections which will share the same capacity over the same link. For example, $S3-D3$ can share the one unit spare capacity at link 5 either with connection $S1-D1$ or with connection $S4-D4$. However, $S1-D1$ and $S4-D4$ cannot share that capacity since their primary paths are not link-disjoint. This freedom can be utilized in converting sharing groups to valid coding groups with zero or unappreciable additional capacity.

In the given solution of SPP, protection paths are coupled under the provision of the first rule. However, while building the CPP solution, protection paths are coupled and coding groups are formed in a way such that both rules are satisfied. The sharing structure in Fig. 3(a) is converted to the coding structure in Fig. 3(b) in this manner. It should be noted that at link 5 $S1-D1$ and $S3-D3$ are coupled to share the one unit capacity in the SPP solution. However, in CPP solution $S3-D3$ is coded with $S4-D4$, not with $S1-D1$. If that is

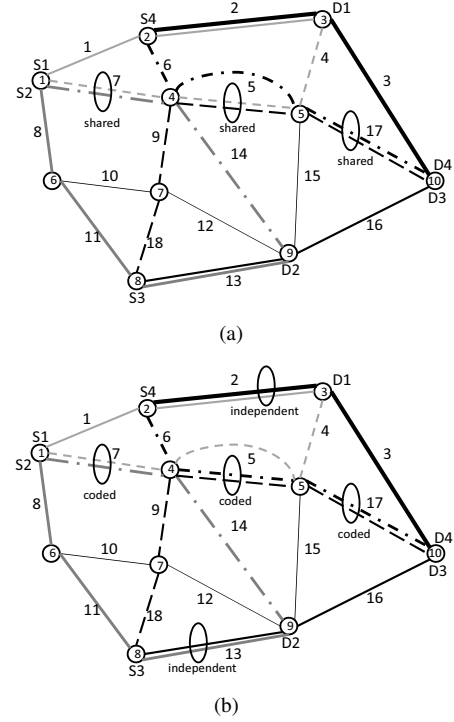


Fig. 3. Possible coding and sharing scenarios over a network, (a) Sharing of protection capacities, (b) Coding of protection paths

not done, then $S1-D1$, $S2-D2$, $S3-D3$ and $S4-D4$ would be in the same coding group because they would be indirectly related. Then the second rule about link-disjointness would not be satisfied. After this modification, we can divide this coding group into two, one group consists of $S1-D1$ and $S2-D2$ and the other consists of $S3-D3$ and $S4-D4$. Then both of the rules are satisfied. In this example, no extra capacity is required to convert a SPP solution to a CPP solution with the aid of limited freedom in the SPP solution. However, that is not the case in general. Therefore, we developed an ILP formulation to conduct the conversion with minimum extra capacity.

IV. THE ILP FORMULATION

We developed an ILP formulation to find the optimal SPP solution for a given set of traffic scenarios and networks. We also developed an ILP formulation to convert this SPP solution into a viable CPP solution. The ILP formulation is varied to include both the wavelength continuity constraint and its absence in order to cover different types of optical networks.

As stated in [14], the problem of joint path routing and wavelength assignment is a very complex problem. Therefore we developed a suboptimal ILP formulation for the SPP algorithm. However, the ILP formulation of conversion from the SPP to the CPP algorithm is optimal. We input a set of possible paths for the connections and run the optimal wavelength assignment and sharing algorithm. Due to space limitations, the ILP formulation of SPP is not discussed here.

The ILP formulation of CPP has the following set of input parameters (When it is stated “equals 1 if A is true,” it simultaneously means “equals 0 if A is not true”)

- $G(V, E)$: The network graph
- N : Enumerated list of bidirectional connections
- c_e : Cost of each link
- $d_e(i)$: Equals 1 if the protection path of connection i traverses over link e , is acquired from the solution of SPP
- $m(i, j)$: Equals 1 if the primary paths of connection i and connection j are link-disjoint, is acquired from the solution of SPP.

In addition to the input parameters, there are a number of binary variables

- $n(i, j)$: Equals 1 if connection i and connection j are in the same coding group
- $r_e(i, j)$: Equals 1 if connection i and connection j are coded together over link e
- $s_e(i)$: Equals 1 if protection capacity of connection i over link e can be saved with coding

Both $n(i, j)$ and $m(i, j)$ are defined in a way that $i < j$. The objective function is

$$\min \sum_{e \in E} \sum_{i \in N} c_e \times (d_e(i) - s_e(i)) \quad (1)$$

subject to the following constraints

$$r_e(i, j) \leq n(i, j), \quad \forall i, j \in N, i < j, \forall e \in E, \quad (2)$$

$$r_e(i, j) \leq d_e(i), \quad \forall i, j \in N, i < j, \forall e \in E, \quad (3)$$

$$r_e(i, j) \leq d_e(j), \quad \forall i, j \in N, i < j, \forall e \in E, \quad (4)$$

$$n(i, j) \leq m(i, j), \quad \forall i, j \in N, i < j. \quad (5)$$

Inequality (2) ensures that if two connections are coded over a link, then they must be in the same coding group. Inequalities (3) and (4) ensure that if two connections are coded over a link their protection paths must traverse over that link. Inequality (5) makes sure only link-disjoint connections can be in the same coding group. In addition,

$$n(i, j) \geq n(i, k) + n(k, i) + n(j, k) + n(k, j) - 1 \quad (6)$$

$\forall i, j, k \in N, i < j$. If $i \geq k$ and $j \geq k$, $n(i, k) = n(j, k) = 0$ and vice versa. Inequality (6) ensures that if connection k is in the same coding group with connection i and j , then connection i and j are also in the same coding group. The inequality

$$s_e(i) \leq \sum_{1 \leq k < i} r_e(k, i), \quad \forall i, k \in N, k < i, \forall e \in E \quad (7)$$

calculates the savings due to the coding operation. When multiple protection paths are coded over the same link, only the one with the smallest index is accounted for the used capacity. Others save capacity by coding over the smallest indexed path.

ILP formulation for the p -cycle approach is adapted from [15]. Simulations are based on cycle exclusion-based ILP for spare capacity placement [15].

V. RESTORATION TIME AND SIGNALING

In this section, we conduct a qualitative and a quantitative analysis in terms of restoration time of the SPP, CPP, and the p -cycle approaches. The analysis is extended to cover

both opaque and transparent optical networks [16]. The developments in the optical XOR operations [17] allow coded shared protection to be applicable in all-optical networks [16]. Wavelength assignment of CPP is trivial after converting the wavelength assignment solution of SPP because CPP is inherently suited to the wavelength continuity constraint. With this constraint, protection paths in the same coding group make use of the same wavelength throughout the network. SPP [3] is proposed for all-optical networks but some shared path protection techniques, such as [6], and the p -cycle techniques make use of “optical-electrical-optical”(o-e-o) conversion at intermediate nodes.

Protection in CPP is a proactive mechanism because the second (protection) copy of any data is generated and transmitted by the source node to the destination node after a fixed time delay. An advantage of proactive protection mechanism is the continuous operation over protection paths which means there is no need to configure and test an OXC after any failure. OXC configuration and testing is the main source of delay in routing based protection mechanisms [18]. In addition, this proactive mechanism eliminates the need of complex signaling and assures transmission integrity because the operations are all automatic. As stated in [18], transmission integrity can be the main problem in configuring protection paths and routing data over these paths in optical networks. This claim is supported by the stability concerns cited in [5].

Despite the fact that CPP is a proactive mechanism, it can utilize the signaling capabilities of opaque optical networks to fasten the recovery process. In some cases, a synchronization mechanism with different data streams can neutralize the time savings of CPP. For that purpose, we propose a two-tier protection mechanism available for opaque networks. Transparent networks need to stick with the proactive mechanism due to the weak signaling capability of all-optical networks. In the first step, protection prompts as it is synchronized. As a second step, when the end-nodes receive the error signals, they add one bit control message to the data signals, which transform them into “ambulance” signals. The “ambulance” signals skip the buffers and are coded and encoded with data streams consisting of all zeros. The second step is similar to SPP but there is no need to configure the cross-connects. As a result, CPP is faster and more stable even in the worst case. The restoration time of the first part of the operation is

$$RT_{CPP1} = d_{sd} + h_b \times M + S,$$

where d_{sd} is the propagation delay from node s to node d and h_b is the number of hops in the protection path between d and s . The symbol M denotes the node processing time and varies on the type of optical network. It is taken 0.3 ms in [12] and 10 μ s in [2]. The symbol S represents the delay due to synchronization. The restoration time formulation of the second step is

$$RT_{CPP2} = F + 2 \times d_{sd} + (h_{is} + 1) \times M + (h_b + 1) \times M,$$

where F is the failure detection time and h_{is} is number of nodes between node i and node s . The exact formulation of CPP for opaque optical networks is

$$RT_{CPP} = \min(RT_{CPP1}, RT_{CPP2}).$$

The recovery process in SPP starts with failure detection. Failure notification is required before end-nodes switch the traffic from primary to protection paths. Intermediate nodes configure themselves after they receive error state signals. In the protection switching step, some researchers claim that nodes in the protection path configure the OXCs simultaneously which leads to significant restoration time savings [14]. Error state signals should be transmitted over a specialized control plane to notify every node to enable simultaneous configuration of cross-connects over the protection path. As a tradeoff, this incurs high signaling complexity throughout the network. The restoration time formulation of SPP is under discussion, e.g., the results of some of the formulations [3], [14] do not match the numerical results in [19]. The OXC configuration time is stated to possibly be about 10 ms, but it is also reported to be as much as one second [5]. In addition, in [20] it is pointed out that an extra 40-80 ms is required only for signaling and reconfiguration, such as uploading maps. This means the OXC configuration time is not the only source of delay in SPP. Keeping the ambiguity in mind, we adopt the formula in [14] for the quantitative analysis of restoration time in SPP, assuming a separate packet-based control plane exists and has the same topology with the network of interest. The symbol X represents the OXC configuration and test time, so that

$$RT_{SPP1} = F + 2 \times d_{sd} + (h_{si} + 1) \times M + X + (h_b + 1) \times M.$$

If a specialized control plane does not exist, in other words if in-band signaling is employed, then the OXCs cannot transmit the control message before they reconfigures themselves. This leads to higher restoration time due to the reconfiguration of OXCs in series. The formula for this case is adopted from [3]

$$RT_{SPP2} = F + d_{sd} + (h_{is} + 1) \times M + (h_b + 1) \times X + 2 \times d_{sd} + 2 \times (h_b + 1) \times M.$$

The p -cycle approach is fundamentally a mixture of link protection technique and ring-type protection technique. This approach generally results in lower restoration time than SPP since the operations are local. P -cycle offers pre-connected OXCs around the cycle, so it minimizes the number of reconfigurations of OXCs after a failure. Only the end-nodes of the failure need to switch the traffic. However, an efficient p -cycle consists of many nodes, and traverses a long distance. This can add significant propagation and node processing delays in relatively large networks, such as the U.S. long-haul network. We employ the formula in [21] to calculate the restoration time of the p -cycle technique. This formula is modified to include the propagation delay after the source-end node of the failed link switches to the p -cycle until destination-end node of the failed link receives new data packets. In this paper, M is taken as 0.3 ms [22], because the p -cycle uses o-e-o conversions. 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. Then,

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

Numerical results of worst case restoration time (WCRT) for the three techniques and a quantitative analysis are provided in the next section.

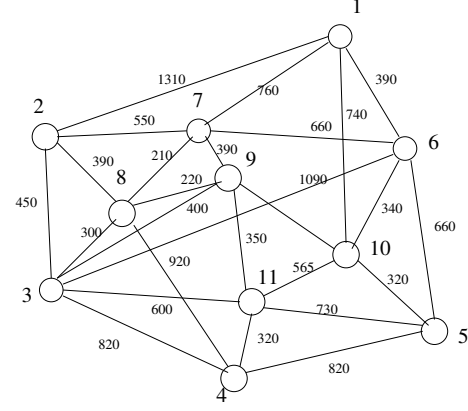


Fig. 4. European COST 239 network.

VI. SIMULATION RESULTS

In this section, we will present simulation results for link failure recovery techniques previously discussed, in terms of their spare capacity requirements and worst case restoration time. In order to conduct a fair comparison between protection schemes, we input the same set of possible routing scenarios to ILP formulations of the SPP and the p -cycle approach. Since the CPP solutions are based on the SPP solutions, they also utilize the same set of routing scenarios.

The first network studied is the European COST 239 [23] network whose topology is given in Figure 4. In Fig. 4 and Fig. 5, the numbers associated with the nodes represent a node index, while the numbers associated with the edges correspond to the distance (cost) of the edge. The distances are useful to calculate the propagation delays. The traffic demand is uniform. SCP represents spare capacity percentage and explained in [10]. We provide the SCP values without the wavelength continuity constraint and WCRT time results for the three schemes in Table I. In the third column, ESCP means the extra spare capacity percentage required to satisfy the wavelength continuity constraint [24].

Second network studied is the NSFNET network [12], similar to the U.S long-haul network [10]. Again, the traffic scenario is uniform. We provide the SCP values and WCRT results for the three schemes in Table II.

As seen from the results, converting the SPP solution to CPP results in approximately 6-7% extra spare capacity percentage.

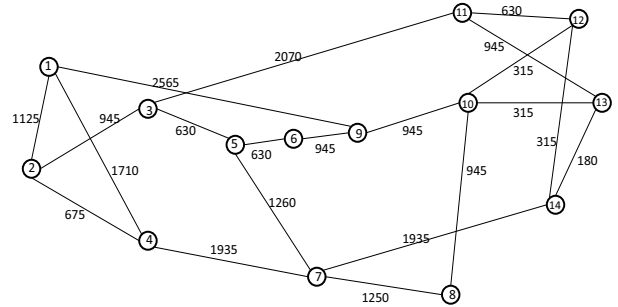


Fig. 5. NSFNET network.

TABLE I
SIMULATION RESULTS OF COST 239 NETWORK

COST 239 Network, 11 nodes, 26 spans						
Scheme	SCP	ESCP	WCRT for different X values (ms)			
			0.5ms	1ms	5ms	10ms
CPP	72.71%	0%	11.57	11.57	11.57	11.57
SPP1	64.67%	0%	17.86	18.36	22.36	27.36
SPP2	64.67%	0%	28.5	31.1	51.1	76.1
p-cycle	44.82%	40-60%	25.31	25.81	29.81	34.81

TABLE II
SIMULATION RESULTS OF NSFNET NETWORK

NSFNET Network, 14 nodes, 21 spans						
Scheme	SCP	ESCP	WCRT for different X values (ms)			
			0.5ms	1ms	5ms	10ms
CPP	95.26%	0%	34.65	34.65	34.65	34.65
SPP1	88.41%	1.43%	51.54	52.04	56.04	61.04
SPP2	88.41%	1.43%	79.01	81.51	101.5	126.5
p-cycle	84.51%	40-60%	76.43	76.93	80.93	85.93

On the other hand, the restoration speed increases three times over SPP2 when in-band signaling is used and increases two times over SPP1 when there is a separate control plane for SPP scheme. The restoration time of SPP increases as the expected time of OXC configuration and test increases. Realistically, in some cases it may take seconds. The p -cycle technique results in lower SCP than CPP without the wavelength continuity constraint. Under the wavelength continuity constraint, CPP is as capacity efficient as the p -cycle technique for the COST 239 network and is more capacity efficient than p -cycle technique for the NSFNET network. It is observed that capacity efficiency of the p -cycle technique vanishes while going towards more sparse networks. The CPP is at least twice as fast as p -cycle technique.

VII. CONCLUSION

In this paper, we introduced a proactive network restoration technique we call Coded Path Protection (CPP). The technique makes use of symmetric transmission over protection paths and link-disjointness among the connections in the same coding group. We modified the coding structure and leveraged its flexibility to convert sharing structure of a typical solution of SPP into a coding structure of CPP in a simple manner. With this approach, it is possible to quickly achieve close to optimal solutions. As a result of this operation, the CPP algorithm has the following properties.

- The restoration speed is 2 to 3 times faster than SPP and the p -cycle technique
- Full transmission integrity and stability
- Low signaling complexity
- Protection is independent of any single link failure scenario
- Simulation complexity significantly reduced over generalized 1+N coding
- Lower spare capacity than p -cycle under wavelength continuity constraint

with the tradeoff of

- At most 6-7 % extra spare capacity over SPP
- Lower capacity efficiency than p -cycle technique in dense networks

- Additional synchronization and buffering

Although the capacity placement algorithm for SPP employed in this paper is not optimal, this does not affect the optimality of the conversion algorithm.

REFERENCES

- [1] J. Zhang and B. Mukherjee, "Review of fault management in WDM mesh networks: Basic concepts and research challenges," *IEEE Network*, vol. 18(2), pp. 41–48, March–April 2004.
- [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, December 2009.
- [3] S. Ramamurthy, L. Sahasrabudhe, and B. Mukherjee, "Survivable WDM mesh networks," *J. Lightwave Technol.*, vol. 21, no. 4, pp. 870–883, April 2003.
- [4] 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.
- [5] 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.
- [6] 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.
- [7] 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.
- [8] —, "Diversity coding for transparent self-healing and fault-tolerant communication networks," *IEEE Trans. Commun.*, vol. 41, pp. 1677–1686, November 1993.
- [9] 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.
- [10] S. Avci and E. Ayanoglu, "Recovery from link failures in networks with arbitrary topology via diversity coding," in *Proc. Globecom (to appear)*, December 2011.
- [11] A. E. Kamal, "1+N network protection for mesh networks: Network coding-based protection using p -cycles," *IEEE/ACM Trans. Netw.*, vol. 18, no. 1, pp. 67–80, Feb. 2010.
- [12] A. E. Kamal and O. Al-Kofahi, "Efficient and agile 1+N protection," *IEEE Trans. Comm.*, vol. 59, no. 1, pp. 169–180, January 2011.
- [13] D. Traskov, N. Ratnakar, D. Lun, R. Koetter, and M. Médard, "Network coding for multiple unicasts: An approach based on linear optimization," in *Proc. IEEE ISIT*, July 2006, pp. 1758–1762.
- [14] C. O. et al., "Subpath protection for scalability and fast recovery in optical WDM mesh networks," *IEEE J. Select. Area Commun.*, vol. 22, no. 9, p. 18591875, November 2004.
- [15] 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.
- [16] B. Ramamurthy, S. Yaragorla, and X. Yang, "Translucent optical WDM networks for the next-generation backbone networks," in *Proc. GLOBE-COM*, vol. 1, November 2001, pp. 60–64.
- [17] M. Zhang, L. Wang, and P. Ye, "All-optical xor logic gates: Technologies and experiment demonstrations," *IEEE Commun. Mag.*, vol. 43, no. 5, pp. 19–24, May 2005.
- [18] 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.
- [19] W. D. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice-Hall PTR, 2004.
- [20] S. Kim, X. Zhang, and S. S. Lumetta, "Rapid and efficient protection for All-optical WDM mesh networks," *IEEE J. Select. Area Commun.*, vol. 25, no. 9, pp. 68–82, December 2007.
- [21] W. He and A. Somani, "Comparison of protection mechanisms: Capacity efficiency and recovery time," in *Proc. ICC*, vol. 1, 2007, pp. 2218–2223.
- [22] G. L. et al., "Experiments in fast restoration using GMPLS in optical/electronic mesh networks," in *Proc. OFC 2001*, Anaheim, CA, March 2001, pp. 2218–2223.
- [23] P. B. et al., "Ultra high capacity optical transmission networks: Final report of action COST 239," Faculty Elect. Eng. Computing, Univ. Zagreb, Zagreb, Croatia, Tech. Rep., 1999.
- [24] C. Mauz, " P -cycle protection in wavelength routed networks," in *Proceedings of the Seventh Working Conference on Optical Network Design and Modelling (ONDM'03)*, Feb. 2007.