A Network Management Architecture for Robust Packet Routing in Mesh Optical Access Networks

Muriel Médard, *Senior Member, IEEE*, Steven Lumetta, *Member, IEEE*, and Liuyang Li

Invited Paper

***Abstract*—We describe an architecture for optical local area network (LAN) or metropolitan area network (MAN) access. The architecture allows for bandwidth sharing within a wavelength and is robust to both link and node failures. The architecture can be utilized with an arbitrary, link-redundant mesh network (node-redundancy is necessary only to handle all node failures), and assumes neither the use of a star topology nor the ability to embed such a topology within the physical mesh. Reservation of bandwidth is performed in a centralized fashion at a (replicated) head end node, simplifying the implementation of complex sharing policies relative to implementation on a distributed set of routers. Unlike a router, however, the head end does not take any action on individual packets and, in particular, does not buffer packets. The architecture thus avoids the difficulties of processing packets in the optical domain while allowing for packetized shared access of wavelengths. In this paper, we describe the route construction scheme and prove its ability to recover from single link and single node failures, outline a flexible medium access protocol and discuss the implications for implementing specific policies, and propose a simple implementation of the recovery protocol in terms of state machines for per-link devices.**

***Index Terms*—Access networks, local area networks, network recovery, optical networks.**

I. Introduction

OUR MOTIVATION is to create an architecture that provides low-cost access to optical bandwidth in a flexible, efficient, and robust manner. We consider the case in which certain wavelengths within a local or metropolitan area network are reserved for access by access nodes that share bandwidth. We propose a network management architecture that manages routes and bandwidth access in a way that is robust to link or node failures while allowing significant flexibility in terms of bandwidth allocation.

The two main elements of our network management architecture are the establishment of routes and the recovery mechanism in case of link failure or node failure. While we present a general type of bandwidth access protocol that can be used with our network management, different types of reservation and scheduling schemes can be used in combination with our network management. The main characteristics that distinguish our architecture from previous work are:

- The architecture implements, through appropriate routes, a local area network (LAN) or metropolitan area network (MAN) over an arbitrary redundant mesh topology, rather than over a star or multiple rings. In particular, our network can utilize an arbitrary link-redundant subgraph of a full network graph, such as a portion of a metropolitan area network. Fig. 1 shows an example in which two groups of nodes share a wavelength (the thick lines) on a single physical network (the thin lines).
- The network management architecture realizes recovery using preplanned rerouting in the case of a link or node failure. While the routes change dynamically, the network takes, for every failure, actions which are predetermined by the network management. Routing and recovery are closely intertwined: the route is constructed to enable recovery and recovery is effected in part through preplanned rerouting.
- Our network management is compatible with lightpath routes. Thus, each shared subnetwork in Fig. 1 requires only a pair of duplex wavelengths in a wavelength division multiplexing (WDM) system.
- We consider the case where each shared subnetwork carries traffic that can fit within a single wavelength.

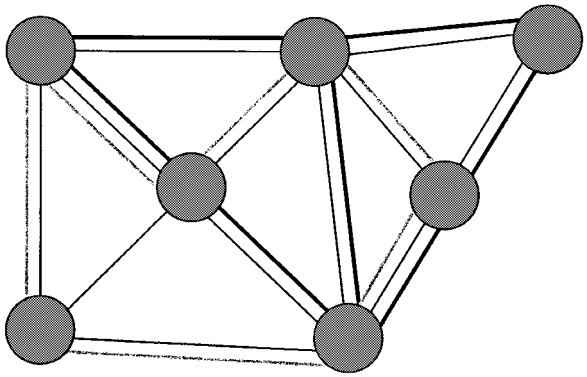

Fig. 1. Implementation of bandwidth sharing on subnetworks of a physical topology.

Manuscript received March 8, 2001; revised December 20, 2001. The work was supported in part by the Defense Advanced Research Projects Agency under Grant MDA972-99-1-0005. This paper was presented in part at the LEOS Summer Topical Meetings, LOCATION, July 2000.

M. Médard is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139 USA (e-mail: medard@mit.edu).

S. Lumetta is with the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: lumetta@uiuc.edu).

L. Li is with the Coordinated Science Laboratory, University of Illinois Urbana-Champaign, Urbana, IL 61801 USA (e-mail: lli@uiuc.edu).

Publisher Item Identifier S 0733-8716(02)05075-8.

Currently, per-wavelength rates reach 10 Gb/s for OC-192, and 40 Gb/s per wavelength systems have been demonstrated and are in commercial development. Enterprise routers currently offer throughputs of the order of tens of Gb/s. Thus, it is reasonable to assume that a single wavelength can carry the traffic of certain enterprise networks or virtual private networks.

The remainder of the paper is organized as follows. Section II provides an overview of the background literature relevant to optical local area networks. In Section III, we describe the main features of our network management architecture: routes and associated recovery mechanisms that provide us with a means of recovering from link or node failures. In Section IV, we outline an access protocol for use with our network management architecture. The protocol allows us to share wavelengths in a flexible, bandwidth-efficient, and fair manner. In Section V, we discuss implementation issues and, finally, present our conclusions and areas for further research in Section VI.

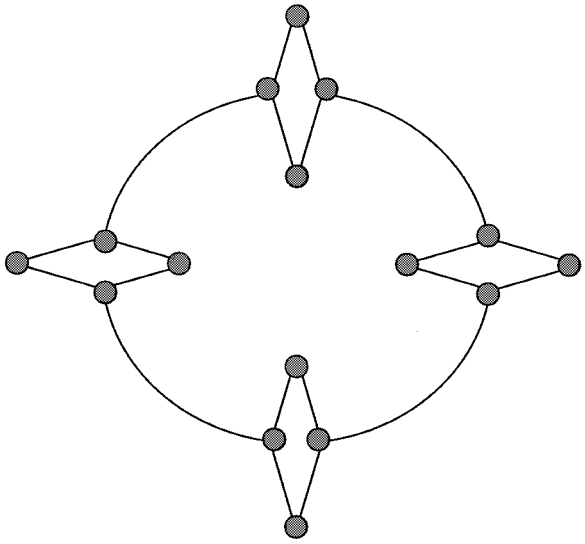

Fig. 2. Example of a topology whose nodes cannot be included in a single star or an Eulerian tour.

II. Background

In this section, we briefly overview previous work in topics that are relevant to our architecture. In particular, we consider the following topics: topologies for optical LANs and MANs; folded bus schemes; redundant tree routes; and access protocols for optical LANs and MANs. There has been significant work in the area of optical LANs and MANs using WDM. The vast majority of the proposed architectures consider star topologies, where some type of switch, router, or other type of hub, is placed in the center of a topology and each node is directly connected to the hub [6]–[9], [15], [17]–[20], [23], [24], [26], [31], [33], [36], [42], [44], [47], [51], [53], [54], [56], [61]–[63], [72], [79]. These star architectures usually involve a passive optical broadcast star. These stars generally have senders and/or receivers that are tunable over the whole spectrum or a subset of the spectrum. Since the topology is very simple, the literature treating stars is generally concerned with issues of scheduling, which we do not address in this paper. We consider a scheduled system but do not specify the algorithm for scheduling and possible reservations. Another topology alternative involves rings, such as fiber distributed data interface (FDDI) [37], [59], [60]. Multiple ring topologies may be interconnected through a hub [30], or rings may coexist in a logically interconnected fashion over a single physical ring [48], [49], [50], or rings may be arranged hierarchically [3], [30], [41].

Our topology considers arbitrary link or node redundant topologies, as is detailed in the next section. The extension from star or ring topologies to mesh topologies is not trivial. To illustrate this point, consider Fig. 2. The nodes in the topology shown cannot be covered by a single star, or ring, or by rings interconnected through a hub.

Our routing scheme is used in a particular type of folded bus, as well as in redundant broadcast trees, which may be viewed as extensions of dual buses. Extensive analysis of folded bus schemes, such as distributed queue dual bus (DQDB) , [4], [10], [11], [25], [28], [32], [46], [58], [69], [74], [77] has been carried out. Dual bus schemes have also been analyzed extensively [25], [64]–[66], [77], [78]. Analysis has also been carried out for other bus schemes, such as cystic reservation multiple access (CRMA) [27], [55], [66], [70], which can use either dual buses or a folded bus, and for optical bus schemes [34] such as helical (HLAN) [2], [16], [57], and optical reservation multiple access (ORMA) [21]. Besides these main bus protocols, there exist a variety of alternative bus schemes [12], [40], [43], [67], [68], [73], [75], [76]. The analysis for folded and dual buses is almost entirely concerned with issues of bandwidth allocation, such as fairness and bandwidth efficiency. This analysis is not directly pertinent to our research, as we do not specify a particular scheduling or reservation scheme. The main aspects of the construction of our folded bus are that the bus may be overlaid over any redundant mesh network and that that the bus is constructed with the goal of being robust to a single link or node failure. None of the work referenced above is concerned with such aspects of robustness.

Besides a folded bus route, our network management architecture also uses a route based upon redundant trees, i.e., pairs of trees in which each node is connected to at least one tree root even after failure of a single link or node. Such pairs of trees were first introduced in [29], [80], using s–t numberings [39], and a more general method of constructing them was given in [52]. This paper extends the work in [52] to permit separation of the two tree roots, which allows the possibility of recovering from the failure of either root node.

For the access protocol within our network management architecture, we address only the mechanism by which nodes can transmit and receive. As mentioned before, our goal is not to establish a particular scheduling or reservation scheme. Selecting the particular implementation of scheduling or reservation is best done when particular performance metrics, such as fairness, bandwidth efficiency, or delay are considered. The choice of appropriate metrics, in turn, depends crucially on the applications for our architecture, a discussion of which lies outside the scope of this paper. Scheduling has been considered extensively in the literature addressing optical stars and buses (referenced above), as well as for optical rings [81], [82], optical switches [1], [38],

[45], [71], and WDM networks with arbitrary topologies [22]. Another protocol aspect that we do not consider in this paper is the specific structure of the signaling for a control channel. Several methods and their performance, in particular in terms of scalability with the number of nodes and of delay, have been presented in [5], [13], [14], [33], [61] . In this paper, we do not address the issue of how transmissions are scheduled. A very large body of literature deals with scheduling in star networks. While we consider timing issues in our protocol, implementation issues such timing recovery and ranging of nodes are outside the scope of this paper.

In the rest of the paper, we present the features of our network management architecture: routing for robustness to link or node failures on mesh networks, access protocol for flexible use of bandwidth, and a simple but flexible implementation, in terms of state machines, of our network management.

III. Robustness of Routing

We consider both link failures and node failures. For each case, we first describe how the route is performed when there are no failures and how the route is modified to recover from a failure. Next, we present a protocol that correctly implements the routes.

We describe an algorithm for route construction using access nodes in such a way that recovery is possible even in the event of any link failure. Our route consists of two parts: a "collection" portion of the route and a "distribution" portion. The collection portion allows all nodes to place their traffic on the access wavelength(s) and the distribution portion ensures that packets can reach all nodes.

A. Route Construction

Consider a link-redundant mesh network on which a wavelength is to be shared. The network can be a subnetwork of another, larger network, but the links in the subnetwork must all be duplex, and the subnetwork must be link-redundant, meaning that all nodes remain connected when any single link fails.[1] Let the directed graph $G = (N, A)$ represent the network on which the architecture must operate. The graph G consists of a set N of vertices and a set A of directed arcs. Every node in the network corresponds to a vertex in the graph, and every link in the network corresponds to a pair of arcs in the graph. For a link connecting the nodes corresponding to the vertices i and j, we create the arcs (i, j) and (j, i). Failure of a network link removes the two corresponding arcs in the graph, and failure of a node removes the corresponding vertex and all arcs incident upon that vertex (i.e., the arcs corresponding to all links incident upon the node in the network).

We describe an algorithm for route construction using access nodes in such a way that recovery from any single link or node failure is possible. The route consists of two elements: a collection portion and a distribution portion. Nodes place packets on the access wavelength in the collection portion of the route, and the distribution portion delivers all packets to all nodes.

[1]Recovering from any single node failure requires a node-redundant network, but recovering from link failures requires only link-redundancy with our architecture.

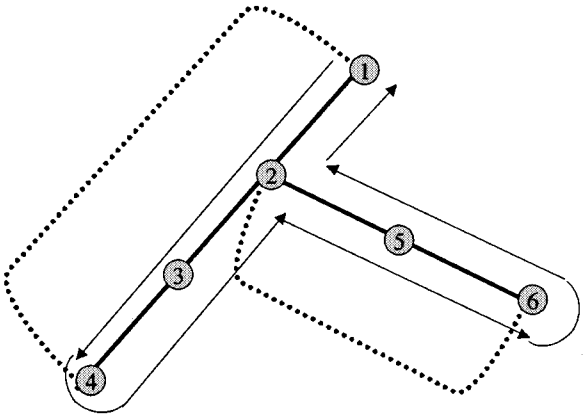

Fig. 3. Example of a DFS tree and the corresponding collection route.

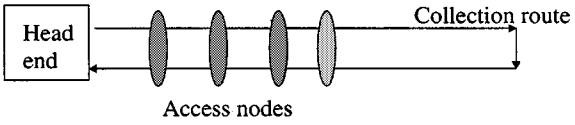

Fig. 4. Collection route for an access network with a single head end and several nodes on the collection route.

The collection portion of the route is constructed as follows. Select a root vertex (any choice suffices) and build a depth-first search (DFS) numbering beginning at the root vertex. The collection route is a walk that traverses nodes as they are considered by the DFS numbering algorithm. Fig. 3 illustrates this process: thick, solid lines represent edges (arc pairs) in the DFS tree, and thick, dotted lines indicate edges not included in the DFS tree. Vertex 1 is selected as the root, from which point vertices 2, 3, and 4 are explored in order. Vertex 4 is a leaf of the DFS tree. The DFS algorithm returns to vertex 3 and subsequently to vertex 2, from which it explores vertices 5 and 6. Vertex 6 is also leaf of the DFS tree. The DFS returns to vertex 5, then 2, and, finally, to vertex 1. The collection route is thus (1, 2, 3, 4, 3, 2, 5, 6, 5, 2, 1), and appears in the figure with thin lines.

The collection route defines a walk that traverses every vertex in G at least once, as shown in Fig. 3. This walk traverses each arc in G at most once; as arcs correspond to fibers in the network, a single wavelength is adequate to support the entire collection route. The collection route is similar to a folded bus, with a single vertex (the root) serving as the head end and all other nodes acting as access nodes, as shown in Fig. 4. The signal collected on the collection portion of the route is distributed on the distribution portion through the head end of the collection route, which is also the root of the distribution route.

The distribution route consists of a directed spanning trees rooted at the DFS tree root. We call this tree the primary tree. Robustness is afforded in the distribution route by constructing a secondary tree that shares the root, but no arcs with the primary tree. The trees are chosen to ensure that removal of any edge (and its two associated arcs) leaves the root connected to every vertex on at least one of the trees. The root then broadcasts the collected traffic on the two trees simultaneously, and any vertex affected by a failure on the primary tree need merely listen to the secondary tree. Methods for constructing such trees are given in

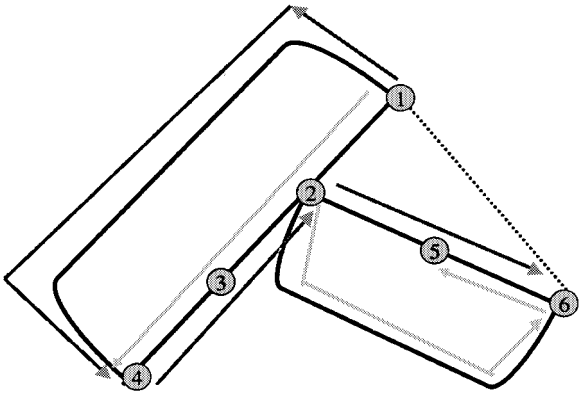

Fig. 5. Example of redundant distribution trees on the network of Fig. 3.

[52]. Fig. 5 shows a pair of redundant trees, depicted in grey lines, defined on the network of Fig. 3.

A single wavelength, used in both directions, is sufficient to construct the primary and secondary trees. The root of the primary and secondary tree must therefore be able to perform wavelength conversion, by placing the traffic from the collection route onto the primary tree. We only require one node to perform wavelength conversion. Alternatively, the two trees may be placed on two separate fibers. Thus, our system may be implemented as two-fiber system, with collection and distribution routes sharing fibers, or as a four-fiber systems, where each fiber carries either one direction of the collection route, or one direction of the distribution trees.

B. Link-Failure Robustness

We may now address how we ensure robustness against link failures. The crux of our algorithm lies in our method of performing link recovery in the collection portion of the route. The recovery is done in the following way. Suppose that link $[i, j]$ fails. If link $[i, j]$ is not included in the DFS tree, its failure leaves the collection route unaffected. We therefore need only consider the case where link $[i, j]$ is included in the DFS tree. We assume wlog that vertex i is the ancestor of j. Failure of $[i, j]$ disconnects j and all of its descendants from the rest of the tree. From the DFS construction and the fact that that we have a two-edge connected graph, j or some descendant of j must have an edge connecting it to some ancestor of j (sibling links cannot exist in a DFS tree). Let k be the descendant of j (possibly j itself) with the lowest number in the DFS numbering such that there is an edge connecting k to some ancestor, say l, of j. Then, the edge $[l, k]$ by construction is not part of the DFS tree. Fig. 6 shows the construction on which our argument is based.

To effect recovery, a new collection route is constructed based on the original collection route. The original collection route included

$$(l_0, l, l_1, \ldots, i_0, i, j, j_1 \ldots, k_0, k, k_1, k_2, \ldots, k_2, k_1, k, k_0, \ldots, j, i, \ldots, l_1, l, l_0, \ldots).$$

Note that any or all of $l_0, l_1, i_0, j_1, k_0, k_1, k_2$ may not exist. The new route is

$$(l_0, l, k, k_0, \ldots, j, \ldots, j, \ldots, k_0, k, k_1, k_2, \ldots, k_2, k_1, k, l, l_1, \ldots, i_0, i, i_0, \ldots, l_1, l, l_0, \ldots).$$

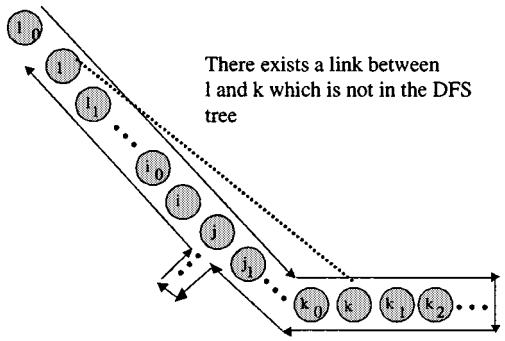

Fig. 6. Collection route before link failure.

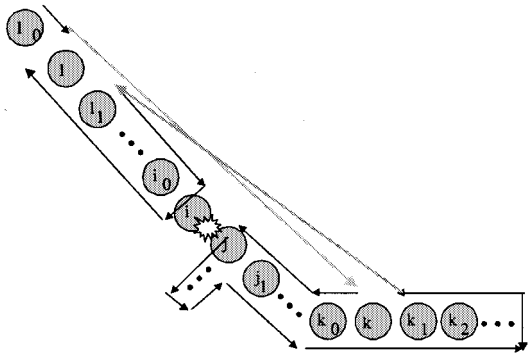

Fig. 7. Collection route after failure of link $[i, j]$.

The new route is shown in Fig. 7, with the portion of the routes that do not use links used by the DFS tree shown in gray lines (the two gray lines in our illustration are for traversing the link $[k, l]$ in both directions). For the route after failure of $[i, j]$, we keep the old route except for the following changes. When we first encounter l, we immediately proceed to k, from where we explore all the descendants of j in the DFS numbering. The exploration of the nodes that are descendants of j can be thought as being done in two parts. First, we explore the nodes that are not descendants of k in the DFS numbering. Next, we explore the nodes that are descendants of k in the DFS numbering. Then, we return to l, from which we explore the nodes to i in the DFS order. At i, we immediately backtrack to l. After we visit l for the third time, we resume exploring nodes with the original route.

We may give an interpretation of the above route in terms of the switching that needs to be done at nodes. For the distribution portion, each vertex that is downstream of the link failure in the primary tree switches to receiving on the secondary tree. On the collection portion, vertex l connects (l_0, l) to (l, k), (k, l) to (l, l_1) and (l_1, l) to (l, l_0). Node k connects (l, k) to (k, k_0), (k_0, k) to (k, k_1) and (k_1, k) to (k, l). Note that branchings may occur at k, l, or j. In that case the above connections must be amended so that all those branchings are explored. Thus, for instance, the first connection into a vertex would be followed by connections ensuring explorations of those branchings. Then, the connections would resume as above.

C. Node-Failure Robustness

Extending the route construction techniques to allow recovery from node failures requires some modifications to

the collection and distribution portions of the routing. The modification to the distribution is necessary to handle the failure of the root node of the two distribution trees. Assume that the graph G is two-vertex redundant (otherwise some node failure is unrecoverable). The collection route construction is identical. The major difference lies in the fact that we cannot rely on a single node to connect the collection portion of the route to the distribution portion as for link failures, since that single node may itself experience a failure. We first examine recovery in the collection portion and next we consider the distribution portion.

The collection portion visits every node at least twice. Let vertex n_1 be the root of the DFS. With a two-vertex connected graph, the resulting DFS tree contains only a single arc originating at n_1. In particular, the DFS root has only a single child, as all other nodes must be reachable from that child without passing through the root and are thus found to be descendants of the child. Denote the root's unique child n_2. The last arc of the collection route is then (n_2, n_1).

In case of failure of a vertex other than n_1, we perform recovery in a manner akin to that given for link failures, with some crucial modifications. Let j be the failed vertex and i the vertex preceding it in the DFS tree. Let us consider all branchings of the DFS tree that split off at j. If no branchings split off at j, we consider that there is a single branching. For ease of exposition, we refer to branchings from j, which include the branchings that split off at j and the single branching when there are no splits at j. Since G is two-vertex connected, there is an edge connecting at least one vertex in each of these branchings from j to a vertex upstream of j in the DFS tree. For a particular branching, say branching x, from j, let us call k^x the vertex that is connected to a vertex, l^x, upstream of j. The arguments carried out for the failure of link $[i, j]$ apply to the case where vertex j fails instead.

For each branching from j, the new collection route is constructed as if link $[i, j]$ had failed. Suppose that there are b branchings from j. Let us suppose at first that all the l^xs are distinct. The branchings are numbered so that, for all x, y between 1 and b, if $x < y$ then l^x has a lower DFS number than l^y. A new collection route is constructed using the original collection route. Fig. 8 illustrates the original routes, with only two branchings shown from j. The original collection route included, without loss of generality

$$
\begin{aligned}
&(n_1, n_2, l_0^1, l^1, l_1^1, \ldots, l_0^2, l^2, l_1^2, \ldots,\\
&\ l_0^b, l^b, l_1^b, \ldots, i_0, i, j, j^1 \ldots,\\
&\ k_0^1, k^1, k_1^1, k_2^1, \ldots, k_2^1, k_1^1, k^1, k_0^1, \ldots\\
&\ j_1, j, j^2, \ldots, k_0^2, k^2, k_1^2, k_2^2, \ldots,\\
&\ k_2^2, k_1^2, k^2, k_0^2, \ldots, j^2, j, j^3, \ldots,\\
&\ k_0^b, k^b, k_1^b, k_2^b, \ldots, k_2^b, k_1^b, k^b, k_0^b, \ldots\\
&\ j^b, j, i, i_0, \ldots, l_1, l, l_0, \ldots, l_1^b, l^b, l_0^b, \ldots, n_2, n_1).
\end{aligned}
$$

Note that any or all of i_0 and of $l_0^1, l_1^1, \ldots, l_0^b, l_1^b$, $k_0^1, k_1^1, k_2^1, \ldots k_1^b, k_1^b, k_2^b$ may not exist. Moreover, l_1^x and l^{x+1} may be the same and l^x and l_0^{x+1} may be the same. The new route is

$$
\begin{aligned}
&(n_1, n_2, l_0^1, l^1, k^1, k_0^1, \ldots j^1, \ldots, k_0^1, k^1, k_1^1, k_2^1, \ldots,\\
&\ k^1, l^1, l_1^1, \ldots, l_0^2, l^2, k^2, k_0^2, \ldots\\
&\ j^2, \ldots, k_0^2, k^2, k_1^2, k_2^2, \ldots, k^2, l^2, l_1^2, \ldots\\
&\ l_0^b, l^b, k^b, k_0^b, \ldots, j^b, \ldots,\\
&\ k_0^b, k^b, k_1^b, k_2^b, \ldots, k^b, l^b, l_1^b, \ldots\\
&\ l_1^2, l^2, l_0^2, \ldots, l_1^1, l^1, l_0^1, \ldots, n_2, n_1).
\end{aligned}
$$

Fig. 9 shows the route after failure of node j for the example shown in Fig. 8.

We may now describe how we may modify the above route when the l^xs are not all distinct. Suppose that l^x and l^{x+1} are the same. Then, after having visited the xth branching from j, we would not proceed to l_1^x but instead proceed to k^{x+1} and proceed as before. In effect, we use the same route as before, for the special case where $l_1^x = l_0^{x+1} = l^{x+1} = l^x$. Note that, if n_1 fails, our collection route can be effected by making n_2 the root of the DFS tree.

The distribution route is made by constructing two trees in such a way that the failure of any node other than the root leaves every other node connected to the root by at least one tree. One approach to constructing such trees is a straightforward

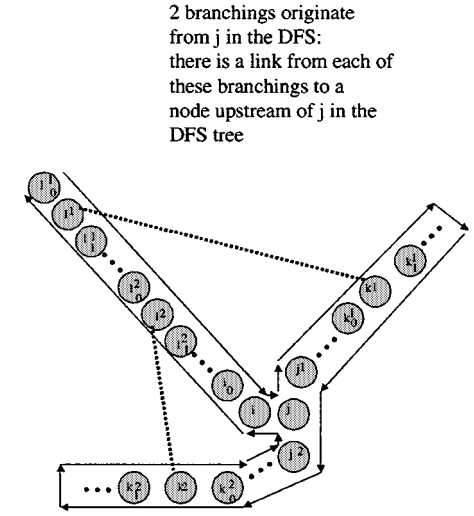

Fig. 8. Collection route before node failure.

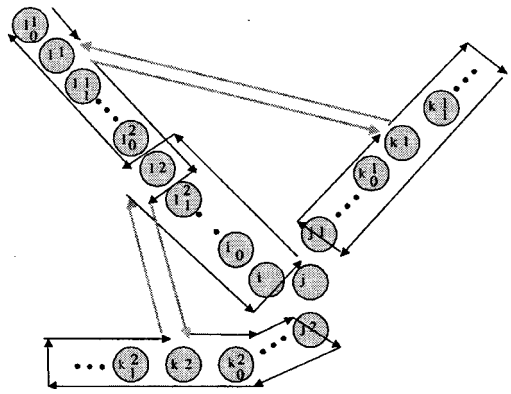

Fig. 9. Collection route after failure of node j.

extension of the algorithm given in [52]. From the algorithm in [52], we can establish that it is possible to include arc (n_1, n_2) in the secondary tree. Indeed, the algorithm first chooses an arbitrary undirected cycle including vertex n_1. From Menger's theorem, such a cycle can be a cycle including edge $[n_1, n_2]$. Moreover, we can arbitrarily choose a direction on that cycle to generate the first portion of the primary tree. If we choose the direction that traverses arc (n_2, n_1), arc (n_1, n_2) is included in the secondary tree. New nodes are explored by searching nodes that are adjacent to nodes already included in the primary and secondary trees. This exploration is effected in the following way: we create a directed path beginning at a covered vertex and ending at another covered vertex and such that all intermediate nodes are uncovered. Using the numberings given to the nodes, the path is traversed (except for the last vertex in the path) in one direction for the primary tree and in the reverse direction for the secondary tree. The root vertex, in this case n_1, is always the starting point of a path inclusion in the primary tree. For all nodes adjacent to n_1, we choose to explore them from n_1. Thus, except for (n_1, n_2), no other arc originating at n_1 is included in the secondary tree.

If a vertex n other than n_1 fails, each vertex downstream of n in the primary distribution tree switches to receiving on the backup tree. Other nodes are unaffected by the failure. The difficulty arises when vertex n_1 fails. In this case, n_1 no longer inserts packets into the collection route, and n_2 has all collected packets at the end of the collection route. We can, thus, truncate the collection portion of the route at n_2 and make n_2 the root of the backup tree. Node n_2 then broadcasts on the secondary tree of the distribution route. As all nodes are downstream of n_1 in the distribution portion of the route, all nodes switch to receiving on the backup tree. Thus, n_2 acts as the root on the backup tree. Although link failure recovery required only a single wavelength changer at n_1, dealing with node failures requires dealing with the possible loss of the wavelength changer, which must thus be replicated at n_2.

IV. Access Protocol

In this section, we overview the access protocol and discuss its bandwidth efficiency and fairness properties. Our scheme consists of a single head end and of access nodes. The head end issues permits to all access nodes on the network. The nodes share a single wavelength and transmit only when they receive, from the head node, the authorization to transmit. The data is collected in the following way: there exists a route, starting at the head end, that traverses all the nodes in a given order once and then traverses the nodes again, but in the reverse order, and terminates at the head end after having collected all the data in one round. The combination of these two traversals of the nodes, during which data from those nodes is collected, we refer to as the collection route. Fig. 4 shows a schematic of the setup we consider.

Bandwidth efficiency and fairness are of concern in access networks. In particular, while packetized access is desirable from the point of view of flexibility and compatibility with standard protocols such as transmission control protocol/Internet protocol (TCP/IP), it may be detrimental to efficient use of bandwidth. Moreover, while path protection and link restoration require the use of excess bandwidth beyond that used for primary communications, we want to be parsimonious in the use of bandwidth devoted to protection and recovery. In this section, we address several issues relating to bandwidth efficiency and fairness. First, we address the issue of wavelength allocation. Our scheme requires at most two wavelengths (bidirectional) over the whole network. These two wavelengths may be carried on different pairs of fibers (for a four-fiber system) or may be carried over a single pair of fibers (for a two-fiber system). Through judicious selection of fibers, we may not need to use two wavelengths (bidirectional) over all links.

The second issue we discuss is that of fair provisioning through a simple reservation scheme. Our reservation scheme relies on the fact that the head node, in an unpruned scenario, is both the originating and the terminal point of the collection portion of the route. The fact that each node sees the traffic at least twice is an important fact in the treatment of our last issue, that of the efficient use of capacity. The efficient use of capacity in our scheme stems from making use of unreserved bandwidth, we propose to make use of both unreserved bandwidth and of *reserved bandwidth that was not used*. We describe a way for achieving utilization of unreserved and unused reserved bandwidth and discuss some means of insuring some measure of fairness. The reuse of unused slots is a feature of other folded bus schemes such as DQDB and CRMA [66] and has been discussed in the context of optical access in [35].

We propose a new protocol to achieve efficient use of bandwidth. The main advantages of our access protocol are:

- reservations are allowed but not necessary
- variable length packets are allowed
- the protocol can rapidly respond to new traffic demands
- both unreserved bandwidth and reserved unused bandwidth can be utilized by users in close to real time.

The protocol is similar to a folded bus scheme, with certain crucial modifications. On the collection portion of the route, each node sees traffic on the collection route in both directions along any link. A node, say i, places requests for reservations on an out-of-band request channel. The request channel is accessed in a time-slotted manner, to ensure that every node can transmit its requests. Note that the timing requirements on the request channel are fairly loose. The head end node processes the requests and, with some delay that depends on the particular implementation of bandwidth assignment strategies at the head end node, assigns bandwidth. The bandwidth assignment is made by transmitting "begin send" (BS) and "end send" (ES) signals on the wavelength that is accessed for the collection route. These signals are addressed to specific nodes, so the message BSi would indicate that node i can begin sending. The time between a BSi and a ESi is called the transmission interval for node i. The time between the transmission of a BSi by the head end node and the reception of a BSi by the head end node is called a transmission cycle. When node i sees BSi, it starts transmitting traffic. Node i transmits until it sees the message ESi or until it has no more traffic to send. If node i ceases transmission because it has no more to transmit, node i places a end-of-transmission (EOT) signal on the access wavelength.

After generating an EOT signal, node i does not transmit until the next ESi signal. For the efficient operation of our protocol, it is important that node i transmit only as long as it has something to transmit, otherwise idle time in a transmission interval of node i cannot be reused, as will become apparent in the sequel.

Efficient use of bandwidth is achieved in the following manner. First, node i can use unreserved bandwidth if node i has traffic that was not accommodated in its last transmission interval. If an ES signal has been seen and no BS signal has been seen, and if node i has been given the appropriate authorizations by the head end node, node i immediately transmits a BTi (begin transmission) signal and commences transmission after a delay τ_i. The delay is given by the head end node. If another BT signal is seen before i commences transmission, i desists until a ET (end transmission) signal is received. Otherwise, node i transmits and, upon completion of its transmission, places a ETi signal on the access wavelength. If a BS signal is received by node i, node i ceases transmission. The head end can control the use of the unreserved bandwidth in different ways.

- By specifying in which intervals node i can transmit. For instance, the head end node may constrain i to be able to use unreserved bandwidth only after ESj.
- By specifying when node i can access unreserved transmissions. For instance, node i may be allowed to transmit only when it sees an ES signal for the second time in a transmission cycle, or when it sees it for the first time.
- By specifying τ_i. A node with a short τ_i may be able to preempt transmission of nodes downstream from it in the collection route.

The second aspect of our access protocol's efficient use of bandwidth is the use of reserved unused bandwidth. Node i, with proper authorization by the head end node, can transmit after receiving an EOT signal. The description of the access is the same as for the use of unreserved bandwidth with the difference that the ES signal is replaced by an EOT signal and the BS signal is replaced by an ES signal. The delay τ_i replaced by a possibly different delay, which we denote by θ_i. In a manner akin to the control of the use of the unreserved bandwidth by the head end, the unused unreserved bandwidth can be controlled by controlling on which unused transmission intervals node i can transmit and the parameter θ_i. Unlike unreserved bandwidth, however, node i can only transmit in the transmission interval of node j the second time that is sees that transmission interval in a transmission cycle, unless node j has already had access to that transmission cycle (because node j is upstream of node i in the collection route).

V. Implementation Issues

This section illustrates the feasibility of implementing our recovery protocol in hardware by outlining a simple yet flexible implementation. In particular, we use the numbering and spanning tree defined by the DFS to implement the protocol through actions and timing local to the network nodes. The discussion focuses on the collection portion of the route rather than the distribution portion. As the redundant distribution trees are a form of active traffic replication, only the recipients affected by a failure need switch over to the second tree. Such switching is a purely local action. Also, with the exception of the redundancy of the root node, the operation of these trees is described in detail in earlier work [52].

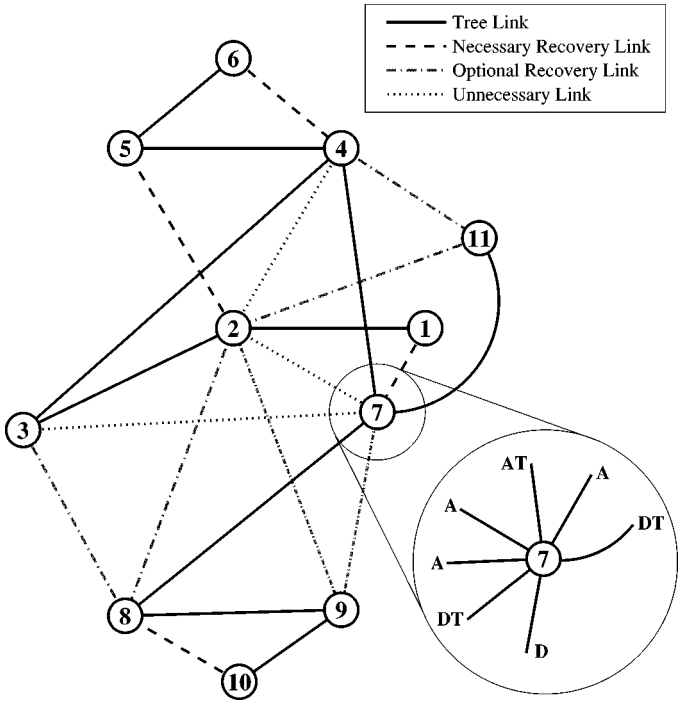

Fig. 10. Example DFS tree and labeling. The graph shown is the NJ LATA network. An expanded version of node 7 is labeled with ancestor (A), descendant (D), and tree (T) markings. Link styles indicate their relevance to the collection route and recovery.

Recovery of the collection route can be accomplished through actions local to each node in a network. For the purposes of the implementation, we assume that each node contains an optical switch fabric capable of supporting lightpath routing between the node's links. To implement the recovery protocol, we add a single configurable device to each link in the subnetwork to be used by the access architecture. After configuration, these devices act independently to implement recovery on the collection route.

A. Strategy and Timing

We begin by labeling links in terms of the numbering defined by the DFS. Each link from a node is labeled as either an ancestor (A) or a descendant (D) link depending on the relative order of the node at the other end of the link. In addition, links in the spanning tree produced by the DFS are labeled as tree (T) links. As an example, consider the DFS exploration of the NJ LATA network shown in Fig. 10. In the implementation, these labels are used to configure the link devices.

Now consider the case of a link failure. A failed link divides the DFS tree into two parts, which we term the upper and lower sections. The upper section contains the root of the tree. At the time of the failure, the nodes at either end of the link detect the absence of the pilot tone and loop back from the failed link. For the upper part of the tree, the resulting flow of traffic is equivalent to a collection routing on a subset of the network. For

the lower part of the tree, the flow becomes cyclic, preserving the majority of the traffic in the fibers until restoration completes and exerting "back pressure" through collision avoidance as necessary.

As shown in the Section III, at least one link outside the tree crosses between the upper and lower parts of the tree. The protocol must select exactly one of these links through which to effect recovery, and must do so in a distributed fashion. Only links to ancestor nodes in the upper part represent viable alternatives for this selection process. As the upper part of the tree can continue to collect traffic without modification, we choose to initiate recovery from the lower part. Detection of the link failure in the lower part occurs first at the node at the end of the failed link and propagates along the collection route using a failure signal similar to that used for pilot tones (either in-band or subcarrier multiplexed).

As detection of a failure propagates from node to node, each node must decide whether or not it can effect recovery. In order to prevent nodes from attempting to recover through ancestors in the lower part of the tree, nodes are required to suppress such recovery when they detect a failure. This suppression is accomplished by asserting a suppression signal over all nontree descendant links. Due to the triangle inequality, these suppression signals typically arrive at a node before the node detects a failure. However, such may not always be the case, and a tunable electronic delay element is necessary to guarantee proper suppression.

Consider the propagation of the failure along the collection route. A simple timing analysis demonstrates that failures must not be propagated downstream until a node has decided that it cannot realize recovery itself. Consider a series of nodes below a link (or node) failure. Before deciding to splice in a nontree ancestor link, a node must wait to guarantee that suppression of such links has been asserted and must also wait to ensure that no node upstream has already recovered. As the recovery decision process requires at least some input from the node immediately upstream, part of this delay cannot be overlapped with the delay at that node. This component of the delay thus accumulates along the path below a cut, requiring that nodes delay based on the global structure of the collection route rather than on purely local constraints.

In contrast, if failures are not detected until all upstream nodes have attempted recovery, a node must wait only for suppression, the time for which depends only on the propagation delays on the nodes' links. Such a scheme can be realized by requiring a node to mask or hold failures until deciding that it cannot recover. As this masking serializes the parallelizable component of the delays at each node, however, global recovery times are longer.

B. Implementation Approach

We are now ready to discuss the implementation. As mentioned earlier, we use a single, configurable device on each link to implement the protocol. Based on the labeling of the associated link and on the current status of the network, the device determines whether or not the link is spliced into the collection route and generates necessary signals.

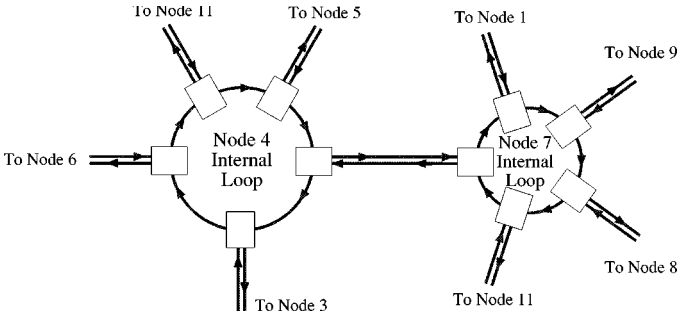

Fig. 11. Example internal loops. The loops shown correspond to those of nodes 4 and 7 from the example of Fig. 10. Links unnecessary to recovery are excluded from the internal loops.

The collection route is formed by using the switch fabric to loop a fiber through all link devices in a node. Fig. 11 shows an example based on the DFS labeling of the NJ LATA network from Fig. 10. The internal loop connects the access wavelength in a single cycle beginning with the tree ancestor, passing through all other ancestor links, then through nontree descendant links, and finally looping through links to descendants in the collection tree before returning to the tree ancestor link. For node 7 in the NJ LATA network, the tree ancestor is node 4, nontree ancestor is node 1 (the links to nodes 2 and 3 are not necessary for recovery, as mentioned in Fig. 10, nontree descendants is node 9, and tree descendants are node 8 and 11. The ordering supports the operation of the protocol: failures detected upstream or immediately above the node are detected at the device attached to the tree ancestor link. After a delay to ensure the arrival of suppression signals, failure signals pass to each nontree ancestor link and propagate only if the link has been suppressed. If no ancestor link is available, the failure marker passes through the nontree descendant links to initiate suppression of descendants. Finally, the failure is passed to the descendants in the tree.

In the example, we have chosen to minimize the number of links used for restoration. This process involves reasoning about link and node failure coverage provided by each link outside the tree. Leaf nodes, for example, require at least one nontree ancestor link to be included for recovery from failure of the link to the tree ancestor (or of the ancestor node itself). However, one can often select an ancestor link that also provides coverage for other link or node failures. In Fig. 10, links in the tree are represented as solid lines and links necessary to recovery are represented as dashed lines. Additional links are necessary for complete failure coverage, but only three of six must be chosen for the graph shown; these links are represented as dash-dotted lines. Finally, several links are unnecessary; these appear as dotted lines in the figure. The fibers not required for recovery can be used to support additional lightpath traffic. Links not used for restoration are simply left out of the internal loop, as shown in Fig. 11.

A model of the link device appears in Fig. 12. The device attaches optically to both the external fiber and to the optical switch fabric. The internal loop structure is then implemented through configuration of the switch fabric, using the same mechanisms as are necessary for lightpath routing. A pilot tone P is used to detect link and node failure, and each link device both

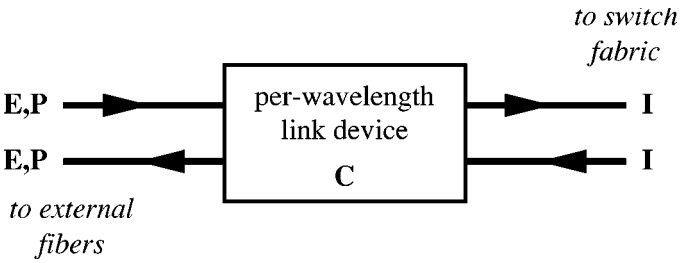

Fig. 12. Model of link device implementing collection route recovery.

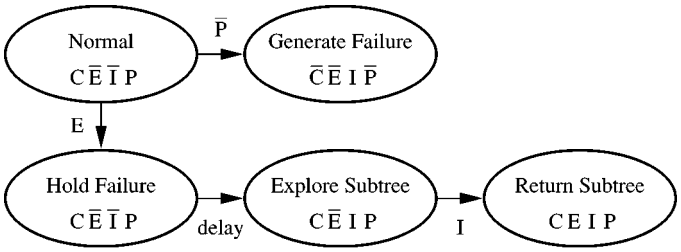

Fig. 13. State diagram for an AT link device.

generates and detects the presence of this pilot tone. Link devices can also transmit a single failure signal both externally (denoted E) and within a node's internal loop (denoted I). Depending on the device configuration, the E signal can represent failure propagation (AT or DT), recovery suppression (A), or a recovery attempt (D). As with the pilot tone, the failure signal can be implemented in-band or out-of-band; the implementation is orthogonal to our discussion. In addition to the three bits of signal generation state (EIP), a link device contains a fourth bit of state corresponding to whether or not the external link is connected to the node's internal loop or not. Such a connection can be implemented with a 2×2 optical switch, which provides loopback in the case of link failure.

After the collection route is selected and the internal loops are configured within the switch fabrics, each link device is provided with ancestor and tree bits (A and T) and placed in the "Normal" state for its type. Until a failure occurs in the network, all link devices remain in their initial states; only when a failure signal arrives or a pilot tone is lost do the link devices change state to restore the collection route. In the four initial states, the link devices generate only the pilot tone P; the connection C between the internal loop and the external link is made for tree links and not made for nontree links.

In the remainder of this section, we discuss the state machine for each configuration of the link devices. Figs. 13–16 illustrate the designs. Each oval in a figure represents a single state for the link device and provides a meaningful name as well as signal generation (EIP) and connection (C) information. A horizontal bar over a variable indicates that the signal is not generated or the link is not connected into the internal loop. The arrows in the figures represent possible transitions; each transition is labeled with the change of input that causes that transition. If no arc exists for a given change of input from a particular state, that change should not occur in the state, and such an event should be treated as a system error and handled by a higher layer of network management.

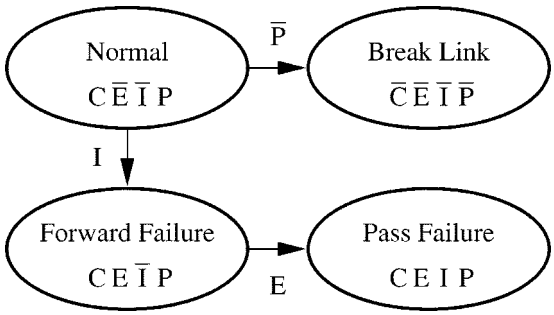

Fig. 14. State diagram for a DT link device.

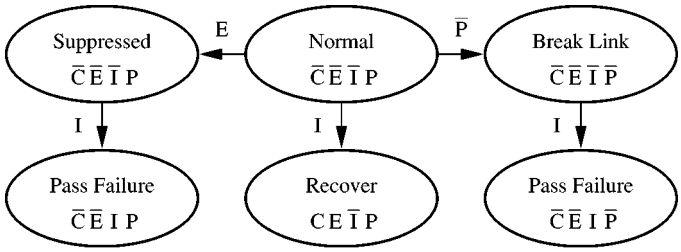

Fig. 15. State diagram for an A link device.

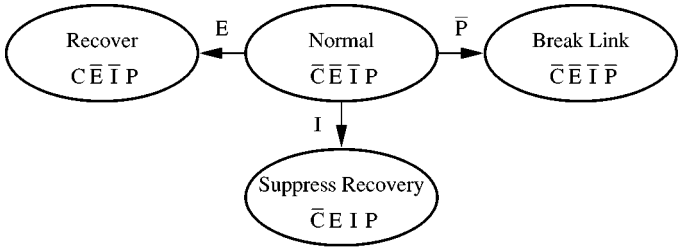

Fig. 16. State diagram for a D link device.

C. Tree Ancestor State Machine

Each node contains a single link device of type AT connecting the node to its ancestor in the DFS tree. This link device serves to introduce a failure signal into the node's internal loop, and does so for one of two purposes: failure generation and subtree recovery. Recall that when a link or node in the collection route fails, only nodes disconnected from the root need act to recover. Such failures are always detected by the AT device of the node just below the failure, and it alone is responsible for generating a failure signal in response to a lost pilot tone. The state machine for an AT device appears in Fig. 13. As shown in the figure, the device immediately removes the link from the internal loop (i.e., loops back by turning off C) and stops generating a pilot tone to address the possibility of partial link failures. The device also emits a failure signal on the internal loop (an I signal), starting the process of recovery.

The AT device in a node also introduces an I signal in order to initiate exploration of the subtree rooted at the node for recovery purposes. It does so in response to a failure signal received from the node's parent (an E signal). The device first holds the failure for a period of time before passing the failure into the internal loop. The delay ensures that suppression signals from ancestor nodes have reached the node, and is based on the relative propagation delays on the node's tree and nontree links. If the failure

signal passes through all other link devices, neither the AT device's node nor any of the node's children was able to recover, and the device returns the failure to the node's parent by generating an E signal.

The normal state lacks a transition based on detection of an I signal. The rationale for this elision is the fact that only the AT device in a node can introduce such a signal into the node's internal loop. Only after the device generates such a signal, in either the generate failure or the subtree explore state, can the signal propagate around the internal loop, passing through all other link devices, and return to the AT device. If the signal returns in the generate failure state, recovery was impossible and must be handled by a higher layer. Such an event can occur only as a result of multiple failures in the network and is thus beyond the scope of this paper.

D. Tree Descendant State Machine

The state machine for a link device of type DT, connecting a node to one of its descendants in the DFS tree, is the simplest of the four types, as it plays only a minor role in recovery of the collection route. A diagram of the state machine appears in Fig. 14. As with the AT device, the DT device responds to the loss of a pilot tone by turning off its own pilot tone and looping back from the link so as to remove it from the internal loop. Unlike the AT device, the DT device takes no action to initiate recovery.

The DT device can also receive an I signal, which it forward across the fiber to the descendant node (as an E signal). If recovery is impossible from the descendant node's subtree, the E signal returns, and the DT device passes the I signal to the next device in its node's internal loop.

The normal state for the DT device lacks a transition based on detection of an E signal. As with the case of the I signal for the AT device, a DT device must send an E signal before it can receive one.

E. Non-Tree Ancestor State Machine

Non-tree ancestor links effect recovery of the collection route. The state diagram for a type A link device appears in Fig. 15. When the device notices an I signal, it splices the link into its internal loop and passes a failure signal (an E signal) to the ancestor node. The ancestor node interprets this signal as a recovery request and splices the link into its internal loop, completing the recovery of the collection route.

An ancestor node below a failure in the DFS tree suppresses recovery attempts by the corresponding A device in its (nonimmediate) descendant node by sending an E signal before the device receives an I signal. This ordering is enforced by the Failure Hold delay in the AT device of the descendant node. Recovery is also suppressed when an A device's link is broken, which may occur in the case of a node failure. In either suppressed state, an I signal is passed to the next device in the node's internal loop.

F. Non-Tree Descendant State Machine

Fig. 16 shows the state diagram for a type D link device. A nontree descendant link can receive a recovery request from its descendant node in the form of an E signal, in response to which it connects the link to its internal loop, completing recovery of the collection route.

A D link device that first receives an I signal must suppress such a recovery attempt, as the D link device's node lies below the failure in the DFS tree. Suppression also takes the form of an E signal, and the I signal is passed to the next link device in the internal loop. Loss of the pilot tone implies that the device's node is above the failure in the DFS tree and that the device will hear no more about it.

VI. Conclusion

We have described a network management architecture for mesh optical local and metropolitan access networks that allows for efficient and fair sharing of an access wavelength and that is robust to link or node failures. Our scheme uses very simple optics with respect to the type of optics required to process headers and buffer packets at very high speeds. In itself, the elimination of buffering enhances reliability insofar at it precludes the loss of packets due to buffer overflow.

Many different directions for future research stem from our architecture. One of these directions is the establishment of effective and simple access policies for ensuring flexibility in the tradeoff between fairness and efficient bandwidth use in the collection route. Another venue of research is the combined choice of collection route and access policies.

References

[1] C. S. Baw, R. D. Chamberlain, and M. A. Franklin, "Fair scheduling in an optical interconnection network," in *Proc. 7th Int. Symp. Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, 1999, pp. 56–65.

[2] R. A. Barry, V. W. S. Chan, K. L. Hall, E. S. Kintzer, J. D. Moores, K. A. Rauschenbach, E. A. Swanson, L. E. Adams, C. R. Doerr, S. G. Finn, H. A. Haus, E. P. Ippen, W. S. Wong, and M. Haner, "All-optical network consortium-ultrafast TDM networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 999–1013, June 1996.

[3] A. Bianco, V. Distefano, A. Fumagalli, E. Leonardi, and F. Neri, "*A-posteriori* access strategies in all-optical slotted WDM rings," in *Proc. Global Telecommunications Conf.*.

[4] C. Bisdikian, "Waiting time analysis in a single buffer DQDB (802.6) network," in *Proc. IEEE INFOCOM*, 1990, pp. 610–616.

[5] M. S. Borella and B. Mukherjee, "A reservation-based multicasting protocol for WDM local lightwave networks," in *Proc. IEEE Int. Conf. Communications*, 1995, pp. 1277–1281.

[6] K. Bogineni, K. M. Sivalingam, and P. W. Dowd, "Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks," *J. Select. Areas Commun.*, vol. 11, pp. 509–604, May 1993.

[7] M.-S. Chen, N. R. Dono, and R. Ramaswami, "A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks," *IEEE J. Select. Areas Commun.*, vol. 8, pp. 1048–1057, Aug. 1990.

[8] I. Chlamtac and A. Ganz, "Design alternatices of asynchronous WDM star networks," in *Proc. IEEE Int. Conf. Communications*, 1989, pp. 23.4.1–23.4.5.

[9] D. Callahan and G. Grimes, "An intelligent hub protocol for local area lightwave networks," in *Proc. Conf. Local Computer Networks*, 1999, pp. 260–261.

[10] M. Conti, E. Gregori, and L. Lenzini, "DQDB under heavy load: Performance evaluation and fairness analysis," in *Proc. IEEE INFOCOM*, 1990, pp. 133–145.

[11] ——, "A methodological approach to an extensive analysis of DQDB performance and fairness," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 76–87, Jan. 1991.

[12] I. Cidon and Y. Ofek, "Metaring—A full duplex ring with fairness and spatial reuse," in *Proc. IEEE INFOCOM*, 1990, pp. 969–981.

[13] R. Chipalkatti, Z. Zhang, and A. S. Acampora, "Protocols for optical star-coupler network using WDM: Performance and complexity study," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 579–589, May 1993.

[14] E. H. Dinnan and M. Gagnaire, "An efficient media access protocol for packet switched WDM photonic networks," in *Proc. Symp. Performance Evaluation of Computer and Telecommunication Systems*, July 1999.

[15] P. W. Dowd, "Random access protocols for high-speed interprocessor communication based on an optical passive star topology," *J. Lightwave Technol.*, vol. 9, pp. 799–808, June 1991.

[16] S. G. Finn, "Hlan—An architecture for optical multi-access networks," in *Dig. LEOS Summer Topical Meetings*, 1995, pp. 45–46.

[17] P. E. Green, L. A. Coldren, K. M. Johnson, J. G. Lewis, C. M. Miller, J. F. Morrision, R. Ramaswami, and E. H. Smith, "All-optical packet-switched metropolitan-area network proposal," *J. Lightwave Technol.*, vol. 11, p. 754, May 1993.

[18] A. Ganz and Y. Gao, "Time-wavelength assignment algorithms for high performance WDM star based systems," *IEEE Trans. Commun.*, pp. 1827–1836, May 1994.

[19] A. Ganz and Z. Koren, "WDM passive star—Protocols and performance analysis," in *Proc. Global Telecommunications Conf.*, 1991, pp. 9A.2.1–9A.2.10.

[20] M. Guizani, "High speed protocol for all optical packet switched metropolitan area networks," *Int. J. Network Manage.*, vol. 8, pp. 9–17, 1997.

[21] M. Hamdi, "ORMA: A high-performance MAC protocol for fiber-optic LANs/MANs," *IEEE Commun. Mag.*, vol. 35, pp. 110–119, Mar. 1997.

[22] E. J. Harder and H.-A. Choi, "Scheduling file transfers in WDM optical networks," in *Proc. 5th Int. Conf. Massively Parallel Processing*, 1998, pp. 186–193.

[23] E. Hall, J. Kravitz, R. Ramaswami, M. Halvorson, S. Tenbrink, and R. Thomsen, "The rainbow-ii gigabit optical network," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 814–823, June 1996.

[24] I. M. I. Habbab, M. Kavehrad, and C. E. W. Sundberg, *Protocols for Very High Speed Optical Fiber Local Area Networks Using a Passive Star Topology*, Dec. 1987, vol. LT-5, pp. 1782–1794.

[25] E. Y. Huang and L. F. Merakos, "On the access fairness of the DQDB MAN protocol," in *Proc. IPCC*, 1990, pp. 325–329.

[26] P. A. Humblet, R. Ramaswami, and K. N. Sivarajan, "An efficient communication protocol for high-speed packet-switched multichannel networks," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 568–578, May 1993.

[27] E. Y. Huang, "Analysis of cyclic reservation multiple access protocol," in *Proc. 19th Conf. Local Computer Networks*, vol. 2, 1994, pp. 102–107.

[28] *Distributed Queue Dual Bus (DQDB)—Subnetwork of a Metropolitan are Network (MAN)*, IEEE Standard 820.6.

[29] A. Itai and M. Rodeh, "The multi-tree approach to reliability in distributed networks,", Rep. 79, 1988.

[30] T. S. Jones and A. Louri, "Media access protocols for a scalable optical interconnection network,", May 1998.

[31] H. B. Jeon and C. K. Un, "Contention based reservation protocols in multiwavelength protocols in multiwavelength protocols with passive star topology," in *Proc. IEEE Int. Conf. Communications*, June 1992.

[32] A. E. Kamal, "Efficient multi-segment message transmission with slot reuse on DQDB," in *Proc. IEEE INFOCOM*, 1991, pp. 869–878.

[33] B. Kannan, S. Fotedar, and M. Gerla, "A protocol for WDM star coupler networks," *IEEE Trans. Commun.*, vol. 40, pp. 730–737, Apr. 1992.

[34] S. Kumar and A. P. Jayasumana, "Request based channel access protocol on folded bus topology," in *Proc. 20th Conf. Local Computer Networks*, 1995, pp. 174–183.

[35] A. C. Kam, K. Y. Siu, R. A. Barry, and E. A. Swanson, "A cell switching WDM broadcast LAN with bandwidth guarantee and fair access," *J. Lightwave Technol.*, vol. 16, pp. 2265–2280, Dec. 1998.

[36] D. A. Levine and I. F. Akyildiz, "PROTON: A media access control protocol for optical networks with star topology," in *Proc. 20th Annu. Computer Science Conf.*, vol. 3, Apr. 1995, pp. 158–168.

[37] R. O. LaMaire, "FDDI performance at 1 Gbit/s," in *Proc. IEEE Int. Conf. Communications*, 1991, pp. 174–183.

[38] T. V. Lakshman, A. Bagchi, and K. Rastani, "A graph-coloring scheme for scheduling cell transmissions and its photonic implementation,".

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

[40] J. Limb and C. Flores, "Description of Fasnet—A unidirectional local area communication network," *Bell Syst. Tech. J.*, vol. 61, pp. 1413–1440, Sept. 1982.

[41] A. Louri and R. Gupta, "Hierarchical optical interconnection network HORN: Scalable interconnection network for multiprocessors and multicomputers,", Jan. 1997.

[42] B. Li, A. Ganz, and C. M. Krishna, "A novel transmission scheme for single hop lightwave networks," in *Global Telecommunications Conf.*, June 1996, pp. 1784–1788.

[43] J. Limb, "A simple multiple access protocol for metropolitan area networks," in *Proc. SIGCOMM*, 1990, pp. 67–79.

[44] J. H. Laarhuis and A. M. J. Koonen, "An efficient medium access control strategy for high-speed WDM multiaccess networks," *J. Lightwave Technol.*, vol. 11, p. 1078, May 1993.

[45] V. P. Lang, E. A. Varvarigos, and D. J. Blumenthal, "The λ-scheduler: A multiwavelength scheduling switch," *J. Lightwave Technol.*, vol. 18, pp. 1049–1063, Aug. 2000.

[46] B. Mukherjee and S. Banerjee, "Incorporating continuation-of-message (COM) information, slot reuse, and fairness in DQDB networks," Division of Computer Science, Univ. California, Davis, CA, Tech. Rep. CSE-90-42.

[47] E. Modiano and R. A. Barry, "Design and analysis of an asynchronous WDM local area network using a master/slave scheduler," in *Proc. INFOCOM '99*, vol. 2, May 1999, pp. 900–907.

[48] M. A. Marsan, A. Bianco, E. Leonardi, A. A. Morabito, and F. Neri, "SR/sup 3/:a bandwidth-reservation MAC protocol for multimedia applications over all-optical WDM multi-rings," in *Proc. INFOCOM '97*, vol. 2, 1997.

[49] M. A. Marsan, A. Bianco, E. Leonardi, F. Neri, and S. Toniolo, "An almost optimal MACprotocol for all-optical WDM multi-rings with tunable transmitters and fixed receivers," in *Proc. IEEE Int. Conf. Communications*, vol. 1, May 1997, pp. 437–442.

[50] M. A. Marsan, A. Bianco, E. Leonardi, A. Morabito, and F. Neri, "All-optical WDM multi-rings with differentiated qos," *IEEE Commun. Mag.*, vol. 37, pp. 58–66, Feb. 1999.

[51] N. Mehravari, "Performance and protocol improvements for very high speed optical fiber local area networks using a passive star topology," *J. Lightwave Technol.*, vol. 8, pp. 520–530, pp. 520–530, Apr. 1990.

[52] M. Médard, S. G. Finn, R. A. Barry, and R. G. Gallager, "Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs," *Trans. Networking*, pp. 641–652, Oct. 1999.

[53] M. Maode, B. Hamidzadeh, and M. Hamdi, "A receiver-oriented message scheduling algorithm for WDM lightwave networks," in *Proc. Global Telecommunications Conf.*, vol. 4, May 1998, pp. 2333–2338.

[54] M. Mishra, E. L. Johnson, and K. L. Sivalingam, "Scheduling in optical WDMnetworks using hidden Markov chain-based traffic predictors," in *IEEE Int. Conf. Networks*, 2000, pp. 380–384.

[55] M. M. Nassehi, "CRMA: An access scheme for high-speed LANs and MANs," in *Proc. IEEE Int. Conf. Communications*, vol. 4, 1990, pp. 1697–1702.

[56] K. Nosu and H. Toba, "An optical multiaccess network with optical collision detection and optical frequency addressing," in *Proc. IEEE Int. Conf. Communications*, vol. 3, Mar. 1990, pp. 968–975.

[57] K. Rauschenbach, S. Finn, R. Barry, K. Hall, J. Moores, and N. Patel, "100-Gbit/s time-division multiplexed multi-access networks," in *Proc. OFC*, 1997, pp. 86–87.

[58] M. A. Rodrigues, "Erasure nodes: Performance improvements for the IEEE 802.6 MAN," in *Proc. IEEE INFOCOM*, 1990, pp. 636–643.

[59] F. E. Ross, "An overview of FDDI: The fiber distributed data interface," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 1043–1051, Sept. 1989.

[60] ——, "Fiber distributed data interface: An overview," in *Proc. 15th Conf. Local Computer Networks*, 1990, pp. 6–11.

[61] T. Strosslin and M. Gagnaire, "A flexible MACprotocol for all-optical WDM metropolitan area networks," in *Proc. IEEE Int. Performance, Computing, and Communications Conf.*, 2000, pp. 567–573.

[62] G. N. M. Sudhakar, N. D. Georganas, and M. Kavehrad, "A multichannel optical star LAN and its application as a broadband switch," , vol. 5, Dec. 1987, pp. 843–847.

[63] S. Selvakennedy and A. K. Ramani, "Analysis of piggybacked token-passing mac protocol with variable buffer size for WDM starcoupled photonic network," in *3rd Int. Conf. High Performance Computing*, 1996, pp. 307–312.

[64] O. Sharon and A. Segall, "A simple scheme for slot reuse without latency in dual bus," *IEEE/ACM Trans. Networking*, vol. 1, pp. 96–104, Feb. 1993.

[65] ——, "On the efficiency of slot reuse in the dual bus configuration," *IEEE/ACM Trans. Networking*, vol. 2, pp. 89–100, June 1994.

[66] ——, "Schemes for slot reuse in CRMA," *IEEE/ACM Trans. Networking*, vol. 2, pp. 269–278, June 1994.

[67] F. A. Tobagi, F. Borgonovo, and L. Fratta, "Expressnet: A high performance integrated-services local area network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 898–913, Nov. 1983.

[68] W. C. Tseng and B. U. Chen, "D-Net: A new scheme for high data rate optical local area network," *IEEE J. Select. Areas Commun.*, vol. SAC-1, pp. 493–499, Apr. 1983.

[69] H. R. van As, "Performance evaluation of bandwidth balancing in the DQDB MAC protocol," in *Proc. EFOC/LAN Conf.*, 1990.
[70] H. R. van As, W. W. Lemppenau, P. Zafiropulo, and E. A. Zurfluh, "CRMA-II: A Gbit/s MAC protocol for ring and bus networks with immediate access capability," in *Proc. EFOC/LAN Conf.*, 1991, pp. 262–272.
[71] E. A. Varvarigos, "The 'packing' and the 'scheduling packet' switch architecture for almost all-optical lossless networks," *J. Lightwave Technol.*, vol. 18, pp. 1049–1063, Aug. 2000.
[72] L. Wang and M. Hamdi, "Efficient protocols for multimedia streams on WDMAnetworks," in *Proc. 12th Int. Conf. Information Networking*, vol. 1, 1998, pp. 241–246.
[73] G. Watson and S. Ooi, "What *should* a Gbits/s network interface look like," in *Protocols for High-Speed Networks*, M. J. Johnson, Ed. Amsterdam, The Netherlands: North-Holland, 1990, vol. II, pp. 237–250.
[74] J. W. Wong, "Throughput of DQDB networks under heavy load," in *Proc. EFOC/LAN Conf.*, 1989, pp. 146–151.
[75] H.-T. Wu, Y. Ofek, and K. Sohraby, "Integration of synchronous and asynchronous traffic on the MetaRing architecture and its analysis," in *Proc. IEEE Int. Conf. Communications*, 1992.
[76] G. Watson, S. Ooi, D. Skellen, and D. Cunningham, "HANGMAN Gbit/s network," *IEEE Network Mag.*, July 1992.
[77] G. C. Watson and S. Tohme, "S++-anew MAC protocol for Gb/s local area networks," *IEEE J. Select. Areas Commun.*, vol. 11, pp. 531–539, May 1993.
[78] M. C. Yuang and M. C. Chen, "A high performance LAN/MAN using a distributed dual mode control protocol," in *Proc. IEEE Int. Conf. Communications*, vol. 1, 1992, pp. 11–15.
[79] A. Yan, A. Ganz, and C. M. Krishna, "A distributed adaptive protocol providing real-time services on WDM-based LANs," in *Proc. Global Telecommunications Conf.*, vol. 14, June 1996, pp. 1245–1254.
[80] A. Zehavi and A. Itai, "Three tree-paths," *J. Graph Theory*, vol. 13, pp. 175–188, 1989.
[81] X. Zhang and C. Qiao, "Pipelined transmission scheduling in all-optical TDM/WDMrings," *Sixth Int. Proceedings*, vol. 37, pp. 144–149, 1997.
[82] ——, "On scheduling all-to-all personalized connection and cost-effective designs in WDMrings," *IEEE/ACM Trans. Networking*, vol. 7, pp. 435–445, 1999.

Muriel Médard (S'91–M'95–SM'02) received the B.S. degrees in electrical engineering and computer science and in mathematics, in 1989, the B.S. degree in humanities, in 1990, and the M.S. and Sc.D. degrees in electrical engineering, in 1991 and 1995, respectively, from the Massachusetts Institute of Technology (MIT), Cambridge.

She is an Assistant Professor in the Electrical Engineering and Computer Science Department at MIT and a member of the Laboratory for Information and Decision Systems. She was previously an Assistant Professor at the Electrical and Computer Engineering Department and a member of the Coordinated Science Laboratory at the University of Illinois, Urbana-Champaign, Urbana, IL. From 1995 to 1998, she was a Staff Member at MIT Lincoln Laboratory, Lexington, MA, in the Optical Communications and the Advanced Networking Groups. Her research interests are in the areas of reliable communications, particularly for optical and wireless networks.

Ms. Medard is the winner of the 2002 IEEE Leon Kirchmayor Prize Paper Award and a NSF Career Award.

Steven S. Lumetta (S'97–M'98) received the A.B. degree in physics, in 1991, the M.S. degree in computer science, in 1994, and the Ph.D. degree in computer science from University of California, Berkeley, in 1998.

He is an Assistant Professor of Electrical and Computer Engineering and a Research Assistant Professor in the Coordinated Science Laboratory at the University of Illinois, Urbana-Champaign, Urbana, IL. He has worked on a wide range of problems in scalable parallel computing, including languages (Split-C), tools (Mantis debugger), algorithms, and runtime systems, culminating in his dissertation on multiprotocol, user-level communication on clusters of SMPs. His research interests are in optical networking, high-performance networking and computing, hierarchical systems, and parallel runtime software.

Liuyang Li received the B.E. degree in electrical engineering from Xi'an Jiaotong University, Xi'an, China, in 1997.

He is an Research Assistant in the Coordinated Science Laboratory and a Ph.D. student in the Computer Science Department at the University of Illinois, Urbana-Champaign, Urbana, IL.