A Generalized Strategy for 1+N Protection

Ahmed E. Kamal
Electrical and Computer Eng. Dept., Iowa State University, Ames, IA 50011, U.S.A.
E-mail: kamal@iastate.edu

***Abstract*—1+N protection was proposed by the author in [1], where a single circuit is used to protect a number of connections. For this purpose, network coding was used to carry a linear combination of the signals, and p-Cycles were used to provide this combination to the destination nodes in order to allow them to extract a second copy of the transmitted signal. In this paper, we introduce a generalized form of 1+N protection. Network coding is used to combine a number of signals on one protection circuit. However, p-Cycles and not used, and the signals are combined on a protection circuit, which is optimally selected to minimize the amount of protection resources. Moreover, and unlike the strategy in [1], the generalized strategy does not require symmetry in resource reservation in the two directions of communication, but the protection resources are provisioned based on need. The strategy introduced in this paper provides 100% protection against single link failure on any of the working paths. A performance comparison between the cost of implementing the proposed scheme and 1+1 protection is provided, and is based on Integer Linear Programming formulations.**

I. Introduction

A large number of techniques for providing optical network survivability have been introduced. Such techniques can be classified as either *Predesigned Protection*, or *Dynamic Restoration* techniques [2]. In predesigned protection, which is a proactive technique, bandwidth is reserved in advance so that when a failure takes place, backup paths which are pre-provisioned, are used to reroute the traffic affected by the failure. These techniques include the 1+1 protection, in which traffic of a lightpath is transmitted on two link disjoint paths, and the receiver selects the stronger of the two signals; 1:1 protection, which is similar to 1+1, except that traffic is not transmitted on the backup path until failure takes place; and 1:N protection, which is similar to 1:1, except that one path is used to protect N paths. The p-Cycle concept [3] is one way of implementing 1:N protection which is close to optimal 1:N protection, especially when the network graph is dense. A generalization of 1:N is the M:N, where M protection paths are used to protect N working paths. Protection techniques are widely used in SONET ring networks [2]. Under dynamic restoration, which is a reactive strategy, capacity is not reserved in advance, but when a failure occurs spare capacity is discovered, and is used to reroute the traffic affected by the failure. Protection techniques can recover from failures quickly, but require significant amounts of resources. On the other hand, restoration techniques are more cost efficient, but are much slower than their protection counterparts.

Recently, the author introduced another new concept for protection, namely, 1+N protection in [1]. The technique is based on using a bidirectional p-Cycle to protect a number of link disjoint connections which are straddling from the cycle, and using network coding [4] to transmit modulo-2 sums of the connections' signals on the cycle. A failure of any link on a working path can be recovered from by using a decoding operation of the signals transmitted on the p-Cycle. This strategy was introduced to provide 100% protection against single link failures. The 1+N protection can be implemented at a number of layers, and using a number of protocols.

This paper to introduces a general strategy for providing 100% 1+N protection against single link failures in mesh networks, and without using p-Cycles. That is, to transmit signals from N connections on one common circuit, such that when a failure occurs, the end nodes of the connection affected by the failure will be able to recover the signals lost due to failure. This is done by combining signals from a number of connections using the technique of network coding, and transmitting this combination on the backup circuit. Hence, survivability is provided without explicitly detecting failures, and rerouting of the signal is not needed. Both the management and control planes in this case will be simpler. In addition to protection, and as a byproduct, in the absence of failures, this scheme provides an error correction functionality, where a data unit corrupted on the working circuit can be recovered from the protection circuit.

The rest of the paper is organized as follows. In Section II we introduce the network model, and a few definitions and assumptions. We illustrate the basic concept of our strategy through an example in Section III, which is then followed by the description of the general strategy. The cost of implementing the proposed strategy is compared to the cost of implementing 1+1 protection in Section IV. This is based on an Integer Linear Program (ILP) formulation for optimally protecting a group of connections in a network using the proposed scheme. Due to the lack of space, the ILP formulation is not shown in the paper. Finally, the paper is concluded with some remarks in Section V.

This research was partially supported by grants CNS-0626741 and CNS-0721453 from NSF, and a gift from Cisco Systems.

II. Definitions and Assumptions

In this section we introduce some preliminaries.

- The network is represented by the graph $G(V, E)$, where V is the set of nodes, and E is the set of bidirectional edges in the graph. For the network to be protected, we assume that the graph is at least 2-connected, i.e., between any pair of nodes, there is at least two link-disjoint paths. Following the terminology in [3], we refer to an edge in the graph as a *span*. A span between two nodes contains a number of channels. The type and number of channels depends on the type of the span, and also on the layer at which the connection is provisioned, and protection is provided. We refer to each of these channels as a *link*.
- There is a set C of unicast connections that need to be provisioned in the network such that 100% 1+N protection is guaranteed. The total number of connections is given by $N = |C|$. It is assumed that all connections require the same bandwidth, B, and this bandwidth is allocated in terms of a circuit on a single link, i.e., single hop, or may consist of a sequential set of circuits on multiple sequential links, i.e., multihop.
- Connections are unidirectional, and a connection c_j from source S_j to destination D_j will transmit data units $d_j^{(n)}$, where n is the sequence number, or round number in which the data unit is transmitted. Connection $c_j \in C$ is identified by the tuple $< S_j, D_j, d_j^{(n)} >$. A bidirectional connection will be treated as two independent unidirectional connections.
- All data units sizes are fixed and equal.
- It may not be possible to protect all N connections together. In this case, the set of connections, C, is partitioned into K subsets of connections, C_i for $1 \leq i \leq K$, where set C_i consists of $N_i = |C_i|$ connections, such that $\sum_{i=1}^{K} N_i = N$.
- The scheme presented in this paper is designed to protect against a single link failure.
- When a link carrying active circuits fails, the tail node of the link will receive empty data units, which can be regarded as zero data units.

It should be pointed out that all addition operations (+) in this paper as **modulo two additions**, i.e., Exclusive-OR (XOR) operations.

III. Generalized 1+N Protection

In this section we introduce the Generalized 1+N Protection for guaranteed protection against single link failures. We first illustrate the basic principles of this scheme using an example, and then present the general scheme, including the operation at different nodes in the network. We also show how to handle the special case of a number of connections with the same destination in order to further reduce resources.

A. Motivation and Basic Principles

In 1:N protection, a backup path is used to protect one of N link disjoint working paths if one working path fails. In this case, if a working path fails, the failure must be detected, and then the failed working path signal can be routed on the protection path. Our objective is to avoid the operations of *failure detection*, which is performed by the management plane, and *rerouting*, which is done by the control plane, and allow all sources to transmit backup copies to their respective destinations, simultaneously and on the same protection circuit. However, signals from the N connections cannot be transmitted simultaneously on the protection path since this will result in contention and collisions. Therefore, the signals are transmitted on the protection path, after being linearly combined using network coding. For example, the signals are added using addition on $\mathbb{GF}(2)$, i.e., XORed, as shown in Figure 1.(a). We refer to this protection path as the *primary protection circuit*. However, when a working path fails, the sum of the signals, which is received on the primary protection circuit, is not sufficient to recover the signal transmitted on the failed working path. For example, in Figure 1.(a), when working path 2 fails, node D_2, which is the receiver at the end of path 2, receives $d_1 + d_2 + d_3$ on the primary protection circuit, where the sum is modulo 2. Node D_2 cannot recover d_2 from this sum. We solve this problem by having all received signals added at the receiver side, and delivered to all receivers on a second protection circuit, that we refer to as the *secondary protection circuit* (see Figure 1.(b)). These two signals can be used to recover the signal transmitted on the failed path. In the example of Figure 1.(c), which includes both *primary* and *secondary* protection circuits, when working path 2 fails, then D_2 receives:

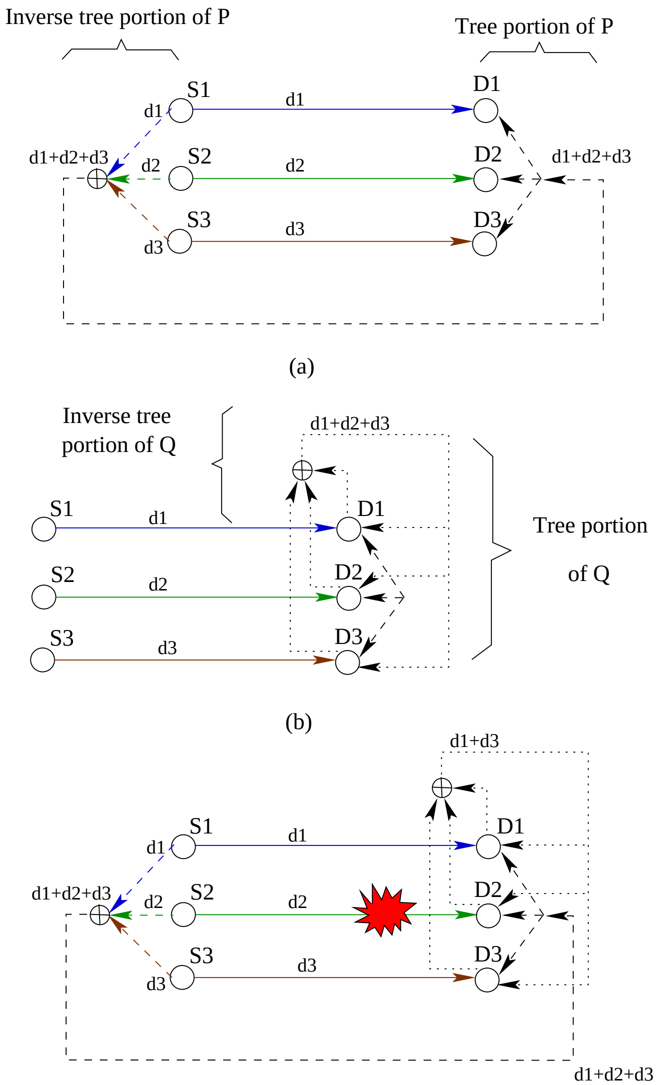
- $d_1 + d_2 + d_3$ on the *primary protection circuit*, and
- $d_1 + d_3$ on the *secondary protection circuit*.

These two sums are added by D_2 to recover the lost signal, d_2.

B. Generalized 1+N Protection Against A Single Failure

For each subset of connections, C_i, that are to be protected together, three types of circuits are provisioned:

- A total of N_i link disjoint working paths are provisioned to carry the signals directly from source S_j to destination D_j, for all connections $c_j \in C_i$. The working path for connection c_j is denoted by W_j. Each path has a bandwidth B, and data units are transmitted from S_j to D_j in rounds.
- A *primary protection circuit*, P_i, is provisioned for all connections in C_i, and is used to deliver the sum of all data units, d_j, transmitted by the sources, S_j, where $c_j \in C_i$, to all receivers, D_j in C_i. P_i is link disjoint from the working paths in C_i. P_i consists of N_i **shared multicast trees** from each source, S_j in



(a)

(b)

(c)

Fig. 1. An illustration of Generalized 1+N protection: (a) the primary protection circuit; (b) the secondary protection circuit; (c) both protection circuits providing data recovery from a failure of path W_2.

C_i to all destinations D_k in C_i. The *sharing* of the multicast trees implies that when m trees share a link, the bandwidth required on that link is still B, and not $m{\cdot}B$. This is because the m data units to be transmitted on the shared link are linearly combined together using XOR operations before transmission and only the sum is transmitted.

The primary protection circuit, P_i, is therefore used to deliver the following to all nodes D_j, $c_j \in C_i$, where the sum is modulo 2.

$$\sum_{c_l \in C_i} d_l \tag{1}$$

The example of Figure 1.(a) shows an implementation of P_i as an inverse tree connected to a tree (this implementation of shared multicast trees may not be optimal, but is shown here for the sake of example only). The inverse tree is used to collect the signals from all sources S_j in C_i, which is connected to a tree that delivers the sum of these signals to the destinations D_j in C_i. At every merging point on the inverse tree, data units transmitted by S_j are added, and are transmitted on the outgoing link. At every branching point on the tree, received sums of data units are transmitted on all outgoing links to all nodes D_j, $c_j \in C_i$.

- There is also a *secondary protection circuit* for C_i, which we refer to as Q_i, which is also implemented as shared multicast trees from each destination, D_j, to all destinations in C_i, including D_j itself. This circuit collects data units received by D_j nodes, and sums these data units using modulo-2 addition and delivers the sum to all destinations. While Q_i needs to be link disjoint from all working paths in C_i, it need not be link disjoint from P_i. Since Q_i delivers the sum of received signals to all receiver nodes in C_i, if, W_k, the working path of connection $c_k \in C_i$ fails, then the signal delivered on Q_i to D_j for $c_j \in C_i$ will be

$$\sum_{c_l \in C_i, c_l \neq c_k} d_j \tag{2}$$

In this case, node k can recover d_k by adding equations (1) and (2).

The example in Figure 1.(c) shows an implementation of Q_i which also has the form of an inverse tree connected to a tree, and collects and adds the received signals from D_j in C_i, and delivers this sum to the D_j nodes in C_i. Again, this may not be an optimal implementation of Q_i, but is only shown for the sake of exposition.

On all three types of circuits above, data units are transmitted in rounds, such that only data units generated in round n are added together on P_i and Q_i. As mentioned above, the data unit transmitted from node S_j to node D_j in round n will be denoted by $d_j^{(n)}$.

Below, we describe the operations performed by all nodes, the source, S_j, the destination, D_j, and intermediate nodes on the primary protection circuit P_j and secondary protection circuit, Q_j.

Role of Node S_j of connection $c_j \in C_i$:
Node S_j will take the following actions:

- Transmit $d_j^{(n)}$ on the working path W_j to D_j in round n.
- Add $d_j^{(n)}$ to the round n data received on the incoming link from P_i, if any, and transmit on the outgoing link(s) P_i.

  Note that this step is necessary since outgoing links of the tree rooted at S_j may be shared by another tree rooted at another node, S_k in C_i.

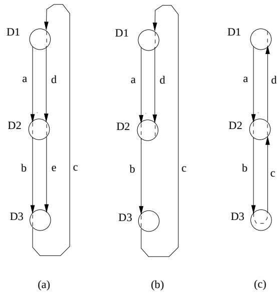


Fig. 2. An example to show *collector* and *delivery* links

Notice that the S_j nodes will have to be synchronized to transmit data units in the same round. Synchronization can be relaxed, and implemented by buffering at one or more nodes. That is, a node that has to perform an addition operation on a number of data units in round n before transmitting their sum, will have to buffer received data units until all data units are available.

Role of Node D_j of connection $c_j \in C_i$
Before describing the operations performed by node D_j, we need to identify two types of incoming links on the Q_i circuit. Some incoming links will be part of the data collection circuit in the shared multicast trees, and these are referred to as *collector links*. These links will not be carrying the sum of all $d_k^{(n)}$, for all $c_k \in C_i$. Other links will be part of the data delivery circuit in the shared multicast trees, and these will be called *delivery links*. These are links which carry $\sum_{c_k \in C_i} d_k^{(n)}$. For example, in Figure 2.(a) the Q_i circuit is implemented as a non-simple path. Span (D_1, D_2) carries two links as part of the Q_i circuit, and so does span (D_2, D_3). However, for each such pair of links one is a *collector* link and the other is a *delivery link*. Links a and b are *collector links*, while links c, d and e are *delivery links*. It should be also noted that the Q_i circuit can be implemented using a fewer number of links, as shown in Figures 2.(b) and 2.(c), depending on link and bandwidth availability. In this case, node D_3 which determines that the combination that it outputs contains all data units in the protected group, need not receive this combination again.

Based on the above definitions, we now define the actions taken by node D_j:

- If node D_j has any outgoing link on the P_i circuit, then any data unit received on an incoming link on P_i will be transmitted on all outgoing links on the P_i circuit.
- Node D_j will receive data transmitted on the working path W_j from S_j in round n, $d_j^{(n)}$. Call this received data unit $d_j^{(n)'}$. In the case of failure of W_j, $d_j^{(n)}$ will not be received, and therefore $d_j^{(n)'}$ will be taken as zero for the purpose of recovery of the lost $d_j^{(n)}$.
- Node D_j will add $d_j^{(n)'}$ to the round n data units received on the incoming *collector* links of Q_i, if any. If there are outgoing *collector* links, the sum will be transmitted on them. If there are no outgoing *collector* links of Q_i, but there are outgoing *delivery* links, the sum will be transmitted on them.
  In the example of Figure 2, node D_2 receives $d_1^{(n)'}$ on incoming *collector* link a, adds $d_2^{(n)'}$ using modulo-2 addition, and since the outgoing link b is a *collector* link, the sum is transmitted on b. However, for node D_3, $d_3^{(n)'}$ will be added to the $d_1^{(n)'} + d_2^{(n)'}$ received on incoming *collector* link b, and the sum will be transmitted on the outgoing delivery link c, since there are no outgoing *collector* links.
- Round n data units received by node D_j on incoming *delivery* links of Q_i will be added to the round n data units received on the P_i circuit. Call this sum $d_j^{(n)''}$. The outcome will depend on $d_j^{(n)'}$:
  - In case $d_j^{(n)'} = 0$, i.e., the W_j working path has failed, then $d_j^{(n)''} = d_j^{(n)}$.
  - In case $d_j^{(n)'} \neq 0$, i.e., the W_j working path has not failed, then $d_j^{(n)''}$ should be 0 in the case of no other failures. However, if $d_j^{(n)''} \neq 0$, this means that either a failure on another working path, or on a protection path has taken place, and node D_j should ignore this signal.

Role of intermediate nodes on P_i and Q_i
Intermediate nodes on P_i and Q_i may either have one, or more incoming links on the same circuit. Therefore, intermediate nodes will add received data units on all incoming links in the same round, $d_j^{(n)}$, and forward them on all outgoing links.

C. Connections with a Common Destination

If a set of connections, which are jointly protected have the same destination, then the secondary protection path is not needed. This is true since if the number of jointly protected connections is m, then if one of the working paths fails, the destination will receive exactly m signals which correspond to linearly independent equations, one of which arrives on the primary protection circuit. Using these independent equations, data units transmitted on the failed working path can be recovered.

The above scheme can be adopted without change, except for doing away with the *secondary protection circuit*, hence achieving further saving in protection resources. This requires that there be m link disjoint paths which are used as working paths. In addition, the shared trees of the *primary protection circuit* should be link disjoint with all those m paths. In the ILP formulation that we developed for the purpose of cost evaluation, this case has been taken into consideration.

IV. Implementation Cost and Comparison

To provision working and protection circuits, link disjoint paths need to be found. The problem of finding link disjoint paths between pairs of nodes in a graph is known to be an NP-complete problem [5]. Hence, even finding the working paths in this problem is hard. We have therefore developed an ILP for solving the optimal Generalized 1+N protection problem introduced in this paper. It is to be noted that the solution is optimal under the given proposed strategy. Due to space limitations, the ILP is not shown in the paper, but it was used to assess the cost of implementing the proposed scheme, and to compare it to 1+1 protection. For the 1+1 protection, the cost is based on an optimal ILP formulation similar to that in [6][1]. The ILPs were solved using the Cplex 10.1.0 solver. Due to the complexity of the ILP formulation of the Generalized 1+N protection, we were able to only consider limited size networks. Moreover, several of the results were obtained by terminating the runs when a gap of 20% was achieved. These are indicated by a * in Table I.

We considered a baseline network with 6 nodes and 12 edges, and hence the nodal degree is 3. We also considered two other networks to compare them to the baseline one: a larger network in terms of nodes and edges, but with the same graph density, and a network with the same number of nodes but with more edges, hence increasing the graph density to 4. With each network, a certain number of connections were randomly generated, and provisioned such that 100% protection against single link failures was guaranteed, using 1+1 protection, and the scheme of this paper.

First, it should be noted that in the ILP formulation, constraints were included to use the shortest possible working paths if this does not result in increasing the overall cost. It can be observed from both tables that the use of 1+N protection has not necessarily resulted in using the shortest working paths. However, it has resulted in reducing the overall resource cost. For network A in Table I, when $N = 6$ and $E = 9$, i.e., an average nodal degree of 3, the Generalized 1+N protection achieved a saving of up to 6% over 1+1 protection. Increasing the network size, in terms of the number of nodes and edges to $N = 8$ and $E = 12$, while keeping the nodal degree equal to 3, which is network B in Table I has also achieved a saving of about 5%. The saving, however, may be more than that since CPLEX was stopped with a gap of 20%. More experiments need to be performed in order to quantify the real savings. When the nodal degree was increased to 4, which is the case for network C in Table I, a greater saving, reaching 12% was achieved. This is due to the fact that more link disjoint alternate routes are available. The other thing to notice is that the increase in the nodal degree also results in reducing the lengths of the working paths, hence reducing the network delay. This means that this technique is more effective in networks with high nodal degrees, such as NJ-LATA and the Pan-European COST239 network.

TABLE I
Cost comparison between 1+1 and 1+N protection

Network	N, E	connections	1+1	Generalized 1+N
A	6, 9	6	21 (8, 13)	20 (9, 11)
		8	30 (11, 19)	29 (14, 15)
		10	34 (13, 21)	32 (14, 18)
B	8, 12	8	33 (13, 20)	33 (15, 18)
		12	54 (22, 32)	52 (24,28)*
		16	71 (27, 44)	68 (30, 38)*
C	6, 12	6	19 (7, 12)	18 (8, 10)
		8	26 (10, 16)	23 (12, 11)
		10	32 (12, 20)	28 (13, 15)

A full quantification of the savings and performance trends of the Generalized 1+N protection technique is the subject of a future study.

V. Conclusions

This paper has introduced a generalized strategy for 1+N protection. The strategy uses network coding to protect a set of unidirectional connections, which are provisioned using link disjoint paths. Network coding is used on a primary protection circuit to combine signals transmitted by the sources, and is also used on a secondary protection circuit to combine signals received by the destinations. The linear combinations are based on simple modulo-2 additions, or XOR operations. The availability of these two combinations allows the destination of a failed working path to recover the lost data units /bin/bash: a: command not found Numerical examples based on optimal formulations were introduced and showed that the resources consumed by this strategy are less than those needed by 1+1 strategies. The advantages of this scheme is the sharing of protection resources in a manner that enables the recovery of lost data units at a speed that is comparable to that of 1+1 protection, but using protection resources at the level of 1:N protection. This sharing was enabled through the use of network coding.

References

[1] A. E. Kamal, "1+n protection in optical mesh networks using network coding on p-cycles," in *in the proceedings of the IEEE Globecom*, 2006.

[2] D. Zhou and S. Subramaniam, "Survivability in optical networks," *IEEE Network*, vol. 14, pp. 16–23, Nov./Dec. 2000.

[3] W. D. Grover, *Mesh-based survivable networks : options and strategies for optical, MPLS, SONET, and ATM Networking.* Upper Saddle River, NJ: Prentice-Hall, 2004.

[4] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, pp. 1204–1216, July 2000.

[5] J. Vygen, "Np-completeness of some edge-disjoint paths problems," *Discrete Appl. Math.*, vol. 61, pp. 83–90, 1995.

[1] A polynomial time algorithm like Bhandari's algorithm may also be used.

[6] C. Mauz, “Unified ilp formulation of protection in mesh networks,” in *7th Intl. Conf. on Telecomm. (ConTEL)*, pp. 737–741, 2003.