Efficient and Agile 1+N Protection

Ahmed E. Kamal Osameh Al-Kofahi

***Abstract*—This paper introduces an efficient implementation of the network coding-based 1+N protection. The strategy provides proactive protection to N link-disjoint full-duplex connections against single link failures. The implementation is efficient and uses a tree shaped minimum cost protection circuit. The protection circuit carries linear combinations of data units originally transmitted on the working circuits, and these linear combinations can be used to recover data units lost due to failures. This recovery is carried out with the assistance of one node on the protection tree, which is chosen to reduce the recovery time. This protection technique requires the same amount of protection resources used by 1:N protection, where the protection circuit is link disjoint from the protected connections.**

The paper also makes other contributions. It introduces an Integer Linear Program (ILP) formulation to evaluate the cost of protection using this technique, and compares it to the cost of 1+1 protection. The comparison shows that a significant saving in cost can be achieved, while recovering from failures within a short time. The performance of this scheme is further evaluated using an OPNET-based simulation, where it was shown that the recovery time conforms to acceptable industry standards. Availability analysis is also conducted.

I. Introduction

One of the important operational requirements of networks is to provide uninterrupted service in the face of failures. This is usually known as *network survivability* or *network resilience*, and network service providers consider this requirement to be one of the key requirements that is usually demanded by customers. To provide survivability, and recover from failures, two steps are involved: detection, and localization of failures, which is executed by the management plane; and rerouting of data, which is done by the control plane. A widely accepted upper bound on the total recovery time from failures is 50 milliseconds [3][1], since this does not trigger any alarms at higher level protocols, e.g., TCP. Different protocols for implementing the self-healing functionality are designed and implemented to meet this standard. However, the cost of implementation can be different for different strategies.

Depending on the type of the network, and the technology employed therein, failures may be more frequent, and even more catastrophic for one type of networks as compared to other types of networks. For example, in optical fiber-based networks, most failures are single failures, and multiple failures are very rare. However, the failure of a single fiber can affect a large number of users and connections, since fibers carry huge amounts of traffic, especially if dense wavelength division multiplexing (DWDM) is used. Hence, it is very important to provide a high degree of survivable network operation in the face of failures in optical communication networks.

Survivability has been an active area of research for a number of years, and several techniques for providing survivable operations, especially in optical networks, have been introduced. These techniques can be regarded as either *Predesigned Protection*, or *Dynamic Restoration* techniques [1]. In predesigned protection, bandwidth on backup circuits is reserved in advance so that when a failure takes place, backup paths which are reserved in advance, are used to reroute the traffic lost due to failure. These techniques include the 1+1 protection, in which traffic is transmitted on two link disjoint paths simultaneously. If the working path fails, or becomes noisy, the receiver then switches to the signal on the backup path. They also include 1:1 protection, which is similar to 1+1, but traffic is not transmitted on the backup path until after a failure is detected. 1:N protection is an extension of 1:1 in which one backup path is used to protect N working paths. M:N is an even more general extension, where M protection paths are used to protect N working paths. Note that 1+1 is faster than 1:1, or its extensions, since it does not require detecting the failure by the sources, reconfiguring the switches, or rerouting the traffic. However, sharing the protection resources (the 1 path in 1:N, and the M paths in M:N) makes 1:N and M:N more efficient in utilizing the network resources when compared to 1+1. In the dynamic restoration strategy, no backup capacity is reserved in advance. Therefore, upon the occurrence of a failure, spare capacity in the network is discovered, and is used to reroute the traffic affected by the failure. Protection techniques are faster than dynamic restoration techniques, since the spare capacity discovery phase is bypassed. However, they require the reservation of significant amounts of backup resources. The spare capacity exploration phase makes dynamic restoration techniques slower than protection techniques. Nonetheless, dynamic restoration is more cost efficient.

Motivated by the savings in backup resources achieved by extending 1:1 to 1:N, the first author introduced 1+N protection in [2]. The proposed mechanism protects a number of link-disjoint full-duplex connections that have their terminal nodes (source and destination) on a bidirectional p-Cycle [3]. By using network coding [4], the bidirectional p-Cycle is used to carry redundant linear combinations of the data units that are forwarded on the protected connections in opposite directions. These linear combinations are simply the modulo-2 sum (bitwise XOR) of the data units sent and received on the protected connections. If a failure occurs, a terminal node is guaranteed to receive enough combinations to recover the data unit destined to it. This strategy was extended to protect against multiple failures in [5]. Also, a simpler implementation of the same strategy, which uses a protection path, instead

A preliminary version of this paper was presented at the Allerton 2008 conference. This research was supported in part by grants CNS-0626741 and CNS-0721453 from the National Science Foundation, and a gift from Cisco Systems.

[1] See Chapter 3 in [3] for a discussion about the 50 millisecond recovery time.

of a p-Cycle, was introduced in [6]. The use of network coding to protect Wireless Mesh Networks, which use many-to-one service, was introduced by the authors in [7]. Network coding was also used to recover lost packets in Wireless Sensor Networks in [8].

We note that using network coding to provide protection of unidirectional connections against failures has been first introduced in [9] for directed acyclic graphs. However, it was shown by example that it is not always feasible to provide static network coding that can protect against all single failures. In this paper we introduce a strategy that uses static coding in order to provide protection of bidirectional connections (including unidirectional connections as a special case). However, this entails the use of an undirected subgraph, that takes the form of a tree, which is link disjoint from the primary connections. This subgraph is used as a protection circuit, and cycles are embedded on this subgraph. The proposed mechanism has exactly the same cost as the 1:N protection technique in terms of backup resources, when the protection circuit is link disjoint of the protected connections. In contrast to the scheme in [2], and instead of solving the linear combinations in order to recover lost data units at the receivers only, the receivers together with one intermediate node cooperate in order to recover the lost data. It is worth mentioning that although the cost of implementation is exactly the same as that of implementing 1:N protection, the time to recover from failures is much shorter, and is comparable to that of 1+1 protection since no fault detection, fault localization, switch reconfiguration or rerouting is required. The scheme has the following properties:

1) Protection against single link (or connection) failure is guaranteed.
2) The scheme can be provisioned to protect either unidirectional or bidirectional connections.
3) 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[2].

It is to be noted that although both the schemes proposed in this paper and that in [2] use network coding to provide protection against single link failures, they are fundamentally different in a number of aspects. In the strategy in [2] only end nodes of connections are involved in the coding process. In this paper, intermediate nodes in the network might be involved. This makes the strategy in [2] simpler in terms of the required computations. However, the simplicity of [2] compromises optimality in terms of resource consumption, as the scheme introduced in this paper is more resource efficient since the protection circuit is the same one used by 1:N protection, and is therefore optimal in this sense. Moreover, the protection circuit in [2] takes the form of a cycle, while it takes the form of a tree in this paper. Therefore, the scheme proposed in this paper is also more optimal in terms of agility, since the tree will always result in a recovery time that is shorter than that provided by the cycle.

The rest of the paper is organized as follows. In Section II we introduce the network model, and a few definitions and operational assumptions. In Section III we illustrate the basic concepts of our strategy to protect unidirectional connections against single link failures. This is followed by the description of the general strategy. Some notes on the implementation of this technique are presented in Section IV. An Integer Linear Program (ILP) formulation for optimally protecting a group of connections in a network using the proposed scheme is presented in Section V. Section VI presents some numerical results based on the ILP formulation, which are compared to 1+1 protection. Section VI also shows some simulation results from an OPNET implementation in order to evaluate outage times and buffer occupancies. Availability modeling and availability results for a case study consisting of three connections provisioned on NSFNET, which are protected by the proposed strategy, are also presented in Section VI. Finally, the paper is concluded with some remarks in Section VII.

II. Definitions and Assumptions

In this section we introduce a number of definitions and assumptions about the network, the connections to be protected, and which connections are protected together.

- The network is represented by the undirected graph $G(V, E)$, where V is the set of nodes, and E is the set of undirected 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. A node can be a router, or a switch, depending on the graph abstraction level and the protection layer. 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*. For example, at the physical layer, the span may be a fiber, and the link may be a wavelength channel, or even circuits with sub-wavelength granularities, e.g., DS3, if a technique like traffic grooming is used.
- There is a set C of bidirectional unicast connections that need to be provisioned in the network such that 100% 1+N protection against single link failures 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. Therefore, link protection is a special case of this technique.
- Connections are bidirectional and they require the same bandwidth in both directions. A connection c_j is between nodes S_j and D_j. Node S_j transmits data units $s_j^{(n)}$, where n is the sequence number, or round number in which the data unit is transmitted, while node D_j transmits $d_j^{(n)}$ in the same round. Such data units are transmitted on a working path dedicated for the connection. The data units received by S_j and D_j will be referred to as $\hat{d}_j^{(n)}$ and $\hat{s}_j^{(n)}$,

[2] If the packet received on the working path is corrupted, it can be assumed as 0, i.e., lost, and can be recovered from using the proposed technique.

respectively. Connection $c_j \in C$ is identified by the tuple $< S_j, D_j, s_j^{(n)}, d_j^{(n)} >$.

- All data units are fixed in size.
- The protection scheme, 1+N protection, will guarantee that if any link on the working path of connection c_j fails, then the end nodes of the connection, S_j and D_j, can recover a copy of the data unit $d_j^{(n)}$ and $s_j^{(n)}$, respectively, using the protection circuit.
- 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 $\sum_{i=1}^{K} N_i = N$. Partitioning the set of connections may also be applied even when the joint protection of all N connections is feasible, if this results in a more efficient provisioning and protection scheme.
- The scheme presented in this paper is designed to protect against a single link failure. That is, when a link fails, recovery of the data lost due to failures will take place, and the failed link will be repaired before another link fails. Because connections are link disjoint, then the protection is also against any connection failure, which may involve multiple link failures.
- When a link carrying an active circuit of connection c_j fails, the two end nodes of the connection will receive empty data units, which can be regarded as zero data units, i.e., $\hat{s}_j^{(n)} = \hat{d}_j^{(n)} = 0$.

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

III. The Generalized 1+N Protection Scheme

In this section we introduce a resource efficient approach for implementing 1+N Protection for guaranteed protection against single link failures, which is based on the use of network coding. 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.

A. Basic Principles

Under 1+1 protection (see Figure 1 for an example of 1+1 protection of three connections), two link disjoint paths are established for each connection[3]. Two copies of the same data are transmitted on the two paths, such that if one path fails, the receiver is guaranteed to receive a second copy. This scheme is fast, since it does not require failure detection, localization or rerouting. However, the more resource efficient 1:N protection, which is an extension of 1:1, protects N link disjoint working paths using one protection path (see Figure 2 for an example of protecting three unidirectional connections). Once a working path fails, e.g., the path from $S2$ to $D2$ in the figure, the failure will have to be detected, localized, switches must be reconfigured, and data from $S2$ must be rerouted to use the protection path. In the same way 1:N is a generalization of the 1:1 strategy, we would also like to extend 1+1 to 1+N, where data from multiple connections are transmitted simultaneously on a shared protection circuit, such that when there is a failure the data affected by the failure would be readily available through the protection circuit. Unfortunately, straightforward transmission of different data units on a shared protection circuit will result in collisions, and hence loss of data. To circumvent this problem, we use network coding to combine multiple data units on the protection circuit.

For example, consider the network in Figure 3, where we show three unidirectional connections for simplicity, and one protection path is used to protect the three working paths. Each of the three connections is from node S_j to node D_j, where $j = 1, 2, 3$. Node S_j sends data unit s_j to node D_j. At the same time, node S_j sends its s_j data unit to one (or more) node(s) in the network (node A in the figure), where all s_j data units are linearly combined by performing modulo-2 addition. The sum is delivered to another node, X, in the network. Node D_j will also send its received data unit to node B in the network, where these data units will also be linearly combined using the modulo-2 addition, and the sum is then delivered to the same node X (nodes A, B and X may be the same or different nodes in the network)[4]. As will be shown in Section III-B, such a node always exists. At node X, the linear combinations received from the S_j and D_j nodes are combined, also using modulo-2 addition, and this sum is then delivered to the D_j nodes. In the absence of failures, this sum will be 0. However, when a failure takes place, e.g., on the connection from node S_2 to node D_2 in the figure, s_2 will not be received by D_2, i.e., $\hat{s}_2 = 0$, and the sum obtained at node B will be $s_1 + s_3$. Therefore, the total sum at X will be the missing data unit, s_2, which will be delivered to D_2.

B. 1+N Protection of Bidirectional Connections Against A Single Link Failure

In this section we describe the design procedure for generalized 1+N protection against single link failures. The example in Figure 4 is a generalization of that in Figure 3, and is used to illustrate the procedure.

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

- A total of N_i link disjoint working paths are provisioned to carry the data units directly between source S_j and 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 unit $s_j^{(n)}$ is transmitted from S_j to D_j in round n, while data unit $d_j^{(n)}$ is transmitted from D_j to S_j in the same round.
- A *protection circuit*, P_i, is provisioned for all connections in C_i. The minimal cost protection circuit takes the form of a tree, as will be proven below. Therefore, the protection circuit has at least one bridge node, and let

[3] Each of the paths shown in the example figure may consist of multiple links. The paths are shown here as single links in order to simplify the presentation. The provisioning problem in Section V will provision the connections and their protection circuits with the fewest number of links, and it will be shown that 1+1 protection requires more resources.

[4] Note that this creates a cycle in the graph between nodes B and X, and it is the introduction of this cycle that enables a static network code to protect against all single failures possible.

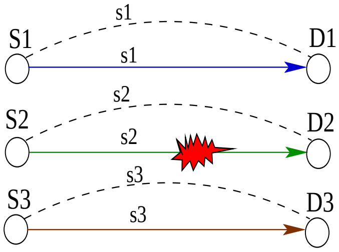

Fig. 1. An example of 1+1 protection.

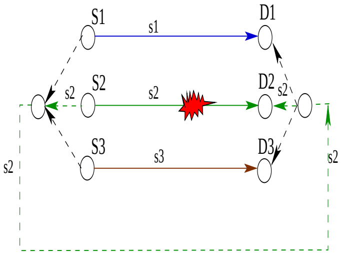

Fig. 2. An example of 1:N protection.

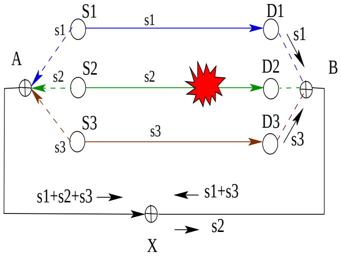

Fig. 3. An illustration of the concept of Generalized 1+N protection for unidirectional connections.

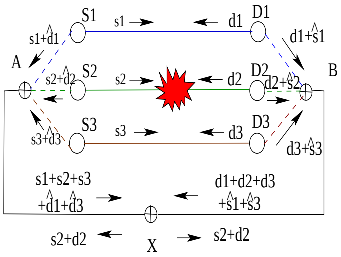

Fig. 4. An illustration of the concept of Generalized 1+N protection against a bidirectional connection failure

us refer to one such bridge node as X_i. Each node S_j transmits the sum $s_j^{(n)} + \hat{d}_j^{(n)}$ on P_i, while node D_j transmits the sum $d_j^{(n)} + \hat{s}_j^{(n)}$ also on P_i. The P_i circuit is used to deliver the sum of data units $\sum_{j, c_j \in C_i} s_j^{(n)} + \hat{d}_j^{(n)}$ from S_j nodes to X_i, and is also used to deliver the sum of data units $\sum_{j, c_j \in C_i} d_j^{(n)} + \hat{s}_j^{(n)}$ from D_j nodes, also to X_i. P_i is link disjoint from the working paths in C_i.

The shape of the minimal cost P_i circuit is a tree, which is proven in the following proposition:

Proposition 1. *Under the assumption of undirected edges in the network graph G, the minimal cost protection circuit, P_i, where the cost is in terms of the number of network edges, is a tree.*

Proof: The circuit P_i is a subgraph that connects all end nodes of all connections in C_i. We prove this proposition by proving the contrapositive, i.e., if P_i is not a tree, then it is not minimal. Let us assume that P_i is not a tree. Therefore, there is a cycle in P_i. The cycle can be removed by eliminating one or more edges of P_i, while still allowing transmissions from S_j end nodes to reach all D_k nodes in C_i, and vice versa. Therefore, this reduces the cost of P_i, and hence the non-tree graph is not minimal. ■

What the above proposition means is that we will have to find the minimal cost tree that connects the end nodes in C_i. Notice that in the above proof, eliminating an edge to remove the cycle can be followed by further reductions in the cost of the tree. This can be achieved by recursively eliminating edges with leaf nodes which are not in the set of end nodes of the connections in C_i. This will eventually lead to a Steiner Tree. However, the minimal cost such tree is a Steiner Minimal Tree (SMT) [10], which is in the class of NP-Complete problems.

Since P_i is a tree, then it is easy to see that any non-leaf node is actually a bridge node of the tree. We choose one such bridge node for collecting the linear combinations of transmissions from all S_j and D_k nodes in C_i, and use these combinations to recover from data lost due to failures. We refer to this node as X_i. The selection of X_i is important to minimize the outage time, which is the time that a receiver node will have to wait after the failure until it starts receiving recovered data. This issue will be addressed below.

The undirected tree, P_i, is then treated as two directed trees: one from the leaf nodes towards X_i, using the shortest distance metric, e.g., number of hops, and the second tree is rooted at X_i, and is directed from X_i towards to the leaf nodes, also using the shortest distance metric. The two trees are identical, except that directions of the edges are reversed. We now describe the role of the different nodes in providing 1+N protection:

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

- Transmit data unit $s_j^{(n)}$ on the working path W_j to D_j in round n.
- When $\hat{d}_j^{(n)}$ is received on W_j, form $s_j^{(n)} + \hat{d}_j^{(n)}$ and transmit this sum on the outgoing link of P_i.
- If $\hat{d}_j^{(n)} = 0$, then add $s_j^{(n)}$ to the data received on the incoming link of P_i corresponding to round n in order to recover $d_j^{(n)}$; otherwise, ignore the data received on P_i.

Role of Node D_j of connection $c_j \in C_i$:
Node D_j will take actions very similar to those taken by S_j, except that $s_j^{(n)}$ and $d_j^{(n)}$ are interchanged:

- Transmit data unit $d_j^{(n)}$ on the working path W_j to S_j in round n.
- When $\hat{s}_j^{(n)}$ is received on W_j, form $d_j^{(n)} + \hat{s}_j^{(n)}$ and transmit this sum on the outgoing link of P_i.
- If $\hat{s}_j^{(n)} = 0$, then add $d_j^{(n)}$ to the data received on the incoming link of P_i corresponding to round n in order to recover $s_j^{(n)}$; otherwise, ignore the data received on P_i.

Role of Intermediate Nodes on P_i:
All intermediate nodes on P_i, except for X_i, e.g., nodes A and B in Figure 4, will take the following actions:

- For data received on incoming links from the leaf nodes, and going towards X_i, add all data units (possibly linear combinations) belonging to round n using modulo-2 addition, and forward the sum towards X_i.
- For data received on an incoming link from X_i and going towards the leaf nodes, duplicate the data and broadcast on all outgoing links.

Note that nodes S_j and D_j in C_i may also act as intermediate nodes, e.g., if P_i is realized as a path. In this case, each such node can be represented by two virtual nodes, e.g., node S_j can be represented by $S_j^{'}$ and $S_j^{''}$, which are connected by a bidirectional edge:

- Node $S_j^{'}$ is connected to W_j, and acts like S_j above, and
- Node $S_j^{''}$ acts like the intermediate node described above.

Figure 5 shows an example of this situation, and the linear combinations formed in the direction of node X_i.

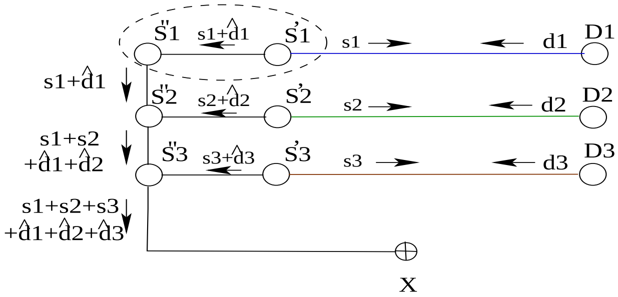

Fig. 5. An example of the case in which S_j nodes act as both end nodes, and intermediates of P_i (the P_i tree is partially shown only for illustration purposes); each source node S_j is treated as two virtual nodes: a source node, $S_j^{'}$, and an intermediate node on P_i, $S_j^{''}$.

Role of Node X_i on P_i:

- Add (modulo-2 addition) linear combinations belonging to round n and received on incoming links.
- The sum obtained in the first step is broadcast on all outgoing links from X_i towards the leaf nodes.

We illustrate this process using the example in Figure 4, when the connection between S_2 and D_2 fails. In this case, $\hat{s_2} = \hat{d}_2 = 0$, and summing the linear combinations arriving at X yields $s_2 + d_2$. This sum is broadcast back to end nodes of all connections. Nodes S_2 and D_2 can recover d_2 and s_2 by adding s_2 and d_2, respectively. Notice that the end nodes of other connections cannot recover either of these data units, which makes this method secure, as far as the end nodes are concerned.

C. The Selection of Node X_i:

As explained above, node X_i is a vertex on the SMT that receives linear combinations from S_j and D_j nodes in C_i, and then after adding them, transmits the sum back to the S_j and D_j nodes. The choice of the X_i will influence the outage time, ψ_i, which is the maximum time between the detection of the loss of signal on the working path and the recovery of the same signal. To see this, we make the following assumptions and definitions:

- Processing times on all paths are included in all delays.
- The working path delay for connection $c_j \in C_i$ is $\tau_j^{(i)}$.
- The delay between nodes S_j in connection $c_j \in C_i$ and X_i is $\sigma_j^{(i)}$.
- The delay between nodes D_j in connection $c_j \in C_i$ and X_i is $\delta_j^{(i)}$.
- The diameter of P_i, i.e., the maximum delay between any pair of vertices in P_i, is θ_i.
- The delay between any two nodes is symmetric in both directions.

Assuming that all data units in the same round are transmitted by all nodes at the same time, then ψ_i can be expressed as follows:

$$\psi_i = \max_{c_j, c_k \in C_i} [\tau_j^{(i)} + 2\max(\sigma_k^{(i)}, \delta_k^{(i)}) - \tau_k^{(i)}] \qquad (1)$$

The above equation is based on the fact that for S_j (D_j) to send the $s_j^{(n)} + \hat{d}_j^{(n)}$ ($d_j^{(n)} + \hat{s}_j^{(n)}$) on P_i, it must receive $\hat{d}_j^{(n)}$ ($\hat{s}_j^{(n)}$) first, which takes $\tau_j^{(i)}$. Then, the linear combinations must be delivered to X_i, and the sum must be sent back from X_i to S_j (D_j) which takes $2\sigma_j^{(i)}$ ($2\delta_j^{(i)}$). Notice that $\max_k(\sigma_k^{(i)}, \delta_k^{(i)})$ is the eccentricity of X_i in the P_i graph, and θ_i is the maximum eccentricity in P_i, which is given by $\max_{j,k}(\sigma_j^{(i)} + \delta_k^{(i)})$.

To minimize the outage time, we note that $\sigma_j^{(i)} + \delta_j^{(i)}$ is equal to the delay on P_i between S_j and D_j. Therefore, equalizing $\sigma_j^{(i)}$ and $\delta_j^{(i)}$ will minimize ψ_i. This can be achieved by choosing X_i as the center of the P_i tree. Note that since P_i is a tree, then it is either central or bi-central, i.e., has two centers. In the latter case, X_i can be chosen as one of the two centers. There are several linear time algorithms in graph theory which can be used to find the tree center, and any of them can be used in this case. Based on this, the outage time is upper bounded by

$$\psi_i \leq \theta_i + \max_{j,k}(\tau_j^{(i)} - \tau_k^{(i)}) \qquad (2)$$

D. Existence Conditions

Although the graph G is assumed to be 2-connected, this does not guarantee that a backup circuit can be found to protect a given group of connections. The following theorem establishes the existence conditions of a protection circuit, P_i, for a given group of connections. We establish the conditions in terms of the max-flow from a source to its destination, which is equivalent to establishing the number of link disjoint paths from a source to its destination.

Theorem 2. *Consider a set of connections, C, which are provisioned in a network with graph G. Each connection, $j \in C$, is provisioned between two terminal nodes, S_j and D_j, such that all the working paths of the connections are link disjoint. Also, the network graph is at least 2-connected which allows a max-flow of at least 2 from any source, S_j, to its sink, D_j.*

A protection circuit exists for the set of connections, C, if and only if, for every connection $j \in C$, there exists a path p_j between the end nodes of the connection, S_j and D_j, such that deleting all the edges on p_j will not reduce the max-flow from any other source S_k to its destination D_k to less than 2, where $j \neq k$. Moreover, p_j is the working path of source S_j, for all j.

Proof:

- We first prove the implication, i.e., if there is a protection circuit, then the removal of a path, p_j, between S_j and D_j, does not reduce the max-flow between end points of any other connection below 2.

  Assume that there is a protection circuit which is link disjoint from all working paths. Also, by assumption, all working paths of the connections in C are link disjoint. Next, consider the working path between S_j and D_j, p_j, and remove all edges on p_j. Since this path is link disjoint from all other working paths, and is also link disjoint from the protection circuit, then each connection $k \neq j$ has at least two link disjoint paths between S_k and D_k, which are link disjoint from the path p_j: one is the working path, and the second is a path on the protection circuit. Therefore, these two paths are unaffected by the removal of p_j, and the max-flow from S_k to D_k is at least equal to 2.

- Next, we prove the converse, i.e., if deleting the path p_j does not reduce the max-flow between S_k and D_k to less than 2, for $k \neq j$, then there is a protection circuit.
  We delete all the edges on p_j. Since under this condition, and by assumption, each connection k, for $k \neq j$, has a max-flow of at least 2, connection k has two link disjoint paths between S_k and D_k. One such path is the working path, p_k, which, by assumption, is link disjoint of all other working paths. A second path, p'_k, must also be link disjoint of all other working paths, p_l. The reason that the last property holds, is that if it did not, and we continue the deletion process of all primary paths, except for p_k, and then p_l is deleted, this will cause at least one edge on p'_k to be deleted, and the max-flow of connection k will be reduced to less than 2. Therefore, the second path must be part of a protection circuit, and the union of all the edges on the second paths (which are not necessarily link disjoint) is this protection circuit.

■

IV. Implementation

The proposed 1+N protection strategy can be implemented at a number of layers, and using a number of protocols. Here, we propose an implementation using the Multiprotocol Label Switching (MPLS) [12]. MPLS has been chosen since Label Switched Paths (LSPs) provisioned under MPLS are stable and do not change route. Moreover, the use of route-pinning during the LSP establishment can guarantee link disjointedness between working and protection circuits. For this purpose, the 1+N protection may be implemented as a shim functionality between the IP and MPLS layers.

Notice that under 1+N protection, only data units which are transmitted in the same round are combined. Therefore, we require the use of round numbers. However, we show that, provided that all sources start transmissions in round 0, only two round numbers, 0 and 1, are needed. These round numbers are virtually implemented by using two MPLS LSPs for every link on the protection tree. Each LSP will be provisioned with half the capacity of the working paths, e.g., $B/2$. Hence, this implementation does not require any added capacity for the protection circuit beyond that described above. The two LSPs, which we refer to as LSP0 and LSP1, will be earmarked for transmitting linear combinations of data units transmitted in even and odd rounds, respectively. The LSPs are established between branch nodes on the P_i tree, i.e., nodes which implement merging in the inbound direction towards the root of the P_i tree, node X, and branching in the outbound direction towards $S_j, D_j \in C_i$ nodes.

To implement 1+N protection using MPLS, the following is implemented:

1) Packets are transmitted from the sources alternately on LSP0 and LSP1, starting from round 0.
2) At a node which is the end node of an LSP, and the start node of another LSP (except for node X) leading to the root of the tree, X, data units are alternately combined from all even LSPs (LSP0) and all odd LSPs (LSP1). When a data unit is not available, the process must wait for a data unit to become available. The IP data units are linearly combined without regard to their contents.
3) At node X, data units arriving from even LSPs and odd LSPs are alternately combined, and the sum is broadcast back to all $S_j,\ D_j \in C_i$ nodes, using the corresponding even and odd LSPs, respectively.
4) At a node which is the end node of an LSP, and the start node of another LSP leading away from X, data units received on an incoming even (odd) LSP are transmitted on all outgoing even (odd) LSPs leading to the $S_j,\ D_j \in C_i$ nodes.

As stated above, with the use of appropriately dimensioned buffers at the end nodes of LSPs, round numbers can be delineated by the use of two LSPs, LSP0 and LSP1 to carry linear combinations of data units transmitted in even and odd rounds, respectively. With the alternate combinations of data units from even and odd LSPs, it is guaranteed that round numbers will be observed. Notice that this means that the combining operation may be blocked by the absence of data units on an incoming LSP, and data units received on other incoming LSPs have to be buffered.

It should be noted that the proposed protection strategy is not limited to MPLS, and can be implemented in other technologies. In concept, this includes the optical domain. This requires the provisioning of two functionalities in the optical domain, namely, optical XORs, and optical delay lines. Delay lines are already available. As for optical XOR functions, progress has already been made [13], and they may be available soon in backbone networks.

Using the same assumptions and arguments used above for the derivation of equation (2), we can derive an upper bound on the buffer size (in terms of packets) per LSP at each of the nodes performing the XOR functions (code and forward nodes). Assuming that $\eta^{(\mathbf{i})}$ is the set of nodes performing the XOR function on the P_i tree between all nodes S_j which are the end nodes of connections $c_j \in C_i$ and node X, and that $\eta^{(i)}_{j,l}$ is the delay between node S_j and node $l \in \eta^{(\mathbf{i})}$, then the upper bound on the buffer at nodes in $\eta^{(\mathbf{i})}$ is given by

$$\max_{c_j,c_k \in C_i}[(\tau_j^{(i)} + \eta_{jl}^{(i)}) - (\tau_k^{(i)} + \eta_{kl}^{(i)})] \tag{3}$$

A similar expression can be derived for the nodes performing the XOR function on the other half of the P_i tree, i.e., on data units sent by nodes D_j on P_i. This bound is derived under the assumption that nodes transmit continuously, end nodes start a round simultaneously, and under the operational requirement that all combined data units must belong to the same round. The upper bound on the buffer at node X requires a slightly different argument, since node X must combine data received from both halves of the branches. This upper bound is given by

$$\max_{c_j,c_k \in C_i}[\tau_j^{(i)} + \max(\sigma_j^{(i)} - \sigma_k^{(i)}, \delta_j^{(i)} - \delta_k^{(i)}) - \tau_k^{(i)}] \tag{4}$$

The buffer size per input port, in terms of packets, can be dimensioned by multiplying the above bounds in (3) and (4) by B/L, where B is the transmission rate in bits/sec, and L is the packet size in bits, The above bounds are made under the assumption that propagation delays are constant. In practice, the propagation delay varies due to a number of reasons, including environmental factors, fiber strand length difference, cabling stress effects and group delay difference, and in multimode fibers, the factors also include numerical aperture and differential mode delay. In [14], the worst case skew in propagation delay was calculated to be 45.4 ps/m in OM3 multimode fiber. With the worst case propagation delay skew, and over a long strand of 1000 km fiber link, this is equivalent to a skew of 45.4 μsec. This skew can be accommodated by Multi-Service Provisioning Platform (MSPP) switches, which implement Next Generation SONET, and can accommodate up to 128 ms of differential delay. If implemented at higher layers, and if the fiber links are signaled at 10 Gb/s while carrying 1,500 byte packets, the skew is equivalent to less than 40 packets. This variation in delay can be accommodated using an elastic buffer. However, it was also stated in [14] that real system measurements showed that the actual skew is far lower than the worst case.

A final implementation issue that needs to be addressed here is the cost, the time and the complexity of implementing the XOR functions. We should note that the hardware needed to perform the bitwise XOR operation is already available in routers. Hence, no significant cost will be added for performing the coding or decoding functions. Also the delay for performing this function should not be significant, since the bitwise XOR operation can be performed sequentially as packets arrive at the routers. However, all of the above requires some added complexity to the operation of the routers, and this complexity is the tradeoff to bandwidth and cost savings achieved by 1+N protection.

V. ILP Formulation

The problem of finding link disjoint paths between pairs of nodes in a graph is known to be an NP-complete problem [15]. Hence, even finding the working paths in this problem is hard. What makes the problem of provisioning both the working and protection circuits under the Generalized 1+N Protection even harder is that the protection circuit is an SMT, which is also an NP-complete problem. We therefore introduce an Integer Linear Program (ILP) for solving the 1+N protection strategy introduced in this paper. It is to be noted that the solution is optimal under the given constraints, i.e., that there is a protection circuit, and that this circuit is link disjoint from the working paths it protects. In the ILP below, P_i is implemented using a group of multicast trees from each $S_j \in C_i$ to all $D_k \in C_i$. The multicast trees share links, and a link that is shared between several trees is only counted once in order to realize the Steiner Tree.

We assume that the number of channels per span is not upper bounded, i.e., the network is uncapacitated.

The following table defines the input parameters:

N number of connections

$s(k)$, $d(k)$ end nodes of connection k

δ^{kl} a binary indicator which is equal to 1 if connections k and l have the same destination

The variables used in the formulation are given below:

n^{kl} binary variable which is 1 if and only if connections k and l are protected together

z^k_{ij} binary variable which is 1 if and only if connection k uses link (i, j) on the working path

p^k_{ij} binary variable which is 1 if and only if connection k uses link (i, j) on protection circuit

P^{kl}_j binary variable, which is 1 if and only if the protection circuit for connections k and l share a node, j (required if $n^{kl} = 1$).

$\mathcal{P}^{kl}_{ij}$ binary variable which is 1 if and only if connections k and l are protected together, and share link (i, j) on the protection circuit.

$\pi^k_{i,j}$ binary variable which is equal to 1 if connection k is the lowest numbered connection, among a number of jointly protected connections, to use link (i, j) on its protection circuit (used in computing the cost of the protection circuit).

Minimize:

$$\sum_{i,j,k}(z^k_{i,j} + \pi^k_{i,j})$$

The summation above is the cost of the links used by the connections' working paths and the protection circuits.

Subject to:

Constraints on working paths:

$$z^k_{i,s(k)} = 0 \quad \forall k,\ i \neq s(k) \tag{5}$$

$$z^k_{d(k),j} = 0 \quad \forall k,\ j \neq d(k) \tag{6}$$

$$\sum_{i \neq s(k)} z^k_{s(k),i} = 1 \quad \forall k \tag{7}$$

$$\sum_{i \neq d(k)} z^k_{i,d(k)} = 1 \quad \forall k \tag{8}$$

$$\sum_i z^k_{ij} = \sum_i z^k_{ji} \quad \forall k,\ j \neq s(k),\ d(k) \tag{9}$$

$$z^k_{ij} + z^k_{ji} + z^l_{ij} + z^l_{ji} + n^{kl} \leq 2 \quad \forall k, l, i, j \tag{10}$$

Equations (5), (7), (6) and (8) ensure that the traffic on the working path is generated and consumed by the source and destination nodes, respectively. Equation (9) guarantees flow continuity on the working path. Equation (10) ensures that the working paths of two connections which are protected together are link disjoint. Since a working path cannot use two links in opposite directions on the same span (or edge in the graph), then two connections which are protected together cannot use the same span either in the same, or opposite directions. Such a condition is included in equation (10).

Constraints on protection circuits:

$$p^k_{i,s(k)} = 0 \quad \forall k,\ i \neq s(k) \tag{11}$$

$$p^k_{d(k),j} = 0 \quad \forall k,\ j \neq d(k) \tag{12}$$

$$\sum_{i \neq s(k)} p^k_{s(k),i} = 1 \quad \forall k \tag{13}$$

$$\sum_{i \neq d(k)} p^k_{i,d(k)} = 1 \quad \forall k \tag{14}$$

$$\sum_i p^k_{ij} = \sum_i p^k_{ji} \quad \forall k,\ j \neq s(k),\ d(k) \tag{15}$$

$$z^k_{ij} + \frac{p^k_{ij} + p^k_{ji}}{2} \leq 1 \quad \forall k, i, j \tag{16}$$

$$z^k_{ij} + \frac{p^l_{ij} + p^l_{ji}}{2} + n^{kl} \leq 2 \quad \forall k, l, i, j \tag{17}$$

$$\sum_i (p^k_{ij} + p^l_{ij}) \geq 2P^{kl}_j \quad \forall k, l, j \tag{18}$$

$$\sum_i (p^k_{ji} + p^l_{ji}) \geq 2P^{kl}_j \quad \forall k, l, j \tag{19}$$

$$\sum_j P^{kl}_j \geq n^{kl} \quad \forall k, l \tag{20}$$

Equations (11), (12), (13), (14) and (15) serve the same purpose as equations (5)-(9), but for the protection circuit. Equation (16) makes sure that the working path and its protection circuit are link disjoint, while equation (17) makes sure that if two connections k and l are jointly protected, then the protection circuit of l must also be disjoint from the working path of connection k. Notice that both of equations (16) and (17) allow a protection circuit to use two links in opposite directions on the same span, and this is why the sum of the corresponding link usage variables is divided by 2 in both equations. Equations (18), (19) and (20) make sure that if two connections, k and l, are protected together ($n^{kl} = 1$), then their protection paths must have at least one joint node. This joint node, identified by j, is computed using equation (20), which makes sure that if k and l are protected together, then at least one of the P^{kl}_j variables is equal to 1.

Constraints on joint protection:

$$n^{kl} + n^{lm} - 1 \leq n^{km} \quad \forall k, l, m \tag{21}$$

Equation (21) makes sure that if connections k and l are protected together, and connections l and m are also protected together, then connections k and m are protected together.

Constraints for cost evaluation:

$$\mathcal{P}^{kl}_{ij} \leq \frac{p^k_{ij} + p^l_{ij} + n^{kl}}{3} \quad \forall i, j, k, l \tag{22}$$

$$\pi^l_{ij} \geq p^l_{ij} - \sum_{k=1}^{l-1} \mathcal{P}^{kl}_{ij} \quad \forall l, i, j \tag{23}$$

Equations (22) and (23) are used to evaluate the cost of the protection circuits, which are used in the objective function. Equation (22) will make sure that $\mathcal{P}^{kl}_{ij}$ cannot be 1 unless connections k and l are jointly protected using link ij. Note that $\mathcal{P}^{kl}_{ij}$ should be as large as possible since this will result in decreasing the protection circuit cost, as shown in equation (23). Equation (23) uses the lowest numbered connection among a group of jointly protected connections to contribute to the cost of the links shared by the protection trees.

VI. Performance Evaluation

In this section we evaluate the performance of our approach.

A. Implementation Cost and Comparison

In this section, we provide some results about the cost of implementing the proposed approach based on the ILP formulation in Section V, and compare them to the cost of implementing 1+1 protection. For the 1+1 protection, the cost is based on the optimal Bhandari's algorithm [11].

We first considered a network with 8 nodes and 12 edges, and hence the average nodal degree is 3. The network graph was randomly generated such that the graph is bi-connected. We also generated random connections, and three cases of the cardinality of the set of connections were considered, namely 6, 8 and 10. The results are shown in Table I. The saving in the number of links used by the protection circuit can reach 28% due to the use of 1+N protection, and a total cost saving close to 18%. We then added 4 more edges to the network graph in order to make the average nodal degree equal to 4. The results are also shown in Table I.

The increase in the graph density resulted in a reduction in the amount of resources required for both working and protection circuits. Moreover, a greater reduction in the amount of protection circuits was achieved when using 1+N protection, reaching 35%, in addition to a total cost reduction of 20%.

TABLE I
Cost comparison between 1+1 and 1+N protection for networks with $|V| = 8,\ |E| = 12$, and $|V| = 8,\ |E| = 16$.

$\lvert V\rvert, \lvert E\rvert$	N	1+1			1+N		
		Total	Working	Spare	Total	Working	Spare
8, 12	6	26	11	15	23	12	11
	8	40	16	24	38	18	20
	10	40	15	25	33	17	16
8, 16	6	20	8	14	17	9	8
	8	30	13	17	24	13	11
	10	36	16	20	30	16	14

By inspecting the results from the optimal solution obtained using the ILP, it was observed that not all connections are protected together. Connections where the end nodes are localized tend to be protected together. The reason for this is to reduce the number of links which are used by separate connections to carry their data to the backup circuit, e.g., the links between S_1, S_2, S_3 and A, and between D_1, D_2, D_3 and B in Figure 3, since these links are not shared. This may cause some connections to be protected separately using 1+1 protection (since it is a special case of 1+N) if this is less expensive than protecting them jointly with other connections.

B. OPNET implementation

The NSF network topology is used in our simulation. We applied the optimal 1+N protection scheme to three bi-directional unicast sessions, (S_1=1,D_1=13), (S_2=3,D_2=12) and (S_3=4,D_3=10), as shown in Figure 6, where the bold links represent the working paths between each pair of connection end nodes, and the dashed links represent the backup tree. Node X is chosen to be node 5. The end-to-end delays between the end nodes of the three connections are 19.4 ms, 10.5 ms and 7.2 ms, respectively. It is to be noted that the S_j nodes (or the D_j nodes) need not be localized together in order to be jointly protected. Even if the S_j nodes were farther apart, it is sufficient that the protection tree P_i be minimized in order to achieve a cost saving.

The cost of the working circuits is 9 links, and the cost of the protection circuits is also 7 links. If these connections are to be protected using 1+1 connections, then connection (1,13) can be protected by the protection path (1,3,9,13), connection (3,12) can be protected by protection path (3,4,5,11,12), and connection (4,10) can be protected by protection path (4,5,8,10). The total protection cost under 1+1 is therefore 10 links, which is more than 40% higher than that under 1+N.

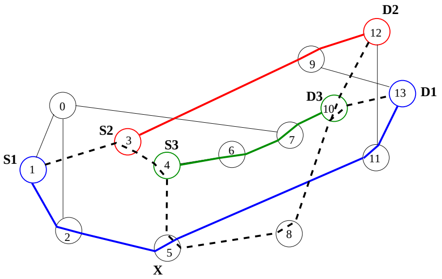

Fig. 6. NSF Network

MPLS was used in the communication between nodes. Each node in Figure 6 corresponds to an MPLS Label Switched Router (LSR). To simplify the implementation and to avoid modifying network protocols we chose to provide the extra functionalities through connecting dummy workstations to LSRs. That is, if a node is a source (e.g., node 3) a workstation that generates traffic is connected to the LSR corresponding to that node. Moreover, if a node performs two jobs (e.g., node 10 is a source and a ”code and forward” node) two workstations (one for each job) are connected to the LSR corresponding to that node. The delay on the links connecting the workstations to LSRs is set to 0 so that it does not affect the simulation results. The delay on remaining links is calculated by OPNET and is distance based. The FECs used by MPLS are based on destination addresses. The LSRs were manually configured to perform static traffic mapping to LSPs and static routing.

TABLE II
SIMULATION PARAMETERS

	Values of parameter		
Parameter	Scenario #1	Scenario #2	Scenario #3
Inter-arrival time	0.5 sec	20 msec	2ms
Packet size	500 bytes	200 bytes	200 bytes

There are six source nodes in the network which generate traffic under the three scenarios with parameters given in Table II. All traffic generated is Constant Bit Rate (CBR). The first scenario corresponds to light traffic, while the second scenario corresponds to VoIP traffic using the 64 Kb/s G.722 speech codec, with one packet every 20ms. The third scenario is also a VoIP example, except that each connection corresponds to a trunk line carrying 10 calls, which are transmitted independently, i.e., without aggregation using RTP. All links use the DS3 carrier, which is signaled at a rate of 45 Mb/s. We are interested in the outage time at receiver nodes, and the maximum buffer size at coding points. The results for the average outage time on all nodes in all three scenarios are shown in Table III. From the table, we can observe that the system provides a very reasonable recovery time. Note that the outage time is equal to the time needed to decode the data unit using the combination from node X regardless if a failure really occurs on the working path or not. Therefore, as it is clear from the figures, the outage time depends on the distance from node X as one would intuitively expect, and the pair with the highest end-to-end delay has the lowest outage times, and vice versa. It can be observed that the outage times for each end node, under the different scenarios, are very close, and they are in the range of 22 to 38 milliseconds, which is much less than the industry standard of 50ms for automatic protection switching.

TABLE III
AVERAGE OUTAGE TIMES FOR CONNECTION END NODES (IN MILLISECONDS)

	S1	D1	S2	D2	S3	D3
Scenario 1	23.6	30.4	29.1	37.0	30.1	37.5
Scenario 2	22.7	29.9	28.4	36.3	29.7	37.0
Scenario 3	22.5	29.9	28.3	36.3	29.6	37.1

The buffer occupancy was measured at the nodes performing the code and forward operation, namely, nodes 3, 4, 5 and 10. The minimum and maximum buffer occupancies are shown in Table IV. The maximum buffer occupancy was found to be 2 packets at node 10, and 1 packet at the remaining coding nodes. The buffers were stable because of the CBR traffic. We leave the case of VBR traffic for future work.

The VoIP scenarios were also run on the same NSF Network, but using OC48 links, which are signaled at the rate of 2.5 Gbps. No significant differences were observed in the results. This is because the propagation delay is dominant over packet processing and transmission times.

TABLE IV
BUFFER OCCUPANCIES FOR CODING NODES IN THE STEADY STATE (IN PACKETS)

	Scenarios 1 & 2 occupancies		Scenario 3 occupancies	
Node	Minimum	Maximum	Minimum	Maximum
3	0	1	5	6
4	0	1	8	9
10	0	2	13	15
5 (X)	0	1	3	4

C. *Network Availability*

As part of the performance study of the proposed mechanism, we also evaluate the steady-state availability of connections provisioned under the scheme. We only consider the 2-terminal availability, which refers to the availability between the two end nodes of a connection. Availability is therefore defined as the steady state probability that the network is operational in a manner that allows the two end nodes of the connection to successfully exchange data units, either over the primary path or using the protection circuit. This takes into account link failures as well as failure repairs.

In order to evaluate the 2-terminal availability of a given connection, we model the changes in states of the links which impact communication between the two terminals under study, using a time homogeneous, continuous time Markov chain. We make the following assumptions:

- Only link failures and repairs are considered; node failures are not considered.
- Link failures are independent and identically distributed.
- Link inter-failure times are exponentially distributed with rate λ, i.e., the Mean Time To Failure (MTTF) is $1/\lambda$.
- Link repair times are independent and identically distributed.
- A link is repaired in an exponentially distributed time with rate μ, i.e., the Mean Time To Repair (MTTR) is $1/\mu$.

It is to be noted that, in practice, MTTF is much larger than MTTR, and therefore, $\mu >> \lambda$. For example, [16] reported that MTTF for 1,000 sheeth miles, and MTTR values are 2,000 hours and 12 hours, respectively. More recently, [17] (see also [3]) documented the average number of fiber cuts per 1,000 miles per year as 13 times and 3 times in metro and long haul networks, respectively. It is therefore expected that the use of protection circuits will improve the 2-terminal availability. Although there is no standard value for terminal availability, but a value of three nines (0.999) for availability is usually expected by the industry.

We also consider the 2-terminal availability of the connections provisioned on the NSFNET, together with their protection circuit, which are shown in the example of Figure 6. Exact modeling of these connections will require the use of a 4-dimensional Markov chain, and we therefore resort to an approximate model.

We consider a target connection with I links on the primary path, while the other two connections are provisioned using K_1 and K_2 links, respectively. The number of links on the protection circuit is equal to J links. We model this system using a two-dimensional Markov chain, where the state of the system is given by the ordered pair (i, j). i is the number of failed links on the primary circuit of the target connection, while j is the number of failed links on the protection circuit as well as the other two connections. Therefore, $0 \leq i \leq I$ and $0 \leq j \leq J + K_1 + K_2$. This means that the two terminals can exchange data units with probability 1 if the Markov chain is in states $(i, 0)$ or $(0, j)$, for $i, j \geq 0$[5].

In case both i and j are greater then 0, the availability of the target connection is evaluated using an approximation. We assume that out of the j links which have failed on the protection circuit, as well as the other two connections, any link can be in the failed state in an equally likely manner. Therefore, the probability that there are k_1 and k_2 failed links on the two connections, and $j - k_1 - k_2$ failed links on the protection circuit is given by

$$p(k_1, k_2, j-k_1-k_2) = \frac{\begin{pmatrix} K_1 \\ k_1 \end{pmatrix}\begin{pmatrix} K_2 \\ k_2 \end{pmatrix}\begin{pmatrix} J \\ j-k_1-k_2 \end{pmatrix}}{\begin{pmatrix} K_1+K_2+J \\ j \end{pmatrix}} \tag{24}$$

If we define $Q = [q_{(i,j),(i',j')}]$ as the transition rate matrix of the Markov chain from state (i, j) to state (i', j'), then the transision rates are given by:

$$q_{(i,j),(i',j')} = \begin{cases} (I-i)\lambda & i' = i+1, j = j', i < I \\ (J-j)\lambda & j' = j+1, i = i', j < J \\ i\mu & i' = i-1, j = j', i > 0 \\ j\mu & j' = j-1, i = i', j > 0 \\ -\sum_{m,n,(m,n)\neq(i,j)} q_{(i,j),(m,n)} & i' = i, j' = j \\ 0 & \text{otherwise} \end{cases}$$

In order to evaluate the steady state 2-terminal availability for a given connection, we solve for the steady state probabilities, $\pi_{(i,j)}$, using the relations:

$$\vec{\pi}Q = 0 \quad \text{and} \quad \sum_{i,j} \pi_{(i,j)} = 1$$

where $\vec{\pi}$ is the vector of the steady state probabilities of all states. The steady state 2-terminal availability, A, can therefore be expressed as

$$\begin{aligned} &\sum_{j=0}^{J+K_1+K_2} \pi_{(0,j)} + \sum_{i=1}^{I} \pi_{(i,0)} + 0.5 \sum_{i=1}^{I} \sum_{k_1=1}^{K_1} \pi_{(i,k_1)} \cdot p(k_1, 0, 0) \\ &+0.5 \sum_{i=1}^{I} \sum_{k_2=1}^{K_2} \pi_{(i,k_2)} \cdot p(0, k_2, 0) \\ &+0.333 \sum_{i=1}^{I} \sum_{k_1=1}^{K_1} \sum_{k_2=1}^{K_2} \pi_{(i,k_1+k_2)} \cdot p(k_1, k_2, 0) \end{aligned} \tag{25}$$

The first term in equation (25) corresponds to the case in which the target connection is operational. The second term is when the target connection has failed, but neither of the other two connections, nor the protection circuit have failed, and the protection circuit can therefore be used to recover data from the failed target connection. The third and fourth terms are the cases in which the target connection fails, as well as only one of the other two connections. The protection circuit does not fail in this case, and it is used to protect the target connection with probability 0.5. The last term is when all three connections fail, while the protection does not fail, and is used to protect the target connection with probability $\frac{1}{3}$.

Based on the above model, in Table V we show the 2-terminal availability for the three connections provisioned in Figure 6, and protected using the proposed 1+N protection scheme. The connections are provisioned over 4, 2 and 3 links, respectively. In the table, there are two cases of MTTR, 12 and 24 hours. For each of these two cases, MTTF takes four different values, namely, 1, 3, 6 and 12 months. First, as expected, it is observed that as MTTF increases, the 2-terminal availability increases. However, also as expected, as MTTR increases, the 2-terminal availability decreases, since it takes longer to repair failed links, which increases the likelihood of concurrent multiple failures. It is also observed that as the number of links between the two end nodes increases, the 2-terminal availability decreases. This observation is consistent with other studies, e.g., [18]. In general, and for all cases of practical interest in terms of MTTR and MTTF, the 2-terminal availability is very high, and it either exceeds, or is very close to the three 9's (0.999) steady state availability expectation, and it rarely goes below 0.99, i.e., it is within 1% of the target availability. Only when MTTR is 24 hours, and the MTTF is less than 3 months that the availability becomes less than 0.99.

[5] Note that the connection may be operational for cases in which both $i > 0$ and $j > 0$. However, considering such cases will significantly complicate the Markov chain, and we therefore ignore these cases.

TABLE V
AVAILABILITY LEVELS FOR THE THREE CONNECTIONS PROVISIONED IN FIGURE 6

	MTTR=12 hours				MTTR=24 hours			
MTTF in months	1	3	6	12	1	3	6	12
Connection 1: (3,9,12)	0.994850	0.999379	0.999841	0.999960	0.981711	0.997615	0.999379	0.999841
Connection 2: (4,6,7,10)	0.992665	0.999113	0.999773	0.999943	0.974043	0.996600	0.999113	0.999773
Connection 3: (1,2,5,11,13)	0.990735	0.998878	0.999713	0.999928	0.967311	0.995702	0.998878	0.999713

Such cases are not typical, since in long haul networks when MTTR is 24 hours, MTTF is about 4 months per 1,000 miles.

VII. CONCLUSIONS

This paper has introduced a strategy for 1+N protection against single link failures, which has the same cost as 1:N protection in terms of the used network resources, when the protection circuit is link disjoint of the protected connections. Hence, the proposed approach is resource optimal under this condition. The strategy uses network coding to protect a set of bidirectional connections, which are provisioned using link disjoint paths. Network coding is used to transmit linear combinations of data units on a protection circuit. The linear combinations are based on simple modulo-2 additions, or the XOR operation. The protection circuit is a tree, and the center of this tree assists the recovery process by adding incoming linear combinations, and broadcasting the sum back to all end nodes. The center of the tree should be carefully chosen in order to minimize the outage time. An implementation in terms of MPLS was proposed for this strategy. An optimal ILP formulation for provisioning the connections as well as the protection circuits was introduced. Numerical examples based on this optimal formulation were introduced and showed that the resources consumed by this strategy are significantly less than those needed by 1+1 strategies. A simulation study using the OPNET simulator showed that the proposed scheme can achieve an outage time which is less than 50 ms, and the buffer occupancy at coding nodes is small. Availability modeling and analysis for the same simulated case was also introduced in the paper, and it was shown that the proposed strategy can achieve availability levels which meet the three 9's availability level. Future work includes the development of heuristic algorithms for finding and provisioning the protection and working circuits, as well as developing implementation strategies in different technologies.

REFERENCES

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

[2] A. E. Kamal, ”1+N Network Protection for Mesh Networks: Network Coding-Based Protection using p-Cycles”, IEEE/ACM Transactions on Networking, Vol. 18, No. 1, Feb. 2010, pp. 67–80.

[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] A. E. Kamal, ”1+N Protection Against Multiple Faults in Mesh Networks”, in the proceedings of the IEEE International Conference on Communications (ICC), 2007.

[6] A. E. Kamal and A. Ramamoorthy, ”Overlay Protection Against Link Failures Using Network Coding”, in the proceedings of the Conference on Information Sciences and Systems (CISS), 2008.

[7] O. M. Al-Kofahi and A. E. Kamal, ”Network Coding-Based Protection of Many-to-One Wireless Flows”, IEEE Journal of Selected Areas on Communications, Vol. 27, No. 5, 2009, pp. 797–813.

[8] O. M. Al-Kofahi and A. E. Kamal, ”Scalable redundancy for sensors-to-sink communication”, in the proceedings of the IEEE Globecom, 2008.

[9] R. Koetter and M. Medard, *An Algebraic Approach to Network Coding.* IEEE/ACM Transactions on Networking, Vol. 11, No. 5, Oct. 2003, pp. 782–795.

[10] F. K. Hwang and D. S. Richards, “Steiner tree problems,” *Networks*, vol. 22, pp. 55–89, 1992.

[11] R. Bhandari, *Survivable Networks: Algorithms for Diverse Routing*. Springer, 1999.

[12] D. Awduche, “Mpls and traffic engineering in ip networks,” *IEEE Communications*, vol. 37, pp. 42–48, Dec. 1999.

[13] M. Zhang, L. Wang and P. Ye, “All-Optical XOR Logic Gates: Technologies and Experimental Demonstrations”, IEEE Communications, Vol. 43, No. 5, May 2005.

[14] P. Kolesar and P. Anslow, ”Propagation Delay Skew in Multimode Channels”, IEEE 802.3ba contribution, May 2008 (available at http://www.ieee802.org/3/ba/public/may08/kolesar_01_0508.pdf).

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

[16] M. To and P. Neusy, ”Unavailability analysis of long-haul networks,” *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 1, Jan. 1994, pp. 100-109.

[17] A. J. Vernon and J. D. Portier, ”Protection of Optical Channels in All-Optical Networks,” *National Fiber Optic Engineers Conference (NFOEC)*, 2002, pp. 1695–1706.

[18] P. Cholda and A. Jajszczyk, “Reliability Assessment of Optical p-Cycles,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 6, Oct. 2007, pp. 1579–1592.

PLACE
PHOTO
HERE

Ahmed E. Kamal Ahmed E. Kamal (S’82-M’87-SM’91) received a B.Sc. (distinction with honors) and an M.Sc. both from Cairo University, Egypt, and an M.A.Sc. and a Ph.D. both from the University of Toronto, Canada, all in Electrical Engineering in 1978, 1980, 1982 and 1986, respectively. He is currently a professor of Electrical and Computer Engineering at Iowa State University. His research interests include high-performance networks, optical networks, wireless sensor networks, wireless networks and performance evaluation. He is a senior member of the ACM, and a registered professional engineer. He was the co-recipient of the 1993 IEE Hartree Premium for papers published in Computers and Control in IEE Proceedings for his paper entitled Study of the Behaviour of Hubnet, and the best paper award of the IEEE Globecom 2008 Symposium on Ad Hoc and Sensors Networks. He was the chair or co-chair of a number of IEEE and ACM sponsored conferenced, and is on the editorial boards of the Computer Networks journal, and the Journal of Communications.

PLACE
PHOTO
HERE

Osameh M. Al-Kofahi Osameh M. Al-Kofahi received his B.Sc. from Jordan University of Science and Technology (JUST) in 2002, and the Ph.D. degree from Iowa State University in 2009, both in Electrical and Computer Engineering. He is currently an assistant professor of Computer Engineering at Yarmouk University in Jordan. His research has focused on developing network-coding based network survivability techniques. He was the co-recipient of the best paper award of the IEEE Globecom 2008 Symposium on Ad Hoc and Sensors Networks. He is a member of the IEEE and the Association of Computing Machinery.