Protection Cycles in Mesh WDM Networks

Georgios Ellinas, *Member, IEEE*, Aklilu Gebreyesus Hailemariam, and Thomas E. Stern, *Life Fellow, IEEE*

***Abstract*—A fault recovery system that is fast and reliable is essential to today's networks, as it can be used to minimize the impact of the fault on the operation of the network and the services it provides. This paper proposes a methodology for performing automatic protection switching (APS) in optical networks with arbitrary mesh topologies in order to protect the network from fiber link failures. All fiber links interconnecting the optical switches are assumed to be bidirectional. In the scenario considered, the layout of the protection fibers and the setup of the protection switches is implemented in nonreal time, during the setup of the network. When a fiber link fails, the connections that use that link are automatically restored and their signals are routed to their original destination using the protection fibers and protection switches. The protection process proposed is fast, distributed, and autonomous. It restores the network in real time, without relying on a central manager or a centralized database. It is also independent of the topology and the connection state of the network at the time of the failure.**

***Index Terms*—Optical network, optical switches, protection, restoration strategies, survivability, wavelength division multiplexing (WDM) networks.**

I. INTRODUCTION

THE MOST prevalent form of communication failures is the accidental disruption of buried telecommunication cables. Fiber cuts may result, among other reasons, from construction work ("backhoe fade"), rodent damage, fires, or human error [1]. Clearly, the need for fast and reliable protection of services is essential in high capacity optical systems. This paper proposes a general methodology for performing link failure[1] protection in optical networks with arbitrary mesh topologies and bidirectional links using automatic protection switching (APS). The reader should note that switch failures are not discussed in this paper as they constitute a different problem with its own set of solutions. A summary of the approach taken when a switch fails can be found in [2] and [3]. For a more extensive analysis, the reader should turn to [4].

In networks using APS as their protection mechanism, failures are circumvented by re-routing signals from *working channels* to *protection channels*, using *protection switches* at the ends of each network link, which are activated immediately when a fault is detected. Some specific physical properties of the signal are monitored at all links and threshold values determining when to switch to the protection fiber are chosen [5]. Generally, the time it takes to detect the failure and switch to the protection fiber is on the order of milliseconds.[2] Within this time frame, the network can recover from a cable failure without interrupting the services transported over the network, e.g., telephone calls, data transfers, etc. [6].

The networks considered in this paper are composed of optical links and optical switches. Under normal operation, the network supports a number of active source–destination connections, whose paths are determined by the settings of the optical switches. The discussion applies generally to networks that may carry multiplexed connections on each fiber (e.g., wavelength division multiplexed networks). In these networks, a typical link consists of a pair of unidirectional working fibers and a pair of unidirectional protection fibers that are terminated by four protection switches. When a fiber link is cut, connections using that link are automatically restored by rerouting their optical signals around the fault using the protection fibers and protection switches.

This paper is divided into two parts. In the first part, a novel technique is presented, showing how to solve the APS problem in mesh networks. The second part demonstrates how this approach is implemented in an optical network to provide full protection capabilities against a fiber link failure.

Section II of this paper describes the general methodology for determining the interconnection configurations for the protection switches. A complete solution of this problem is presented for networks with planar or Eulerian (planar/nonplanar) topologies, together with algorithmic methods for extending it to networks with arbitrary nonplanar topologies. Section III demonstrates the link failure protection process. The APS system components and protocol are presented, followed by an explanation of how the APS process works in the case of a fiber link failure. Examples of link failure protection in networks with planar, nonplanar, and Eulerian topologies are also demonstrated. Bounds on the number of simultaneous possible link failure recoveries are calculated in Section IV. Conclusions follow in Section V.

II. GENERAL METHODOLOGY

The networks are modeled by directed graphs (digraphs), whose vertices represent the network switching nodes and whose directed edges represent the transmission fibers. All networks considered have a pair of unidirectional working fibers (constituting a bidirectional working link) and a pair of unidirectional protection fibers (constituting a bidirectional protection link) in each fiber link. For purposes of APS, only the protection fibers are represented in the digraph.

Manuscript submitted October 18, 1999; revised May 18, 2000. This work was performed when the authors were at the Center for Telecommunications Research, Columbia University, New York, NY 10027 USA.

G. Ellinas and A. Hailemariam are with Tellium Incorporated, Oceanport, NJ 07757 USA (e-mail: gellinas@tellium.com, aklilu@tellium.com).

T. E. Stern is with the Department of Electrical Engineering, Columbia University, New York, NY 10027 USA (e-mail: tom@ctr.columbia.com).

Publisher Item Identifier S 0733-8716(00)09014-4.

[1]A fiber link failure denotes a failure of *all* working and protection fibers (both directions) on that link.

[2]In SONET rings, maximum detection and switching time is 50 ms.

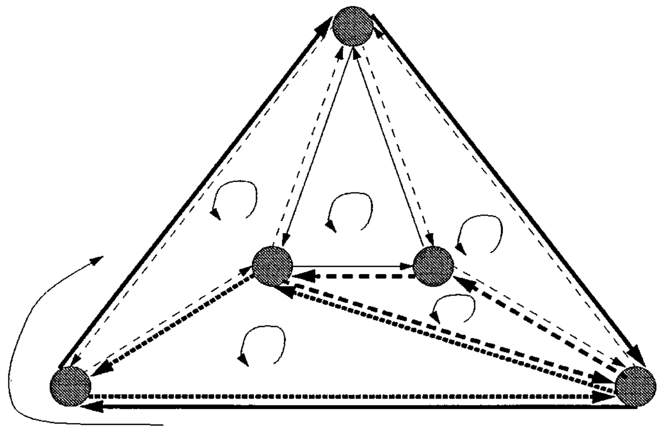


Fig. 1. Directed cycles in a plane graph.

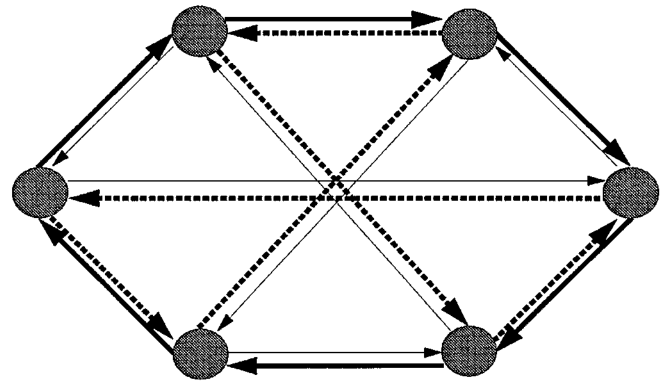


Fig. 2. Directed cycles in the $K_{3,3}$ graph.

This work provides a methodology for protecting any network whose digraph is 2-edge connected (that is, it takes the removal of at least two *bidirectional* edges to break the graph into more than one disconnected component), against the failure of any link. The method proposed also protects 1-edge connected networks (networks containing *bridges*) against any link failure except the failure of a bridge.[3]

The protection methodology proposed is based on interconnecting the protection fibers to create a family of directed cycles, called *protection cycles*, in the following manner [4], [7].

Proposition 1: Recovery from a single link failure in any optical network with arbitrary mesh topology and bidirectional working and protection fiber links can be achieved using APS, if a family of directed cycles using the protection fibers can be found so that: i) all protection fibers are used *exactly once*, and ii) in any directed cycle, a pair of protection fibers are not used in both directions unless they belong to a *bridge*.

The proof of this proposition will become apparent in Section III. Fig. 1 is an example of a (planar) graph with six directed cycles, and Fig. 2 (representing graph $K_{3,3}$) is an example of a (nonplanar) graph with three directed cycles. In both cases, the cycles cover the graph in the manner prescribed in Proposition 1. By appropriately interconnecting the protection switches in the corresponding network, these cycles would be implemented as protection cycles on the corresponding protection fibers.

Note that what is presented in this paper is a scheme for automatic link protection requiring exactly one protection fiber for each edge of the network (independent of the network topology). This is a different approach from the cycle cover methodology presented in [8], where a set of cycles that covers all edges was obtained, and that set of cycles was used as protection cycles. That approach usually requires more protection fibers than network edges. The only case where the approach presented in [8] requires only one protection fiber per edge is the special case of Eulerian networks (that are not typical topologies) which is also addressed in this work.

A. Networks with Planar Topologies

A graph is *planar*, if it can be drawn on the plane in such a way that no two edges intersect (have a common point other than a vertex) [9]. The resultant embedded graph is called a *plane* graph. Any connected planar graph, embedded in a plane, with n vertices ($n \geq 3$) and m edges, has $f = 2 + m - n$ faces [10] (regions defined by the plane graph) where f is denoted as *Euler's number*. The number of faces f includes $(f-1)$ inner faces and one outer face (the unbounded region). The protection cycles are then a set of facial cycles with special orientations.

Proposition 2: Every planar graph G can be decomposed into a family of directed cycles where each edge is used exactly twice (once in each direction), and in each directed cycle that does not include a bridge an edge is used at most once. In each directed cycle including a bridge, the bridge is used twice (once in each direction) in the same directed cycle. (The proof of Proposition 2 is presented in the Appendix).

Thus, for planar graphs, the required set of protection cycles can be obtained by embedding that graph on the plane, identifying the faces of the plane graph, and traversing those faces in a certain direction. The following deals with algorithms that perform these tasks.

1) Planarity Testing—Face Traversal (PTFT) Algorithm: Given a graph G, a planarity testing algorithm has to be invoked to determine whether the graph is planar or not. Furthermore, if the graph is indeed planar, it has to be embedded in the plane and its faces have to be traced in the appropriate directions. It is important to distinguish between knowing a graph is planar and knowing the plane mapping for that graph. The latter problem can be quite difficult if the graph is drawn randomly. For example, the graph presented in Fig. 3(a) is planar, but its embedding in the plane [Fig. 3(b)] is not obvious. The algorithm should provide sufficient information for this embedding to be constructed.

Given a graph G, planarity testing algorithms exist [11]–[14] that test whether graph G is planar. A new algorithm has been developed for this work that tests if a graph is planar and, if it is, the algorithm embeds the planar graph in the plane and traces the faces in the appropriate directions to create a family of cycles with the characteristics defined in Proposition 1. Fig. 4 shows the flowchart of the PTFT algorithm [15]. The PTFT algorithm developed is a variation of the path addition algorithm by Gibbons [13], which is itself a variation of the Demoucron, Malgrange, and Pertuiset path addition algorithm [11]. It is a constructive algorithm that starts from a cycle and adds a single edge (SE) or one edge group (EG) at a time. Each time one SE or

[3]In the rest of the paper, link failure recovery excludes bridge failures.

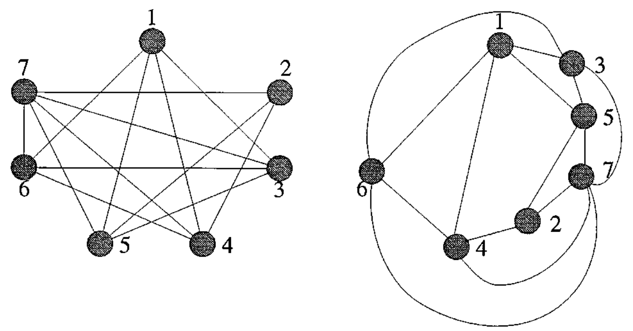


Fig. 3. A planar graph with a not so obvious embedding in the plane.

EG is added, a face is split in two or more faces. The algorithm terminates if it determines that graph G is nonplanar or if the number of faces f equals Euler's number (graph is planar). By adding one or more faces at a time, the graph is automatically embedded in the plane as well. All the edges are bidirectional and in the algorithm both directions of an edge are used in order to construct all faces. Thus, after the plane graph is constructed, the corresponding face tracing is also obtained.

Fig. 5 shows an example of a planar graph with the corresponding faces found using the algorithm. The algorithm was also exercised for a number of nonplanar graphs and it was able to correctly determine that for any given nonplanar graph a plane embedding cannot be found. Fig. 6 shows an example of a nonplanar graph (the Petersen graph) for which the algorithm has concluded that it is not planar.

B. Networks with Nonplanar Topologies

In order to find the protection cycles for a graph with a nonplanar topology, a *cycle double cover* of the graph has to be found. A cycle double cover of a graph G is defined as a cycle decomposition of G such that each edge appears in exactly two cycles [17]. Thus, the set of protection cycles being sought comprise cycle double covers with some additional properties concerning their directions. A crucial conjecture concerning this problem is the following.

Cycle Double Cover (CDC) Conjecture: Let G be a bridgeless finite graph. Then there exists a set of cycles in G such that each edge of G is in exactly two of the cycles (i.e., every bridgeless finite graph G has a cycle double cover).

A cycle double cover is said to be orientable when it is possible to choose a circular orientation for each cycle of the double cover in such a way that each edge is taken in opposite directions in the two incident cycles of the double cover [17]. An orientable cycle double cover is exactly our goal: a cycle decomposition such that each edge appears in exactly two cycles, *and* each edge is used in opposite directions in the two cycles.

Therefore, Proposition 2 applies to all graphs provided that an orientable cycle double cover exists for arbitrary graphs. Certain classical conjectures in graph theory suggest that this is the case [17]. For example, the face boundaries of the plane embedding of a planar graph (oriented properly) constitute an orientable cycle double cover (OCDC). Note that Fig. 2 is also an example of an orientable cycle double cover of a nonplanar graph.

There are limited cases where the validity of the CDC conjecture has been proven. For example, the CDC conjecture was shown to be true for 4-edge connected graphs by a theorem of Jaeger [18], for graphs with Hamiltonian Paths [19], for 2-connected 3-regular graphs containing no subdivision of the Petersen graph [20] and for cubic graphs edge colorable with 3 colors [20]. While the cycle double cover conjecture has never been proven for arbitrary graphs, it was shown in [17] that a minimum counterexample[4] must be a *snark*.[5] Furthermore, Celmins in [21] showed that the minimum counterexample to the CDC conjecture must be a *strong snark*.[6] Finally, it was further proven that a minimum counterexample to the cycle double cover conjecture has girth at least seven [17]. Thus, the minimum counterexample to the CDC conjecture has to be a strong snark of girth at least seven. But, no snark of girth at least seven is known to exist and it was conjectured in [22] that such snarks do not exist (conjectured that every snark has girth at most 6).

Two classes of snarks are shown in Fig. 7. Both examples have girth 5. Obviously, snarks are graphs with unique topologies, and it is not anticipated that telecommunication networks with such topologies will be encountered. So, even if a counterexample to the CDC conjecture does exist, it is highly unlikely that any of the telecommunication networks encountered will be counterexamples. A rule in the design of the network can also be adopted to ensure that this never happens.

1) An OCDC Heuristic: A new heuristic algorithm has been developed in this work to obtain an orientable cycle double cover of any given graph [23]. The heuristic algorithm will also work for graphs with bridges. Obviously, in these graphs the bridges will be traced twice in a single cycle.

Fig. 8 shows the flowchart of the orientable cycle double cover heuristic algorithm. This heuristic is based on a *backtrack* approach that searches the graph in a systematic method in order to obtain a CDC. Every time it has to make a choice for an edge to be included in the current cycle, certain constraints have to be observed. The constraints in choosing an outbound edge ensure that an edge will not be chosen that violates the properties of the cycle double cover. All edges are used in the cycle decomposition, each edge is used only in one cycle (always chosen from a set of unused edges), and both directions of an edge are not used in one cycle unless that edge is a bridge. Furthermore, additional precautions are taken to ensure that bridges attached to cycles are added to these cycles and cycles that are only comprised of the two directions of a bridge are not allowed [23].

The backtrack approach enables the heuristic to "reconstruct" the current cycle if it reaches a vertex where any choice of an outbound edge would violate the constraints presented above. This is done as follows: during the construction of the current cycle, the heuristic keeps in a list all vertices where more than one outbound edge exists that does not violate the constraints. On the top of the list is the latest vertex where such a condition occurs. If the heuristic reaches a vertex where any choice of an

[4]If G is a minimum counterexample to the cycle double cover conjecture, then G is a bridgeless graph with no double cover which has a minimum number of edges among graphs with these properties.

[5]A snark is defined as a cyclically 4-edge-connected cubic graph of girth at least five, which has chromatic index four.

[6] A strong snark is a snark G such that for every edge e, $G * e$ (the unique cubic graph homeomorphic to $G - e$) is not edge colorable with 3 colors.

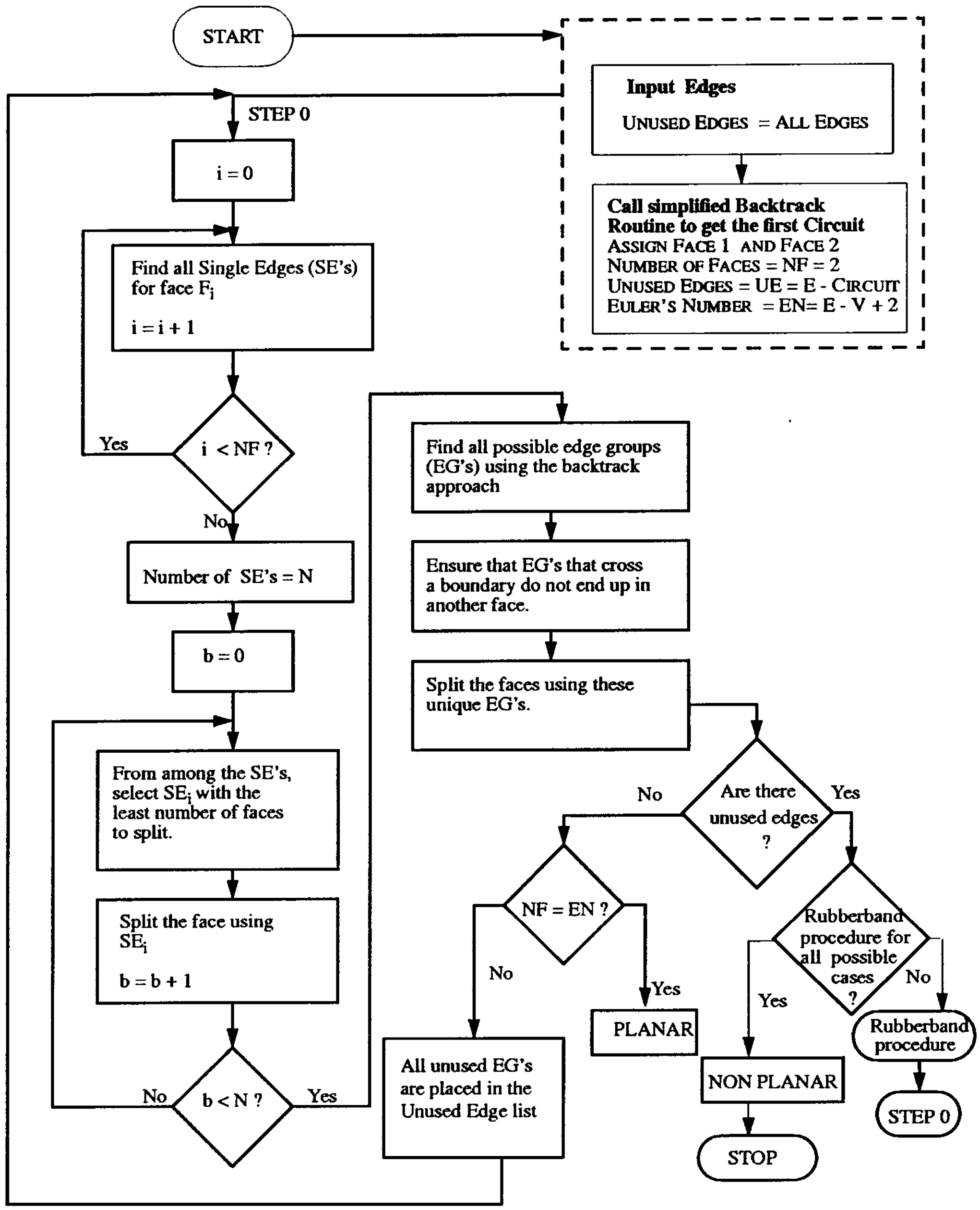
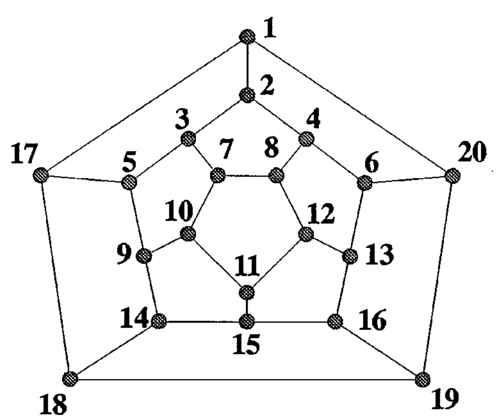


Fig. 4. Flowchart of the PTFT algorithm.

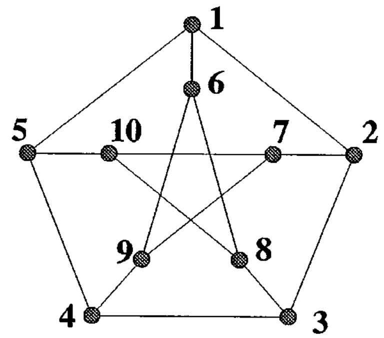


Number of faces = 12

Face[0] = [15 16][16 19][19 18][18 14][14 15]
Face[1] = [16 15][15 11][11 12][12 13][13 16]
Face[2] = [5 3][3 7][7 10][10 9][9 5]
Face[3] = [11 15][15 14][14 9][9 10][10 11]
Face[4] = [20 6][6 4][4 2][2 1][1 20]
Face[5] = [2 4][4 8][8 7][7 3][[3 2]
Face[6] = [5 9][9 14][14 18][18 17][17 5]
Face[7] = [5 17][17 1][1 2][2 3][3 5]
Face[8] = [19 16][16 13][13 6][6 20][20 19]
Face[9] = [12 11][11 10][10 7][7 8][8 12]
Face[10] = [18 19][19 20][20 1][1 17][17 18]
Face[11] = [4 6][6 13][13 12][12 8][8 4]

Fig. 5. Face traversal for the dodecahedron graph.

outbound edge violates the constraints, it takes out of the list the latest vertex with multiple choices of "viable" outbound edges, it chooses another edge that does not violate the constraints, and creates a new cycle from that point on.



Number of faces found = 4

Face[0] = [10 8][8 6][6 1][1 5][5 10]
Face[1] = [3 4][4 5][5 1][1 6][6 8][8 3]
Face[2] = [3 8][8 10][10 7][7 2][2 3]
Face[3] = [3 2][2 7][7 10][10 5][5 4][4 3]

Unused Edges = [1 2][2 1][7 9][9 4][9 6][9 7]
[6 9][4 9]

GRAPH IS NON-PLANAR

Fig. 6. Planarity testing for the Petersen graph.

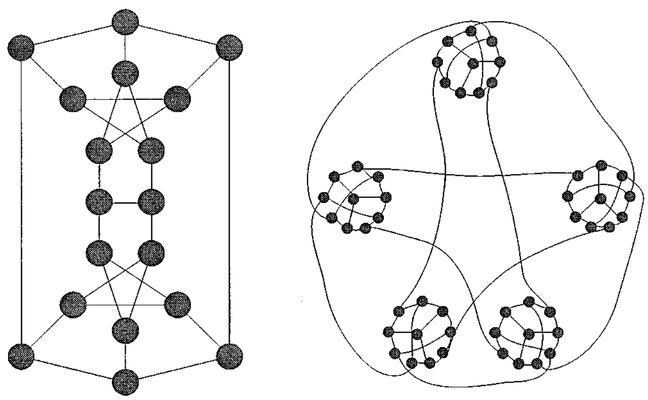


Fig. 7. Two classes of snarks: (a) Blanusa snark (girth = 5), (b) Szekeres' snark (girth = 5).

A final step also ensures that if cycles have already been created that do not enable the algorithm to obtain a cycle double cover of the graph, the heuristic eliminates these cycles and repeats the procedure finding other cycles that can be a part of the cycle double cover of the graph. To do this, the heuristic tries to identify the last vertex where the problem occurs, i.e., the last vertex where a choice of an outbound edge cannot be found. It then eliminates one of the previous cycles that uses this vertex and repeats the procedure to obtain new cycles. This is identical to the backtrack approach explained above, but in this case the heuristic backtracks to previous cycles and not edges. This way, the heuristic can go through all possible combinations of cycles with the characteristics presented in Proposition 1, until it finds a family of cycles that constitutes an orientable cycle double cover.

The backtrack approach used in this work is exponential in the worst case since it may require searching of all possible solutions in a systematic manner. However, on the average it performs much better by pruning the decision tree when it ascertains that traversing that tree in a specific direction is not allowed because it violates constraints imposed on the protection cycles. For the telecommunication networks considered in this work, with only a few hundred switching nodes, the running time of the algorithm was found to be negligible [23]. In addition, since this algorithm will be implemented off-line, during the design of the network, its computational time will not affect the protection process.

First, the heuristic was tested for networks having nonplanar topologies. Fig. 9 shows the 6-cage network and its orientable cycle double cover. The heuristic was also able to find orientable cycle double covers for planar graphs. This is true since the heuristic does not limit itself to nonplanar graphs but is applicable to all types of graphs. Fig. 10 shows a plot of the number of cycles found using the orientable cycle double cover heuristic compared to the number of faces of the plane embedding for various planar networks.[7] As expected, the number of cycles for planar graphs found using this heuristic was less than or equal to Euler's number (number of faces) which is the maximum number of cycles possible. The number of cycles found using the OCDC heuristic, however, was comparable to Euler's number, i.e., the cycles obtained were relatively small in size. Note that the heuristic outputs *elementary cycles* by *peeling off* cycles that pass through a vertex more than once from the cycles in cycle set C. This is an attempt to obtain as large a number of protection cycles as possible, so as to increase the number of simultaneous failures that can be protected (as is later shown in Section IV).

The size of the protection cycles is also important in regards to the quality of the optical signal that traverses them. As stated above, the algorithms try to obtain elementary cycles in order to keep the size of the cycles small. However, there could exist cycles that are relatively large. Obviously, the protection cycles should be engineered in such a way that the optical signal reaches the other side of the protection cycle in a satisfactory condition. Nevertheless, if the protection cycle size is extremely large, this cannot always be achieved. We are currently in the process of trying to aleviate this problem utilizing two distinct approaches. The first approach tries to limit the size of the protection cycles by adding an extra constraint in the OCDC heuristic and the second approach tries to "breakup" the network in smaller subnetworks before it applies the OCDC/PTFT heuristic (thus ensuring that the protection cycles cannot exceed a certain size).

C. Networks with Eulerian Topologies

In the mesh networks considered, all working and protection fiber links are bidirectional. Therefore, by modeling the network (with only the protection fibers accounted for) as a digraph D, each vertex of the digraph will have the same indegree and outdegree $[id(v) = od(v) \; \forall \; v \in V(D)]$ and the degree of each vertex will be even. Since for any two vertices in D a bidirectional walk exists between them, the digraph is also strongly

[7]These networks were obtained by deleting a number of edges from the corresponding nonplanar networks.

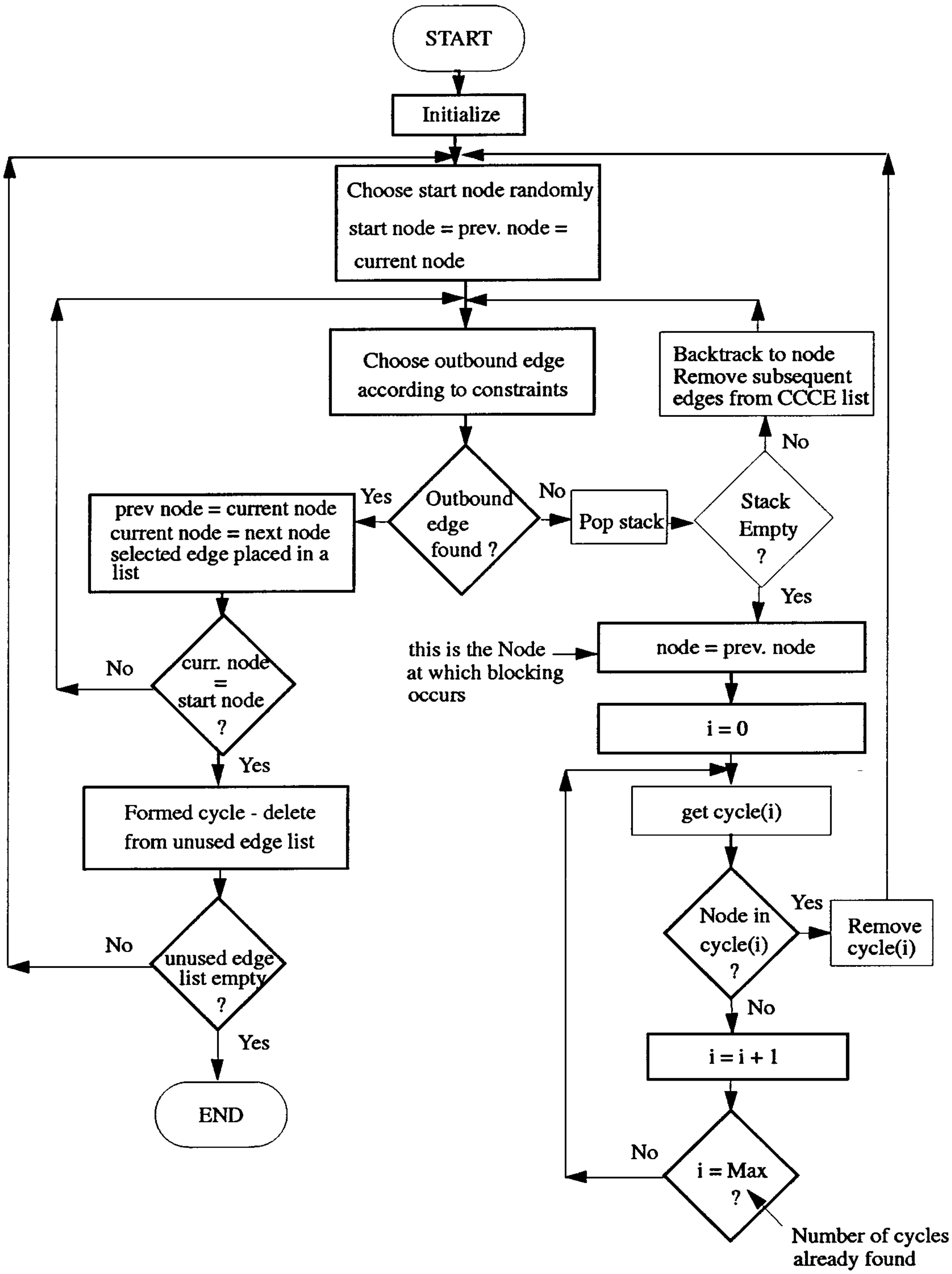


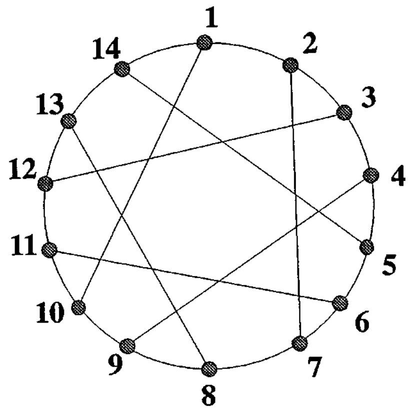
Fig. 8. Flowchart of the OCDC heuristic.

connected. Thus, all networks considered can be modeled as strongly connected Eulerian digraphs.

A cycle decomposition S for any Eulerian digraph can always be obtained [10]. Furthermore, simple algorithms exist on how to perform a cycle decomposition[8] of an Eulerian digraph. However, the family of cycles found will not correspond to the family of protection cycles with the characteristics as defined in Proposition 1, since a cycle decomposition S of an Eulerian digraph does not guarantee that for an edge (which is not a bridge) both directions are not included in a single cycle.

The directed graph D (which represents a model of the original network) is transformed into an undirected graph G by replacing both directions of a bidirectional edge with an undirected edge. If the resulting graph G is Eulerian (denoted as graph G_E), its cycle decomposition S can be used to provide a family of protection cycles. This is true because graph G_E does not have any multiple edges interconnecting two vertices. Thus, a cycle decomposition S of Eulerian graph G_E will not contain an edge in the same cycle twice. When the cycle decomposition

[8]A cycle decomposition S of a digraph D is defined as a set of pairwise arc-disjoint cycles of D such that every arc $a \in A(D)$ belongs to precisely one element of S [10].



Number of Cycles = 7

Cycle[0] = [1 14][14 13][13 12][12 11][11 10][10 1]
Cycle[1] = [8 7][7 2][2 3][3 12][12 13][13 8]
Cycle[2] = [4 3][3 2][2 1][1 10][10 9][9 4]
Cycle[3] = [9 8][8 13][13 14][14 5][5 4][4 9]
Cycle[4] = [9 10][10 11][11 6][6 7][7 8][8 9]
Cycle[5] = [3 4][4 5][5 6][6 11][11 12][12 3]
Cycle[6] = [7 6][6 5][5 14][14 1][1 2][2 7]

Fig. 9. Orientable cycle double cover of the 6-cage network.

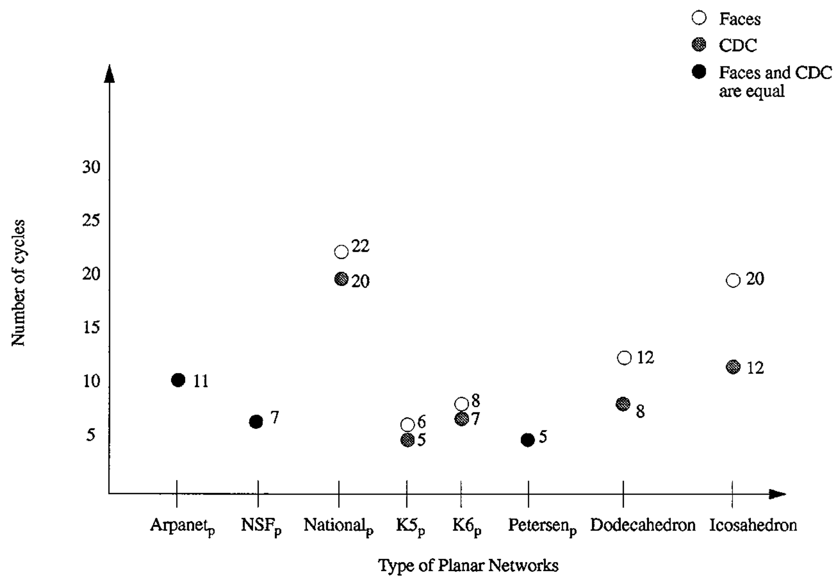


Fig. 10. Number of cycles found for various planar networks using the OCDC/PTFT heuristics.

S of the Eulerian graph G_E is found, each cycle in S is traced twice (once in each direction) to account for the bidirectional edges in digraph D (corresponding to the bidirectional protection fibers in the original network). These double-traced cycles will satisfy the requirements for the protection cycles defined in this paper.

An algorithm was implemented that transforms digraph D to graph G, tests to determine whether graph G is Eulerian, and if it is indeed Eulerian, the algorithm determines an Eulerian circuit (EC) and a cycle decomposition (set of cycles S) for the Eulerian graph G_E [24]. The algorithm used in this work, obtains an Eulerian circuit of graph G_E and traverses that EC, *peeling off* one cycle C after another by deleting certain segments of EC such that the remainder of EC forms an Eulerian circuit for graph $G_E - C$ [25].

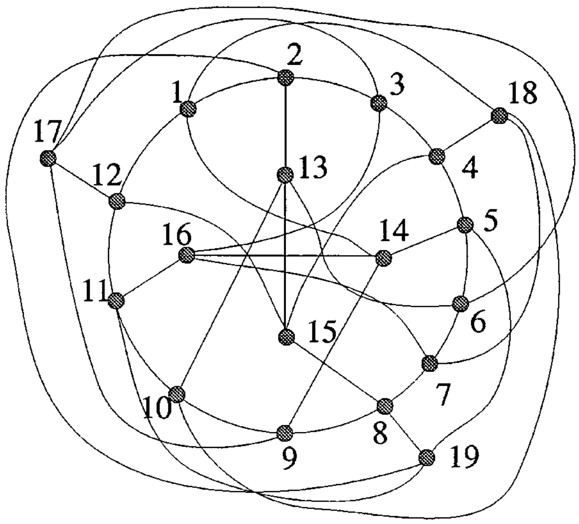
Fig. 11 shows an example of the algorithm for the $\{4, 5\}$-cage graph. The algorithm recognizes that the graph is Eulerian, it finds one of the possible Eulerian circuits, and using that EC, it performs a cycle decomposition of the graph. This cycle decomposition results in a set of 4 cycles. These cycles are consequently double-traced to account for the bidirectional edges in the original digraph. Since this graph is nonplanar, the OCDC heuristic was also applied to get a second (different) set of protection cycles. Using the OCDC heuristic, 9 cycles were obtained. Obviously, the OCDC heuristic will be preferred (for this example), since it allows for more cycles and thus for a possibility of more simultaneous link failure restorations. Note that for a planar network that is also Eulerian, the PTFT algorithm is always preferred since it will result in the maximum number of protection cycles possible (equal to the number of faces in the plane embedding).

III. APS Process Implementation

A. APS System Components

1) Protection Fibers: Each link consists of a pair of unidirectional working fibers (in opposite directions) and a pair of unidirectional protection fibers (in opposite directions). The protection fibers are the ones responsible for carrying the signals around a fault once a failure occurs.

2) Protection Switches: The protection switches used to perform the APS functions are 2×2 switches. These switches can be optomechanical switches or some other type (such



This is an Eulerian graph.
The sequence of the Eulerian Circuit is
[1 2 3 4 5 6 7 8 9 10 11 12 1 18 4 15 8 19 2 13 6 17 3 16 7 18 10 13 15 12 17 9 14 5 19 11 16 14 1]
The cycle decomposition of this graph is :
cycle # 1 [1 2 3 4 5 6 7 8 9 10 11 12 1]
cycle # 2 [18 4 15 8 19 2 13 6 17 3 16 7 18]
cycle # 3 [14 5 19 11 16 14]
cycle # 4 [1 18 10 13 15 12 17 9 14 1]

Fig. 11. Cycle decomposition of (Eulerian) $\{4, 5\}$-cage graph.

as lithium niobate ($LiNbO_3$) based electrooptic switches, acoustooptic, thermooptic, magneto-optic, etc.).

Each bidirectional fiber link has four of these switches associated with it as shown in Fig. 12. A 2×2 protection switch has a working fiber input and output port and a protection fiber input and output port.

During normal operation, the switches are connected as follows: working input port is connected to working output port, and protection input port is connected to protection output port. The ports of different protection switches are interconnected in a way that creates the family of protection cycles described in Section II. These are calculated during the initialization of the network. Fig. 13 shows the interconnection of the protection switches in a seven-node planar network. It corresponds to the five faces of the corresponding plane graph that constitute the set of protection cycles.

B. APS Protocol

Several APS switching commands are issued either automatically or manually as a response to various initiation criteria. These commands modify the settings of each protection switch in order to achieve signal restoration after a fiber link failure.[9]

APS Switching Commands:

Clear (C): This command clears the LP, ASP, and MSP commands and allows the switch to revert back to its default state.

Lockout of Protection (LP): This command prevents switching from any working to protection fiber.

Automatic Switch to Protection (ASP): This command automatically switches from working to protection fiber unless an LP command was previously issued. For a protection switch this means switching from working input port to protection output port and from protection input port to working output port.

Automatic Switch to Working (ASW): This command automatically switches from protection to working fiber.

[9]These commands are similar to the APS switching commands used in point-to-point and self-healing ring (SHR) SONET systems [26], [27].

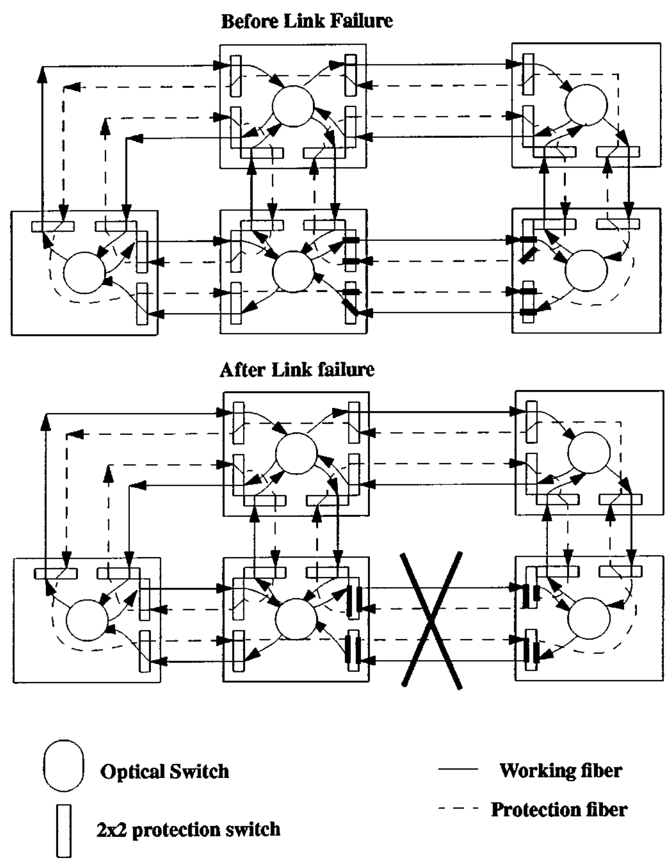


Fig. 12. Protection switch settings before and after a link failure.

Manual Switch to Protection (MSP): This command manually switches from working to protection fiber.

Manual Switch to Working (MSW): This command manually switches from protection to working fiber.

Fig. 14 shows the state diagram for the proposed protection switching process. The protection switch can be in one of six states (default, lockout, engaged, LOS(W) timing, signal on protection fiber (SPF) timing, or LOS(PF) timing). Initially, the protection switch is in the default (enabled) state. If a loss of signal (LOS) is detected at the working fiber for a sustained period of time (longer than a preset timer), the controller issues

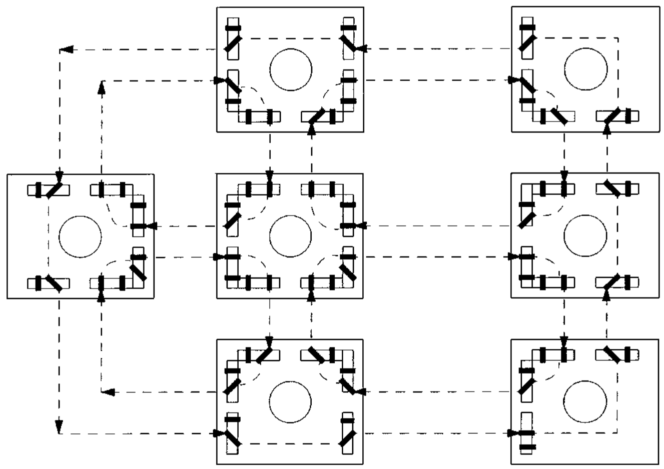


Fig. 13. Protection switch settings in a seven-node planar network.

command ASP and the protection switch engages. The switch will return to its default state when it receives an ASW (for revertive systems), MSW (for nonrevertive systems), or *Clear* command. Similarly, if an optical signal is now detected at the protection fiber (SPF) for a sustained period of time (longer than a preset timer), the protection switch will enter a lockout state and remain there until there is no longer optical power on the protection fiber or a *Clear* command is issued. Optical power on a protection fiber means that the protection switching mechanism has engaged at another location in the network, so this switch should not engage until the other failure has been restored. When there is no longer an optical signal on the protection fiber (for a time longer than a preset timer), this means that the failure was either restored or the protection fiber itself has failed. In either case, the protection switch reverts to its default state. Finally, an MSP command will force the protection switch to engage, provided it is not used in another protection process. This command is used to test the protection switching mechanism, up to and including the final switch.

If only one direction of the fiber is cut, a signaling mechanism is required to carry the APS messages between the network nodes. The signaling messages can share the same physical links as the transport network, or they can use a completely different (physically diverse) network. The former case is much simpler and less expensive, but signaling can be compromised in the case of a failure in the transport network. The latter method, even though more expensive, is more robust, since signaling information is guaranteed to flow in the network even in the event of a failure.

For optically amplified links, a separate supervisory channel using a dedicated wavelength (in-band or out-of-band WDM channel) is being standardized by the ITU [28]. A separate signaling wavelength carrying a low (in the Mb/s range) bit rate signal and using an appropriate protocol can be used in this scheme as well to provide communication between the switching nodes.

C. APS Process

The objective of the APS process is to protect the connections passing through the failed fiber link once a failure occurs. That means routing the signals from the switching node at one side of the failed fiber link to the switching node at the other side of the failed fiber link using redundant protection fibers. Note that the APS process will restore the connections that use the failed link in both directions.

As soon as the fiber link fails, the failure is detected in the switching nodes on both sides of the failed link. The protection switches then switch from working fiber to protection fiber and vice versa. The protection switches associated with the rest of the links in the same switching node do not engage and remain at their original (default) position. Fig. 12 demonstrates the settings of the protection switches in two switching nodes before and after a link failure. By switching the signals from the working fiber to the protection fiber, the failed fiber link is automatically bypassed. When the signals come out of the switching node at one side of the failed link, they are switched onto a protection path (part of a protection cycle), which circumvents the failed link and reaches the other side of the failed link using the protection fibers. Once at the other side of the failed fiber link, they switch from the protection fiber to the working fiber, and follow the same path as before the failure occurred. It is important to emphasize that since the signals eventually switch back to the working fiber, the optical switches do not change their settings because of the failure.

Proof of Proposition 1 is thus trivial. Each bidirectional working link is associated with two protection cycles. Taking one direction only, the working link is associated with a single protection cycle. When the link fails, the protection cycle associated with that link will *always* carry the signal (using the protection fibers) to the other side of the link and thus restore the connection by bypassing the failure.

If the length of the protection cycles is large, one or more multiwavelength optical amplifiers may be needed at appropriate locations in the protection cycles to provide optical signal amplification. When there are no network faults, the protection switches are in their default positions, creating closed paths on the protection cycles. In the absence of an optical signal, the amplifiers on these paths will produce a noise output due in large part to spontaneous emission (ASE noise).[10] This ASE noise can circulate in the closed loop, reaching significant levels, and ultimately leading to amplifier oscillation and saturation [29]. After protection switching is engaged because of a failure, there will be a transition period (time for the ASE noise to exit the network as well as time for the amplifier gains at the signal wavelength to stabilize) during which the ASE noise will interfere and adversely affect the information signal. A number of different solutions have been proposed to eliminate these amplification loops. These include *frogging* algorithms based on Euler network design, *dilation* of the cross-connect switches, *two-sided* network design, and the installation of *circuit breakers* [29], [30].

1) Examples of the Protection Process: Fig. 15 shows a seven-node planar network with a bidirectional connection from S to D. The protection ports at different protection switches are interconnected in such a way (see Fig. 13) that

[10]When there is an optical signal present in the protection fibers, no loops exist because the optical signal enters and leaves the cycle at different points.

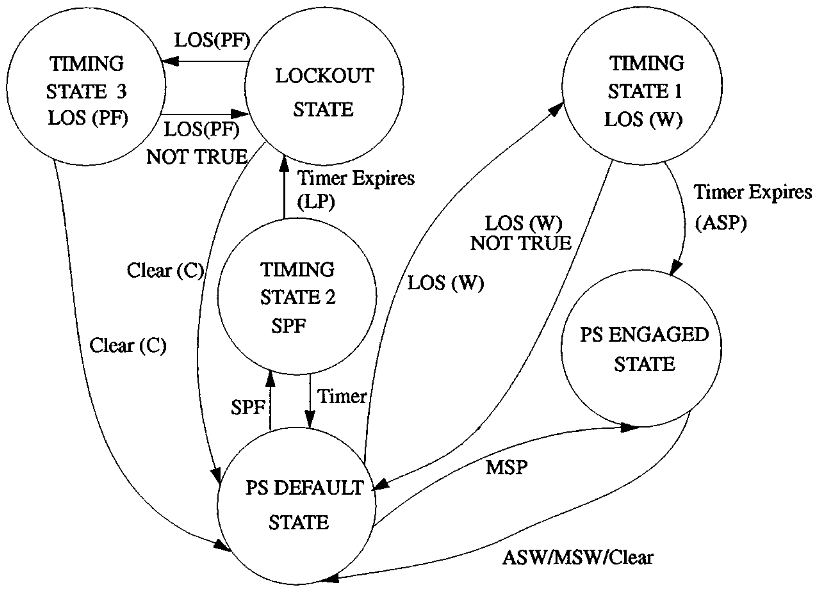


Fig. 14. State diagram of the protection switching process.

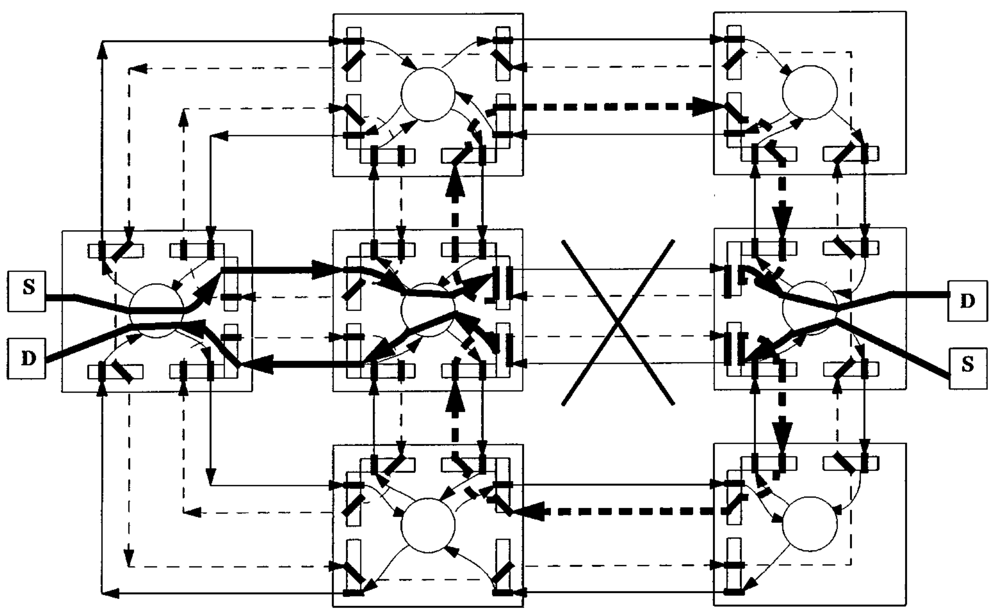


Fig. 15. Restoration of a bidirectional connection in a seven-node planar network after a link failure.

the protection fibers associated with them form a family of directed cycles as defined previously in Section II. Assuming a link failure occurs, the protection switches on both sides of the failed link switch from working to protection and vice versa. The path followed by the signals is shown by a solid dark line up to the failed link. The signals then switch to the protection fiber and follow the dotted dark line until they reach the other side of the failed link. There, they switch back to the working fiber and follow the same path as before the link failure.

Since a family of directed cycles is found using the protection fibers, if two (or more) link failures occur, with all links on distinct protection cycles, all unidirectional connections passing through these links can be protected.

A network based on the (nonplanar) $K_{3,3}$ graph is shown in Fig. 16. Based on a set of protection cycles, the appropriate connections for the protection fibers and switches are indicated. The figure shows failure recovery for (unidirectional) connections $S1$–$D1$ and $S2$–$D2$ after two link failures for the corresponding $K_{3,3}$ network.

D. Scalability Issues

When a network node is added to the network topology, the main consideration is the effect of this addition to the characteristics of the topology (i.e., if it was planar, does it still remain planar, and so on). There are three cases of interest here. In the first case, a node is added to a planar topology T_1, and the resulting graph is also planar (with topology T_2). The addition of the node (with its corresponding edges) disturbs only one of the faces of topology T_1. The maximum number of affected nodes is the number of nodes on that face. Thus, only the protection switches in the nodes lying at the boundary of the affected face

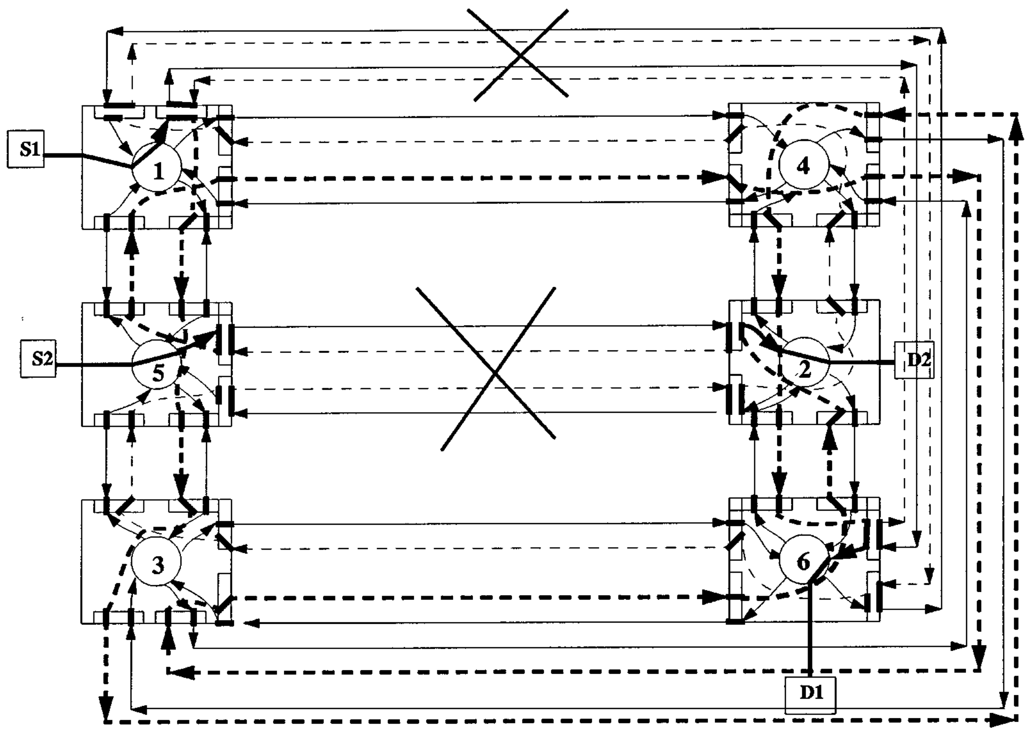


Fig. 16. Restoration of two unidirectional connections in the $K_{3,3}$ network after two link failures.

will have to change their interconnections to account for the new faces created. All other network nodes will be unaffected by the node addition as far as survivability is concerned. The maximum number of affected nodes in a graph will be equal to the maximum face degree $d_{\max}(F)$.[11]

If the addition of a node to a planar topology T_1 results in a nonplanar topology T_2, the protection cycles will not correspond to the faces of the plane graph anymore, but they will have to be recalculated using the OCDC heuristic developed in this work. Similarly, in the third case when a nonplanar topology T_1 scales to a nonplanar topology T_2, the protection cycles will again have to be recalculated.

IV. Multiple Link Failures

A. Bounds for Planar Topologies

As illustrated in Section II-A, planar graphs can be decomposed into directed cycles by first embedding the planar graph in a plane. The number of such cycles equals the number of faces f in a plane graph ($f = c = m + 2 - n$). It is trivial to show that for a planar graph this is the maximum number of directed cycles that can be found with the properties described in Proposition 1. Assuming bidirectional connections, every time a link fail, it uses *at most* two protection cycles. If the link that failed corresponds to a bridge, then only one protection cycle is used. Otherwise, a link failure will affect the two protection cycles that use that link. If another link in these two protection cycles fails, recovery from both failures is not possible. Therefore, the maximum number of link failure recoveries possible is

$$P^b_{\max} = \left\lfloor \frac{f}{2} \right\rfloor. \quad (1)$$

[11]The degree of a face $d(F)$ is the number of edges bounding the face F and $d_{\max}(F) = \max\{d(F_i)\}\ \forall F_i \in G$.

In the worst-case scenario ($f = 2$), only one fault can be restored. As an example, in the dodecahedron network with 12 faces shown in Fig. 5, this approach can protect all the bidirectional connections passing through at most 6 simultaneous link failures.

The above formula is valid only when every connection in the network is bidirectional. If every connection in the network is unidirectional, then the maximum number of link failure recoveries possible is

$$P^u_{\max} = f - 1. \quad (2)$$

Clearly, the best scenario occurs when all the inner faces have a common adjacent face, namely the outer face. $(f - 1)$ unidirectional connections can then be simultaneously restored if the failures occur at the $(f - 1)$ edges that are common to both the inner and outer faces. Again, in the worst case (only two directed cycles found), only one fault can be restored.

B. Bounds for Nonplanar Topologies

For nonplanar graphs with an S protection cycle decomposition, the following conjectures provide a bound on the number of cycles S, and thus a bound on the number of possible link failures that can be simultaneously restored.

The short cycle double cover (SCDC) conjecture [31] states that every 2-edge connected simple graph G of order n has a CDC with fewer than n cycles (holds for all maximal and Hamiltonian planar graphs). Thus, the maximum number of link failures that can be simultaneously restored for bidirectional connections is

$$NP^b_{\max} = \left\lfloor \frac{n-1}{2} \right\rfloor. \quad (3)$$

For practical purposes, however, the maximum number of link failures that can be simultaneously restored for bidirectional connections equals $\lfloor S/2 \rfloor$, where S is the number of protection cycles found using the OCDC heuristic. For example, in the 6-cage network (Fig. 9) with 7 protection cycles found, this approach can protect all the bidirectional connections passing through at most 3 simultaneous link failures. This is true since a link failure only affects the two protection cycles that use that link.

Similarly, for unidirectional connections, the maximum number of link failures that can be simultaneously restored is

$$NP_{\max}^{u} = n - 2. \tag{4}$$

This is true since both the order of a graph (n) and the number of cycles in the OCDC are integers and thus the maximum value of S (according to the SCDC conjecture) is $n - 1$.

Again, for practical purposes, the maximum number of link failures that can be simultaneously restored for unidirectional connections equals $S - 1$. For example, in the 6-cage network, this approach can protect all the unidirectional connections passing through at most 6 simultaneous link failures.

For specific graphs, i.e., cubic graphs, the upper bound shown in (3) becomes even more tight. Bondy in [32] conjectured that if G is a 2-connected simple cubic graph of n vertices ($n \geq 6$), then G admits a CDC consisting of *at most* $(1/2)n$ cycles. This conjecture is sharp. The Petersen graph, for example, cannot be covered with fewer than 5 cycles. This was confirmed by the OCDC heuristic explained in this work.

C. Bounds for Eulerian Topologies

The maximum number of cycles partitioning the edge set of an Eulerian graph embedded in the plane is equal to the maximum number of cycles of an Eulerian partition without crossings and is equal to the number of faces [33]. Thus, planar Eulerian graphs are not considered here as they will give results identical to Section IV-A. Only bounds for nonplanar Eulerian graphs are considered.

Eulerian topologies can be decomposed into a family of cycles, such that each cycle does not have any common edges with the other cycles. To account for the original digraph D with its pairs of directed edges, each cycle in the cycle decomposition is double-traced. Thus, the maximum number of multiple failures that can be simultaneously restored is equal to the number of cycles in the cycle decomposition. A different number of cycles can be obtained for different Eulerian circuits.

The problem of finding all the Eulerian circuits of an arbitrary Eulerian graph is related to the more general optimization problem known as the Chinese postman problem (CPP) [34] (to find the shortest closed walk such that each edge is traversed at least once), which was shown to be NP-complete [35]. In Section II-C, a single EC and cycle decomposition are found for a given Eulerian graph G_E, and no attempt is made to find all the possible EC's and corresponding cycle decompositions.

However, an upper bound can be obtained from Hajos' conjecture which states that a simple Eulerian graph of order n has a cycle decomposition into at most $\lfloor (n-1)/2 \rfloor$ cycles [31]. Thus, the maximum number of link failures that can be simultaneously restored (bidirectional or unidirectional connections) is

$$E_{\max} = \left\lfloor \frac{n-1}{2} \right\rfloor. \tag{5}$$

V. Conclusion

This paper proposed a general methodology for performing APS for link failures in optical networks with arbitrary mesh topologies and bidirectional links. It introduced the concept of *protection cycles* and developed algorithms to obtain these cycles in networks with planar, nonplanar, and Eulerian (planar/nonplanar) topologies. A complete solution was presented for planar and Eulerian networks. The method was also extended to nonplanar networks, and an orientable cycle double cover heuristic was developed that obtained the required family of protection cycles. Bounds on the number of possible simultaneous link failure recoveries showed that, depending on the position of the failures, a large number of simultaneous restorations may be possible.

The paper also demonstrated the implementation of the APS technique by showing that the protection switches located at the ends of the network links could be interconnected according to the identified protection cycles to ensure protection from a link failure in a comprehensive, conflict free system. This was achieved by rerouting signals from working to protection channels, using the protection switches, which were activated immediately after a fault was detected.

The protection process is independent of the source–destination connections currently on the network and is transparent to the rest of the network. Only the network nodes that are attached to the ends of the failed link engage their protection mechanisms as a response to the link failure. Therefore, the fault recovery system is distributed, autonomous, and network state independent.

Appendix

The Proof of Proposition 2 is embodied in the three claims that follow.

Claim 1: Every plane graph G that is k-edge connected, ($k \geq 2$) with n vertices ($n \geq 3$) and m bidirectional edges, can be decomposed into a family of directed cycles where each edge is used exactly twice (once in each direction), and in each directed cycle an edge is used at most once.

Proof: A plane graph G which is k-edge connected ($k \geq 2$) with n vertices ($n \geq 3$) and m bidirectional edges consists of f faces. For each inner face, a facial directed cycle can be formed using all the edges bounding that face in their counterclockwise direction. Since for plane graphs each inner face has only one edge in common with an adjacent inner face, and for both faces directed facial cycles are formed, then their common edge is used twice. By creating both facial directed cycles in the counterclockwise direction, however, it is ensured that their common edge is used once in each direction. Thus, all "inner edges" (edges that are part of inner faces only) are used exactly twice (both directions) while ensuring that each cycle uses an edge in only one direction. For the outer face, a facial directed cycle can be formed using all the outer edges in their clockwise direction. This way, all "outer edges" (edges that are part of inner faces and the outer face) are also traversed twice (both directions): once when the inner facial directed cycles are created (in one direction), and once when the outer facial directed cycle is formed (in the other direction).

Claim 2: Every plane graph G which is 1-edge connected, with n vertices and m bidirectional links, can be decomposed into a family of directed cycles where each edge is used exactly twice (once in each direction), and in each directed cycle which does not include a bridge an edge is used at most once. In each directed cycle including a bridge, the edge defined as a bridge is used twice (once in each direction) in the same directed cycle.

Proof: The proof for the second claim follows directly from the proof for Claim 1. The only difference here is the existence of bridges that lie in inner faces or the outer (unbounded) face. For inner faces that have bridges lying in their bounded region, the counterclockwise inner facial cycles will include these edges. If the outer face has bridges lying in its unbounded region, its clockwise outer facial cycle will also include these edges. All bridges will be traversed twice (both directions) in the inner (counter-clockwise) or outer (clockwise) facial cycles.

Claim 3: Every planar graph G which is 1-edge connected, with n vertices and m bidirectional links, can be decomposed into a family of directed cycles with the characteristics defined in Claim 2.

Proof: A graph is planar if it has an embedding in the plane. Since the validity of Claim 3 has already been proven for plane graphs, then by the definition given above, it is true for planar graphs.

REFERENCES

[1] W. D. Grover, "Selfhealing networks: A distributed algorithm for k-shortest link-disjoint paths in a multi-graph with applications in real time network restoration," Ph.D. dissertation, Univ. Alberta, 1989.

[2] G. Ellinas, A. Hailemariam, and T. E. Stern, "Protection of a priority connection from an optical switch failure in mesh networks with planar topologies," in *Proc. IEEE/OSA Optical Fiber Commun. Conf.*, San Diego, CA, Feb. 1999.

[3] ——, "Optical switch failure protection in mesh networks with arbitrary nonplanar architectures," in *Proc. IEEE Globecom*, Rio de Janeiro, Brazil, Dec. 1999.

[4] G. Ellinas, "Fault restoration in optical networks: General methodology and implementation," Ph.D. dissertation, Columbia Univ., 1998.

[5] "Transport systems: Generic requirements," Tech. Rep. TR-NWT-000499, Bell Commun. Res., Red Bank, NJ, Dec. 1993.

[6] M. Berger, M. Chbat, A. Jourdan, M. Sotom, P. Demeester, B. Van Caenegem, P. Godsvang, B. Hein, M. Huber, R. Marz, A. Leclert, T. Olsen, G. Tobolka, and T. Van den Broeck, "Pan-European optical networking using wavelength division multiplexing," *IEEE Commun. Mag.*, vol. 35, no. 4, pp. 82–88, Apr. 1997.

[7] G. Ellinas and T. E. Stern, "Automatic protection switching for link failures in optical networks with bi-directional links," in *Proc. IEEE GLOBECOM*, London, U.K., Nov. 1996, pp. 152–156.

[8] L. M. Gardner, M. Heydari, J. Shah, I. H. Sudborough, I. G. Tolis, and C. Xia, "Techniques for finding ring covers in survivable networks," in *Proc. IEEE GLOBECOM*, San Francisco, CA, Nov. 1994, pp. 1862–1866.

[9] M. Aigner, *Graph theory—A development from the 4-color problem*. Moscow, ID: BCS Assoc., 1986.

[10] H. Fleischner, *Eulerian Graphs and Related Topics Part I: Vol. 1*. Amsterdam, The Netherlands: Elsevier Science, 1990.

[11] G. Demoucron, Y. Malgrange, and P. Peruiset, "Graphes planaires," *Revue Francaise de Rech. Operat.*, vol. 30, pp. 33–47, 1964.

[12] S. Even, *Graph Algorithms*. Rockville, MD: Computer Science Press, 1979.

[13] A. Gibbons, *Algorithmic Graph Theory*. Cambridge, U.K.: Cambridge University Press, 1985.

[14] A. Lempel, S. Even, and I. Cederbaum, "An algorithm for planarity testing of graphs," in *Proc. Theory of Graphs—Int. Symp.*, Rome, Italy, July 1966, pp. 215–232.

[15] G. Ellinas, A. Hailemariam, and T. E. Stern, "Planarity testing of a graph—Creation and traversal of faces," Tech. Rep. CU/CTR/TR 474-97-07, Center Telecommun. Res., Columbia Univ., 1997.

[16] M. Behzad, G. Chartrand, and L. Lesniak-Foster, *Graphs and Digraphs*. Belmont, CA: Wadsworth International Group, 1981.

[17] F. Jaeger, *Cycles in Graphs*, Amsterdam, The Netherlands: Elsevier Science, 1985.

[18] ——, "Flows and generalized coloring theorems in graphs," *J. Combinatorial Theory*, vol. 26, no. 2, pp. 205–216, Apr. 1979.

[19] M. Tarsi, "Semi-duality and the cycle double cover conjecture," *J. Combinatorial Theory*, vol. 41, no. 3, pp. 332–340, Dec. 1986.

[20] P. A. Catlin, "Double cycle covers and the Petersen graph," *J. Graph Theory*, vol. 13, no. 4, pp. 465–483, 1989.

[21] U. Celmins, "On conjectures relating to snarks," Ph.D. dissertation, Univ., Waterloo, 1987.

[22] F. Jaeger and T. Swart, *Combinatorics 79*. Amsterdam, The Netherlands: Elsevier Science, 1980.

[23] G. Ellinas, A. Hailemariam, and T. E. Stern, "Creation of a family of cycles corresponding to the orientable cycle double cover," Tech. Rep. CU/CTR/TR 474-97-08, Center Telecommun. Res., Columbia Univ., 1997.

[24] G. Ellinas, S. Rong, and T. E. Stern, "Protection cycles in mesh networks with Eulerian graphs," Tech. Rep. CU/CTR/TR 491-98-25, Center Telecommun. Res., Columbia Univ., 1998.

[25] R. C. Read, *Theory of Graphs*. New York: Academic, 1969.

[26] "Automatic protection switching for SONET," Tech. Rep. SR-NWT-001756, Bell Commun. Res., Red Bank, NJ, Oct. 1990.

[27] "Synchronous optical network (SONET) transport sytems: Common generic criteria," Tech. Rep. TR-NWT-000253, Bell Commun. Res., Red Bank, NJ, Dec. 1991.

[28] *Architecture for Optical Transport Networks*: ITU-T Recommendation G.872, 1999.

[29] K. Bala and C. A. Brackett, "Cycles in wavelength routed optical networks," *IEEE/OSA J. Lightwave Technol.*, vol. 14, pp. 1585–1594, July 1996.

[30] J. Iness, B. Ramamurthy, B. Mukherjee, and K. Bala, "Elimination of all-optical cycles in wavelength routed optical networks," *IEEE/OSA J. Lightwave Technol.*, vol. 14, pp. 1207–1217, June 1996.

[31] C.-Q. Zhang, *Quo Vadis, Graph Theory?*. Amsterdam, The Netherlands: Elsevier Science, 1993.

[32] J. A. Bondy, *Cycles and Rays*. Norwell, MA: Kluwer Academic, 1990.

[33] M. L. Vergnas, *Graph Theory and Combinatorics*. New York: Pitman, 1979.

[34] J. P. Shen and J. P. Hayes, "Fault-tolerance of dynamic-full-access interconnection network," *IEEE Trans. Comput.*, vol. C-33, pp. 409–416, Mar. 1984.

[35] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to NP-Completeness*, W. H. Freeman, Ed. New York, 1979.

Georgios Ellinas (S'90–M'98) received the B.S., M.S., and Ph.D. degrees all in electrical engineering from Columbia University, New York, in 1991, 1993, and 1998, respectively.

From 1993 to 2000, he was with the Optical Networking Research program at Telcordia Technologies (formerly Bellcore), Red Bank, NJ for both the Optical Networks Technology Consortium (ONTC) and Multiwavelength Optical Networking (MONET) projects. In 1998, he joined Telcordia Technologies as a full-time Research Scientist in the Optical Networking Research Group. In 1999, he served as an Adjunct Assistant Professor at Columbia University and in 2000 he served as an Adjunct Assistant Professor at the University of Maryland (Baltimore County) teaching a course on Multiwavelength Optical Networking. He is currently a Senior Network Architect with Tellium, Incorporated, working on lightpath provisioning and fault restoration algorithms in mesh networks. His research interests are in the areas of architectures, routing and wavelength assignment algorithms, fault protection/restoration techniques in arbitrary mesh optical networks and packet switching in multiwavelength optical networks. He is currently working on Optical-Label Switching techniques for a DARPA NGI project.

Dr. Ellinas was awarded a Fullbright fellowship (1987–1991) for undergraduate studies at Columbia University. He has authored and coauthored more than 35 conference and journal papers. He is also the holder of two U.S. Patents on optical networking, and has nine U.S. patent applications currently pending.

Aklilu Gebreyesus Hailemariam received the B.S.E.E. degree from Addis Ababa University, the M.S.E.E. degree from Columbia University, New York, and the M.S.C.S. degree from City College of New York. He is currently pursuing the Ph.D. degree at Columbia University.

From 1995 to 1997, he was a Research Assistant at Columbia University working on optical network architectures in collaboration with the National Transparent Optical Network Consortium (NTONC) project. From 1997 to 2000, he worked at Lucent Technologies designing management systems for SONET and DWDM equipment. He is currently designing next generation optical network equipment at Tellium Incorporated. His main research interests are in the design and management of survivable optical networks.

Thomas E. Stern (S'54–M'57–SM'67–F'72–LF'95) received the B.S. and M.S. degrees in 1953 and the Sc.D. degree in 1956, all in electrical engineering, from the Massachusetts Institute of Technology, Cambridge.

He served in the U.S. Air Force Research and Development Command from 1956–1958, and joined Columbia University in 1958. At present he holds the title of Dicker Professor Emeritus at Columbia. He served in various leadership capacities at Columbia including Chairman of the Department of Electrical Engineering, Technical Director of the Center for Telecommunications Research (CTR), and leader of the CTR's Lightwave Networks Research Group. He holds several patents in communications and has authored two textbooks and more than 100 papers in networks and related fields.