Hitless Recovery from Link Failures in Networks with Arbitrary Topology

Serhat Nazim Avci, Xiaodan Hu, 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. To recover from such failures, a number of methods such as SONET rings, protection cycles, and source rerouting have been investigated. Two important considerations in such approaches are the recovery time and the needed spare capacity to complete the recovery. Usually, these techniques attempt to achieve a recovery time less than 50 ms. In this paper we introduce an approach that provides link failure recovery in a hitless manner, or without any appreciable delay. This is achieved by means of a method previously introduced, named diversity coding. We present an algorithm for the design of an overlay network to achieve hitless recovery from single link failures in arbitrary networks via diversity coding. This algorithm is designed to minimize spare capacity for recovery. We compare the spare capacity performance of this algorithm against conventional techniques from the literature via simulations. Based on these results, we conclude that the spare capacity requirements of the proposed technique are favorably comparable to existing techniques, while its recovery time performance is much better, since it can provide hitless recovery.**

I. Introduction

Failures in communication networks are common and can result in substantial losses. For example, in the late 1980s, the AT&T telephone network encountered a number of highly publicized failures [1], [2]. In one case, much of the long distance service along the East Coast of the U.S. was disrupted when a construction crew accidentally severed a major fiber optic cable in New Jersey. As a result, 3.5M call attempts were blocked [1]. On another occasion, of the 148M calls placed during the nine-hour-long period of the failure, only half went through, resulting in tens of millions of dollars worth of collateral damage for AT&T as well as many of its major customers [2].

Observing that such wide-scale network failures can have a huge impact, in February 1992, the Federal Communications Commission (FCC) of the U.S. issued an order requesting that carriers report any major outages affecting more than 50K customers lasting for more than 30 minutes. Over a decade, the reports made available to the public showed that network failures are very common and cause significant service interruptions. According to the publicly available data, while most of the reported events impacted up to 250K users, some impacted millions of users [3].

In the early 1990s, AT&T decided to address the restoration problem for its long distance network with an automatic centrally controlled mesh recovery scheme, called FASTAR, based on digital cross-connect systems [4]. Since then, this subject has seen a significant amount of research. In mesh-based network link failure recovery, the two nodes at the end of the failed link can switch over to spare capacity. Alternatively, all the affected paths could be switched over to spare capacity in a distributed fashion. While the former is faster, the latter will have smaller spare capacity requirement. In this paper, we will use the term source rerouting to refer to mesh-based link or path protection algorithms. In simulations we employ the Simplest Spare Capacity Allocation (SSCA) algorithm [5].

In the mid-1990s, specifications for an automatic protection capability within the Synchronous Optical Networking (SONET) transmission standard were developed. These later became the International Telecommunications Union (ITU) standards G.707 and G.708. The basic idea for protection is to provide 100% redundant capacity on each transmission path through employment of ring structures. SONET can accomplish fast restoration (telephone networks have a goal of restoration within 50 ms after a failure to keep perception of voice quality unchanged by human users) at the expense of a large amount of spare capacity [6], [7]. The restoration times for mesh-based rerouting techniques are typically larger than those of SONET rings, however, the extra transport capacity they require for restoration in the U.S. is generally better than that achievable by SONET rings. In late 1990s, with other major U.S. long distance carriers moving to SONET rings for restoration purposes, an industry-wide debate took place as to whether the mesh-based restoration or the SONET ring-based restoration is better. This debate still continues today. Although most researchers accept that mesh-based restoration may save extra capacity, restoration speeds achievable with mesh-based restoration are generally low and the signaling protocols needed for message feedback are an extra complexity that can also complicate the restoration process.

An extension of the SONET rings is the technique known as p-cycles [8]. In a network, a p-cycle is a ring that goes through all the nodes once. Such a ring will provide protection against any single link failure in the network because there is always an alternative path on the ring that connects the nodes at the end of the failed link, unaffected by the failure. The recovery is carried out by the two nodes that detect the failure at the two ends of the failed link. These nodes reroute the traffic on this link to the corresponding part of the p-cycle. Constructing p-cycles and the corresponding spare capacity

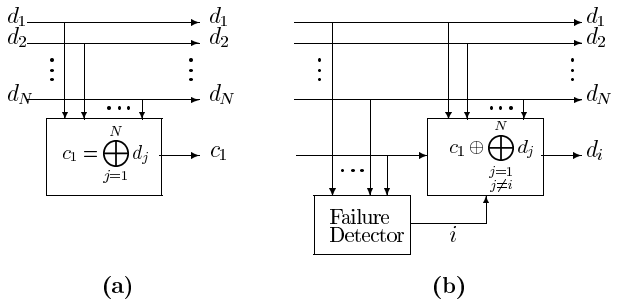

Fig. 1. Diversity coding where N parallel data links are protected against failure by one coded link. (a) Encoder and (b) Decoder.

assignment can be solved by a number of algorithms [8]. Some of these algorithms employ linear programming while there are a number of simpler design algorithms. In this paper, we employ the algorithm in [8, p. 699], which is considered to be within 5% of the optimal solution [8]. We would like to add that in the technique of p-cycles, it is possible to subdivide the network nodes and generate different p-cycle rings for each division separately [8]. We will provide an example of this division in an example in the sequel.

In the case of the Internet Protocol (IP), the restoration time performance is actually significantly worse. It is known that recovery from link failures in IP networks can take a long time [7]. This is becasue IP routing protocols were not designed to minimize network outages. There has been Internet research that shows a single link failure can cause users to experience outages of several minutes even when the underlying network is highly redundant with plenty of spare bandwidth available and with multiple ways to route around the failure [7]. Needless to say, depending on the application, outages of several minutes are not acceptable, for example, for IP telephony, e-commerce, or telemedicine.

Within the telephony transmission and networking community, hitless restoration from failures is often described as an ideal [8]. Nevertheless, with the methods considered, it could not be achieved because these methods are based on message feedback and rerouting, both of which take time. Whereas, with our method, hitless or near-hitless recovery from single link failures becomes possible. The basic technique is powerful enough that it can be extended to other network failures such as multiple link or node failures.

II. Diversity Coding

The basic idea in diversity coding is given in Fig. 1 [9], [10]. Here, digital links of equal rate d_1,d_2,d_3,...,d_N are transmitted over disjoint paths to their destination. For the sake of simplicity, let's assume that these links have a common source and a common destination, and have the same length. A "parity link" c_1 equal to

$$c_1 = d_1 \oplus \ d_2 \oplus \cdots \oplus d_N = \bigoplus_{j=1}^{N} d_j$$

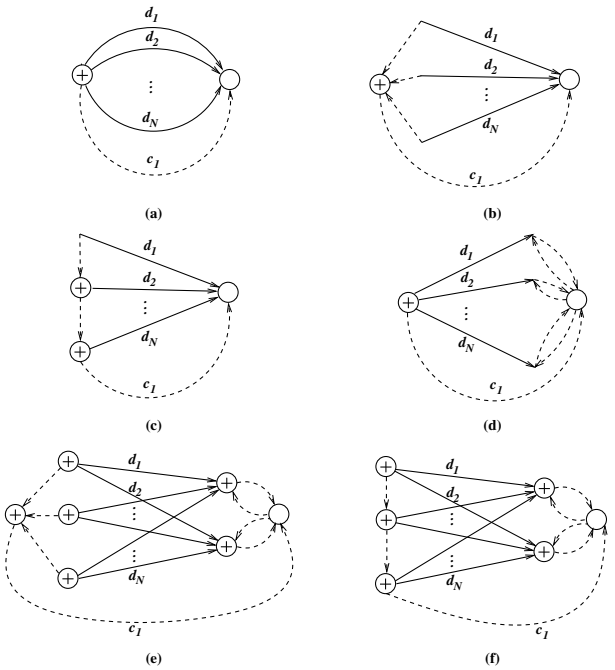

Fig. 2. Network topologies diversity coding can be employed.

is transmitted over another equal length disjoint path. In the case of the failure of link d_i, the receiver can immediately form

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

since it has d_1,d_2,...d_{i-1},d_{i+1},...,d_N available and $d_j \oplus d_j = 0$ in modulo-2 arithmetic or logical XOR operation. As a result d_i is recovered by employing c_1 and $d_1, d_2, \ldots, d_{i-1}, d_{i+1}, \ldots, d_N$. It is important to recognize that this recovery is accomplished in a feedforward fashion, without any message feedback or rerouting.

We assumed above that the sources and the destinations of $d_1, d_2, \ldots, d_N, c_1$ are the same. Diversity coding can actually be extended into network topologies where the source or the destination node is not common. Some examples of such network topologies are provided in Fig. 2. In this figure, solid lines represent actual data links, broken lines represent coded links, a circle with a plus sign in it represents an encoder as in Fig. 1(a), and a circle with blank interior represents a decoder as in Fig. 1(b). Fig. 2(a) represents a point-to-point topology as discussed related to Fig. 1. Fig. 2(b) and Fig. 2(c) are two multipoint-to-point topologies where in the former the encoder is located at a node different than the source nodes while in the latter there is no separate encoder node and the encoding operation is carried out incrementally at each source node. Fig. 2(d) depicts a point-to-multipoint topology, where the decoding is carried out at a node distinct from all the destination nodes. Finally, Fig. 2(e) and Fig. 2(f) are two examples of multipoint-to-multipoint topologies protected against single link failures by diversity coding [9], [10].

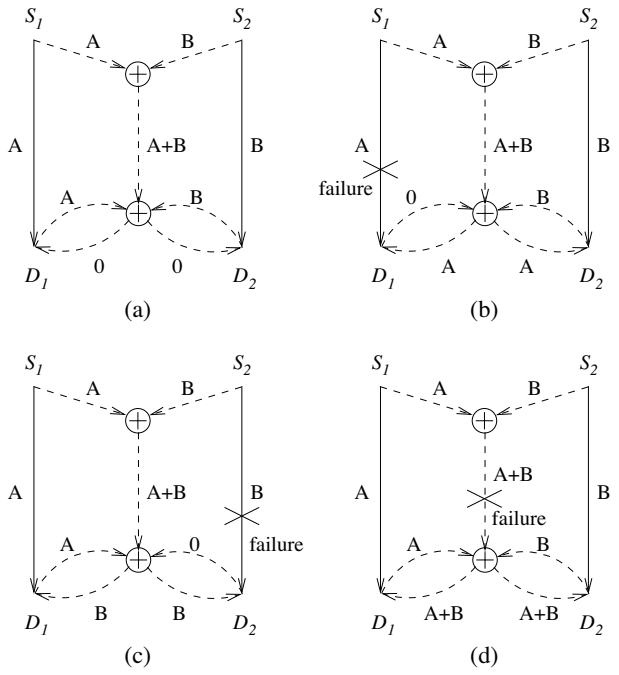

Fig. 3. A simple example for link failure recovery via diversity coding.

Diversity coding papers [9], [10] predate the work that relate the multicast information flow in networks to the minimum cut properties of the network [11] by about a decade. This latter work has given rise to the general area of *network coding*. However, in network coding, discovery of optimal techniques to achieve multiple unicast routing in general networks has remained elusive. In this paper, we provide a heuristic approach to the related problem of designing an overlay network for link failure recovery in arbitrary networks, based on [9], [10].

As stated above, the main advantage of diversity coding as a recovery technique against failures in networks is the fact that it does not need any feedback messaging. Whereas, mesh-based source rerouting techniques, SONET rings, and the technique of p-cycles do need signaling protocols to complete rerouting. With diversity coding, as soon as the failure is detected, the data can be immediately recovered. As in network coding, this requires synchronization of the coded streams. We refer the reader to [9] for a description of the need for synchronization as well as how to achieve it in diversity coding.

A. Example 1

We will now provide a simple example regarding the use of diversity coding for link failure recovery. Consider the network in Fig. 3(a). This network has a similar topology to the well-known butterfly network commonly used to illustrate the basic concept of multicasting via network coding, first appeared in [11]. Also, it can be observed that this network is a special case of the one in Fig. 2(e). In this example, the source node S_1 wishes to transmit its data A to destination node D_1 and the source node S_2 wishes to transmit its data B to destination node D_2, shown by solid lines. The restoration network is shown via dashed lines. There is an encoder on top which forms $A \oplus B$ which we show as $A + B$. This data is then transmitted to the decoder node. The decoder forms the summation of the data received from the encoder and the two destination nodes. In the case of failures, some of these data will not be present. However, the network is designed such that the destination node will automatically receive the missing data from the restoration network in an automatic fashion. In this example, the central decoder does not carry out any failure detection. This task is carried out by the destination nodes D_1 and D_2 as described below.

In the case of regular operation, the destination nodes receive their data from their data links and receive "0" from the restoration network, as shown in Fig. 3(a). Assume the link from S_1 to D_1 carrying data A failed. In this case, both of the nodes D_1 and D_2 receive data A automatically from the restoration network, as shown in Fig. 3. Node D_1 uses this data instead of what it should have been receiving directly from node S_1. Since node D_2 is receiving its regular data B directly from S_2, it ignores the data transmitted by the central decoder. The symmetric failure case for the link from S_2 to D_2 is shown in Fig. 3(c). Other failure scenarios will be ignored by D_1 and D_2 since in those cases they receive their data directly from the respective sources S_1 and S_2. An example of this latter mode of operation is depicted in Fig. 3(d).

B. Example 2

In this example, we will show that the diversity coding can result in less spare capacity than source rerouting or p-cycles. Please refer to Fig. 4(a). This figure shows the available topology of the network, each link is assumed to have the transmission rate equal to one. There are three flows a, b, and c with the same rate of one each. Flows a and b originate from node 1 and terminate on node 3. Flow c originates from node 2 and terminates on node 3. The solution for diversity coding is shown in Fig. 4(b). In this solution, a spare link, *A,* is employed between the nodes 1 and 3. *A* carries $a \oplus b$, which we show simply as $a+b$. The solution also employs two links between nodes 2 and 3. The first one, *B,* carries c, and the second one, *C,* carries $a + c$. The fourth link, *D,* between nodes 1 and 3, carries b. The receiver 3 can always decode a, b, and c under normal operation as well as even if any one of the links in the network failed. We monitor links *A, B,* and *D*. As long as all of them are active, all of a, b, and c can be recovered in normal operation or if the link 1-2 or *C* failed. If one of *A, B,* or *C* fails, as long as that is the only link failure, all of a, b and c can still be recovered. Fig. 4(c) represents the best solution in the case of source rerouting. The upper link is used to protect any failure in transmitting a or b. The second link between 2 and 3 is used to protect flow c. The best solution for p-cycles is given in Fig. 4(d). In this solution, two rings are used to cover all the nodes. The traffic for a, b, and c are all divided into two and transmitted on different links, with capacity $p/2$ being reserved to carry any failed flow $a/2$, $b/2$, or $c/2$, depending on the particular failure. It can be easily

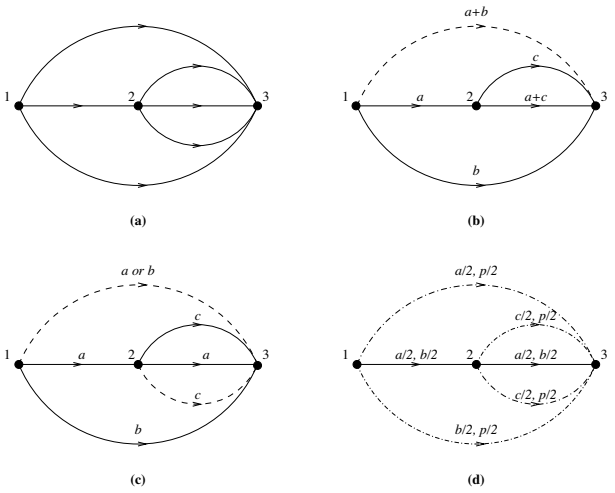

Fig. 4. Spare capacity comparison example.

checked that this solution covers all possibilities of single link failures and guarantees full operation after the failure recovery. Clearly, in this example, both of the approaches of source rerouting and p-cycles result in more spare capacity as compared to the approach of diversity coding.

III. Diversity Coding in Networks with Arbitrary Topology

We will now apply the technique described in the previous section to the design of an overlay network for recovery from link failures in arbitrary networks. We approach this problem by examining all possible combinations of standard diversity coding schemes whose examples are as shown in Fig. 3. In doing this, our goal is to come up with a network for which the spare capacity introduced due to diversity coding is minimized. To that end, we need a measure that will quantify the efficiency of a particular diversity coding combination chosen so that we are able to evaluate a set of such schemes and pick the one that has the best efficiency.

A. Redundancy Ratio Measure

In order to explain this measure, we provide an example diversity coding configuration. Consider the configuration in Fig. 5. This is the same diversity coding configuration discussed in Fig. 3. We will now associate it with a measure to help us evaluate its efficiency so as to be considered in our design algorithm. As previously, the paths $S_1 - D_1$ and $S_2 - D_2$, shown via solid lines, are the working paths, whereas all others are restoration paths. Our goal is to minimize the amount of spare capacity due to restoration paths, given the shortest distance for the working paths. The numbers expressed by X, Y, Z, U, and V in Fig. 5 are the costs associated with the restoration paths. For this paper, we take the cost as the product of the transmission rate on the path and the length of the path.

The measure we introduce will identify the relative redundancy due to a particular diversity coding group. We call this measure the *redundancy ratio*. In order to describe it,

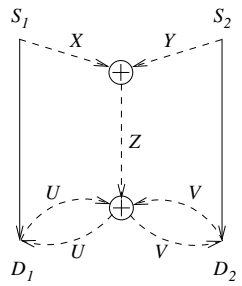

Fig. 5. Network for an example of diversity coding efficiency measure. The letters represent the costs associated with the links.

we need to identify a number of variables. Let N be the number of working paths in the diversity coding group under consideration. Define

p_i: Cost of the shortest path belonging to the i^{th} end-to-end connection in the combination

$p_{total} = \sum_{i=1}^{N} p_i$

w_i: Cost of the working path belonging to the i^{th} end-to-end connection in the combination

$w_{total} = \sum_{i=1}^{N} w_i$

f_i: Rate of the i^{th} end-to-end connection in the combination

s_{total}: Cost of the total spare capacity of the diversity coding combination

As an example, s_{total} for the diversity coding combination shown in Fig. 5 is given as

$$s_{total} = X \cdot f_1 + Y \cdot f_2 + Z \cdot \max(f_1, f_2) + 2 \cdot U \cdot f_1 + 2 \cdot V \cdot f_2.$$

With these variables, we now define the *redundancy ratio* as

$$\begin{aligned} \textit{redundancy ratio} &= \frac{\textit{total capacity} - p_{total}}{p_{total}} \\ &= \frac{w_{total} + s_{total} - p_{total}}{p_{total}}. \end{aligned}$$

B. Proposed Algorithm

We will now discuss how we employ the *redundancy ratio* measure introduced in the previous subsection to designing an efficient diversity coding scheme for a network with arbitrary topology.

The proposed algorithm is intended to search for all possible diversity coding combinations and select those with the smallest redundancy ratio. To that end, we employ a variable called *Threshold.* The threshold begins with a small value (*ThrsdLow*). Diversity coding combinations of N working paths with redundancy ratio values smaller than *Threshold* are accepted, and then *Threshold* is incremented up to its maximum value (*ThrsdHgh.*) Within this process, the value N is decremented from a maximum of N_{max} down to 2. The set of unprotected paths is called the *DemandMatrix,* and when N working paths satisfying the *redundancy ratio* are found, they are taken out of *DemandMatrix.*

At the end, a number of paths may remain uncoded. We protect every such path by a dedicated spare path which carries the same data, known as 1+1 APS (Automatic Protection Switch).

ALGORITHM I: CODE ASSIGNMENT FOR LINK FAILURE RECOVERY VIA DIVERSITY CODING

```
for Threshold=ThrsdLow to ThrsdHgh do
    for all combinations of N = N_max, ...,3,2 do
        if diversity ratio of combination ≤ Threshold
        then
            if flow_1, ..., flow_K ∈ DM then
                for i = 1 to K do
                    DM = DM − {flow_i}
                end
                Update the total, working, and space
                capacities
            end
        end
    end
end
for all flow_k ∈ DM do
    Apply 1+1 APS protection
    DM = DM − flow_k
    Update the total, working, and space capacities
end
```

A description of the algorithm is given under the heading Algoritm I above. In our simulations for this paper, the numerical values used are *ThrsdLow* = 1.6, *ThrsdHgh* = 3.0, and $N_{max} = 4$.

IV. SIMULATION RESULTS

In this section, we will present simulation results for link recovery techniques previously discussed, in terms of their spare capacity requirements. We employ the following formula for calculating the spare capacity percentage (scp) in all simulations.

$$scp = \frac{(total\ capacity - shortest\ working\ capacity)}{shortest\ working\ capacity}.$$

Shortest working capacity refers to the total capacity when there is no recovery technique and traffic is routed through the shortest working paths. Total capacity is the capacity that results when spare and working routes are jointly calculated. Then, the relative spare capacity percentage is calculated as described above. We will now provide the simulated spare capacity percentage values for a number of representative networks.

The first network studied is the European COST 239 [12] network whose topology is given in Figure 6. In this graph as well as the others in the sequel, the numbers associated with the nodes represent a node index, while the numbers associated with the edges correspond to the distance associated with the edge. The traffic demand is adopted from [12] and applied to the simulation. This network was previously studied in the context of link failure recovery [8]. We provide the spare capacity percentage results for the three schemes in Table I.

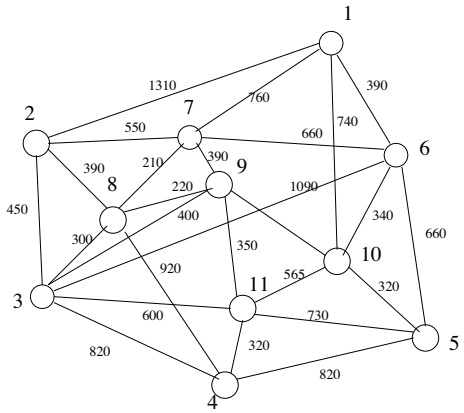

Fig. 6. European COST 239 network.

TABLE I
SIMULATION RESULTS OF COST 239 NETWORK

COST 239 Network, 11 nodes, 26 spans	
Protection Scheme	Spare Capacity Percentage
Diversity Coding	98%
Source Rerouting	90%
p-cycles	64%

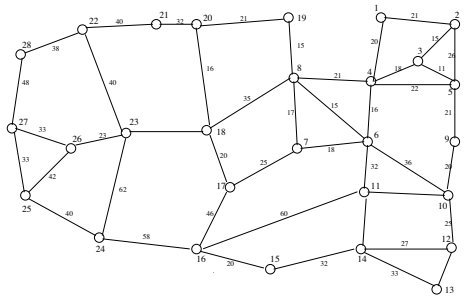

Fig. 7. U.S. long distance network.

TABLE II
SIMULATION RESULTS OF U.S. LONG DISTANCE NETWORK

US Long Distance Network, 28 nodes, 45 spans	
Protection Scheme	Spare Capacity Percentage
Diversity Coding	106%
Source Rerouting	91%
p-cycles	107%

The second network is based on the US long-haul optical network. The topology of this network is shown in Figure 7. It is based on the topology given in [5]. In order to calculate the traffic, we employed a gravity-based model [13] and assumed the traffic between two node is directly proportional to the product of the populations of these nodes. We assigned cities corresponding to each node in the network and calculated the approximate population that each node represents. The spare capacity percentage values for this network are given in Table II.

Finally, we would like to provide simulation results for

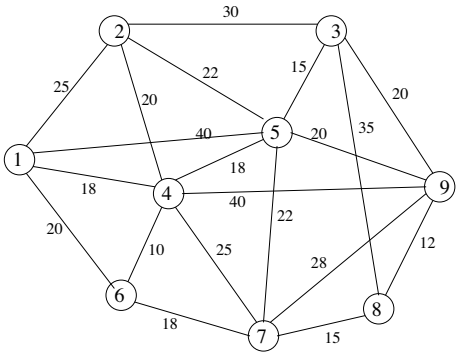

Fig. 8. Synthetic network.

TABLE III
DEMAND MATRIX FOR THE SYNTHETIC NETWORK

	1	2	3	4	5	6	7	8	9
1	0	10	30	10	10	10	20	10	50
2	10	0	3	1	1	1	2	1	5
3	30	3	0	3	3	3	6	3	15
4	10	1	3	0	1	1	2	1	5
5	10	1	3	1	0	1	2	1	5
6	10	1	3	1	1	0	2	1	5
7	20	2	6	2	2	2	0	2	10
8	10	1	3	1	1	1	2	0	5
9	50	5	15	5	5	5	10	5	0

TABLE IV
SIMULATION RESULTS OF SYNTHETIC NETWORK

Synthetic Network, 9 nodes, 20 spans	
Protection Scheme	Spare Capacity Percentage
Diversity Coding	81%
Source Rerouting	100%
p-cycles	85%

a network that favors diversity coding over the other two approaches. We came up with this network in order to provide a different example than the two previous networks. The topology of this network is given in Fig. 8. The demand matrix for it is specified in Table III. The spare capacity percentage results are provided in Table IV. For this network, the best spare capacity results are obtained by employing the diversity coding approach, similarly to the case we showed in the example in the Section II.B.

V. CONCLUSION

In this paper, we employed the technique of diversity coding [9], [10] to providing a single link failure recovery technique in networks with arbitrary topologies. This is accomplished by finding groups of links that can be combined in basic diversity coding topologies, or in other words, mapping the arbitrary topology into efficient groups of basic diversity coding topologies. This approach results in link failure restoration schemes that do not require message feedback or rerouting and therefore are extremely efficient in terms of their restoration speed. In most cases, immediate recovery after the failure will be possible and therefore the technique can be considered *hitless,* an ideal goal in communication networks that has so far remained unaccomplished.

We have shown via examples and simulation results that the networks designed using the technique proposed remain competitive in terms of the spare capacity requirements they dictate when compared to standard techniques from the literature. All of the techniques employed in simulations are based on shortest distance algorithms. Although better performing or optimum techniques in each case are a possibility, the important conclusion is that hitless recovery via diversity coding with competitive spare capacity against standards network restoration techniques is possible. We have provided examples where diversity coding even provides the best spare capacity requirement.

Although the diversity coding work [9], [10] predates the work that gave rise to the currently popular field of network coding [11], it can be argued that what is provided in this paper falls within the application areas of network coding. As in network coding as well as what is discussed in [9], [10], erasure coding techniques can be employed to extend this technique to recovery from more than one link and node failures.

REFERENCES

[1] J. C. McDonald, "Public network integrity - Avoiding a crisis in trust," *IEEE J. Sel. Areas Commun.*, vol. 12, pp. 5–12, January 1994.
[2] "Ghost in the machine," *Time Magazine*, January 29, 1990.
[3] A. P. Snow and D. Straub, "Collateral damage from anticipated or real disasters: Skewed perceptions of system and business continuity risk," in *Proc. IEEE International Engineering Management Conference*, vol. 2, September 2005, pp. 740–744.
[4] C.-W. Chao, P. M. Dollard, J. E. Weythman, L. T. Nguyen, and H. Eslambolchi, "FASTAR - A robust system for fast DS3 restoration," in *Proc. IEEE Global Communications Conference*, December 1991, pp. 1396–1400.
[5] Y. Xiong and L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Trans. Netw.*, vol. 7, pp. 98–110, February 1999.
[6] S. V. Kartalopoulos, *Understanding SONET/SDH and ATM: Communications Networks for the Next Millenium*. Wiley-IEEE Press, 1999.
[7] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Elsevier, 2004.
[8] W. D. Grover, *Mesh-Based Survivable Networks: Options and Strategies for Optical, MPLS, SONET, and ATM Networking*. Prentice-Hall PTR, 2004.
[9] 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.
[10] ——, "Diversity coding for transparent self-healing and fault-tolerant communication networks," *IEEE Trans. Commun.*, vol. 41, pp. 1677–1686, November 1993.
[11] 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.
[12] P. Batchelor and *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.
[13] Y. Zhang, M. Roughan, N. Duffield, and A. Greenberg, "Fast accurate computation of large-scale IP traffic matrices from link loads," in *Proc. ACM SIGMETRICS*, June 2003.