A network management architecture for robust packet routing in optical access networks *

Muriel Médard
medard@mit.edu,
Laboratory for Information and Decision Systems,
Room 35-212,
Massachusetts Institute of Technology,
77 Massachusetts Avenue, Cambridge, MA 02139,

Steven Lumetta
steve@crhc.uiuc.edu,
Coordinated Science Laboratory,
University of Illinois Urbana-Champaign,
1308 Main Street, Urbana, IL 61801.

.

March 2, 2001

Abstract

We describe an architecture for optical local area network (LAN) access networks which allows for efficient and fair sharing of bandwidth and which is robust to link or node failures. Our architecture allows for sharing a single wavelength in a flexible manner. Unlike most WDM LAN, schemes, we do not assume a star topology but allow our topology to be an arbitrary mesh network, as long as that network is link or node redundant. Our architecture allows for reservation of bandwidth and uses a simple head-ends rather than a routers. Unlike a router, the head end does not perform any action on individual packets and, in particular, does not buffer packets. Our architecture thus avoids the difficulties of processing packets in the optical domain, while allowing for packetized shared access of wavelengths.

*This work was supported by DARPA NGI.

1 Introduction.

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

One approach to providing the benefits of the high data rates afforded by optics is simply to extend the traditional electronic and electro-optic approaches and attempt to implement them using optical technologies. In electronics, operations such as buffering, adding packets and dropping packets, or merging packet streams, are done with ease. In optics, buffering is onerous. Operations such as retrieving a packet from a traffic stream affects the whole stream. The architecture we propose seeks to make use of the strengths of optics and avoid the operations which are cumbersome or expensive in optics. Our goal is to do so in a way which is robust and reliable.

Our design rationale is the following:

- **Avoid buffering**: buffering in the network, with the attendant issues of cost and possibility of overflow, is to be avoided. The onus for buffering is pushed back to the users. Thus, a user who sends excessive traffic will have to buffer its traffic, but will not be able to affect other users in the network by creating buffer management problems. Our design uses no buffering in the network.

- **Do not use a switch unless you have to**: this issue is partly related to the previous issue, as we discuss later. The advent of very large routers and switches will enable packet-switched communications in the Tb/s range (aggregate). However, while traffic to remote locations should be handled by such switches, local, enterprise-level traffic need not be handled by these switches. Moreover, there are several reasons while local traffic should avoid these switches. The first is that network managers should be able to control the policies for bandwidth use locally without having access to a large switch. The second is that switches are subject to congestion problems. It is desirable to isolate enterprise networking from the vagaries of congestion and other causes of failure associated with external traffic. This issue is related to the buffering issue mentioned first. Finally, switches are expensive and often extensibility is difficult. Thus, not routing through switches traffic which does not need to be routed through switches avoids unnecessary costs and the disruptions caused by switch upgrades.

- **Use the optical layer to provide recovery**: our architecture is well suited to packet-switched traffic but does not rely on a specific protocol, say IP/TCP, IP/UDP or ATM. Instead, our access scheme runs below such protocols. The cornerstone of our robustness is that recovery is performed at the optical layer, in a manner which is robust, simple and rapid enough that recovery at higher layers does not need to be triggered.

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 which can be used with our network management, different types of protocols can be used in combination with our network management. The main characteristics which distinguish our architecture from previous work are:

- our network management architecture implements, through appropriate routings, a local area network over an arbitrary redundant mesh topology, rather than over a star or multiple rings. In particular, our network may be a logical topology overlaid over some other physical arbitrary redundant topology, such as a portion of a metropolitan area network, as shown in figure ??.

- Our network management architecture enables recovery by preplanned rerouting in the case of a link or node failure. Routing and recovery are closely intertwined, since the routing is selected so as to enable recovery and recovery is effected in part through preplanned rerouting.

- We consider the case where the traffic can be carried over a single wavelength. Currently, per wavelength rates reach 10 Gbps for OC-192 and 40 Gbps per wavelength systems have been demonstrated and are in commercial development. Enterprise routers currently offer throughputs of the order of tens of Gbps. Thus, it is reasonable to assume that a single wavelength can carry the traffic of certain enterprise networks or virtual private networks. Extensions to several wavelengths are discussed in Section 6.

Our paper is organized as follows. In the next section, we overview some of the literature in the area of optical local area networks. In section 3, we describe the main features of our network management architecture: routings and associated recovery mechanisms which provide us with a means of recovering from link or node failures. In section 4, we describe a possible access protocol for use with our network management architecture. Our protocol allows us to share wavelengths in a flexible, bandwidth-efficient and fair manner. In section 5, we discuss implementation issues, presenting a possible implementation and the functionality required at the nodes and at the head end. We present our conclusions and areas for further research in section 6.

2 Background

In this section, we briefly overview previous work in topics that are relevant to the main aspects of our network management architecture. In particular, we consider the following topics: topologies for optical LANs and MANs; folded bus schemes; redundant tree routings; 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 ([MB99, HRS93, LK93, MHH98, NT90, LA95, Gui97, SR96, CG89, GK91, LGK96, YGK96, HKS87, SGK87, Meh90, JU92, GG94, BSD93, CG99, Dow91, MJS00, WH98, SG00, HKR+96, GCJ+93, KFG92, CDR90]). These star architectures usually involve a passive optical broadcast star. These stars generally have senders and/or receivers which 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) ([Ros89, Ros90, LaM91]). Multiple ring topologies may be interconnected through a hub [JL98], or rings may coexist in a logically interconnected fashion over a single physical ring ([MBL+97b, MBL+99, MBL+97a]), or rings may be arranged hierarchically ([BDF+, JL98, LG97]). Our topology considers arbitrary link or node redundant topologies, as will be detailed in the next section. Also, the bulk

Our routing scheme uses in a particular type of folded bus, as well as redundant broadcast trees, which may be viewed as extensions of dual buses. Extensive analysis of folded bus schemes, such as DQDB ([IEE, Won89, Bis90, CGL90, CGL91, HM90, vA90, MB, Kam91, Rod90, WT93]) has been carried out. Dual bus schemes have also been analyzed extensively ([HM90, WT93, SS93, SS94b, SS94a, YC92]). Analysis has also been carried out for other bus schemes, such CRMA ([Hua94, SS94b, Nas90, vALZZ91]), which can use either dual buses or a folded bus, and for optical bus schemes ([KJ95]) such as HLAN ([]) and ORMA ([Ham97]). Besides these main bus protocols, there exist a variety of alternate bus schemes ([Lim90, LF82, CO90, WOS92, WO90, WOSC92, TBF83, TC]). The analysis for 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, since 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 works referenced above are concerned with such aspects of robustness.

Besides a folded bus routing, our network management architecture also uses routing based upon redundant trees, i.e. pairs of trees where 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 [IR88, ZI89], using s-t numberings ([LEC66]), and a more general method of constructing them was given in [MFBG99].

For the protocol which enables 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 a fairness, bandwidth efficiency, or delay are considered. The choice of appropriate metrics, in turn, depends crucially on the applications for our architecture. The choice of applications is outside the scope of our paper. Scheduling has been considered very extensively in the literature addressing optical stars and buses, referenced above, as well as for optical rings ([ZQ99, ZQ97]), optical switches ([Var00, LVB00, BCF99, LBR]), or WDM networks with arbitrary topologies ([HC98]). Another protocol aspect which we do not consider in this paper is the specific structure of the signalization channel. Several methods and their performance, in particular in terms of scalability with the number of nodes and of delay, have been presented in [SG00, CZA93, KFG92, RZ92, BM95, DG99].

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.

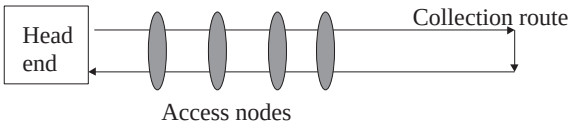

Figure 1: Collection route for an access network with a single head end and several nodes on the collection route.

3 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, which 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. Figure 1 shows a schematic of the setup we consider. In the next section, we will detail how to select such a route in a way which allows us to perform recovery. We will also describe how data is distributed in a manner which is robust to failures. For the remainder of this section, we simply assume that we have a collection routing, without regard to how the ordering on the collection routing is established.

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 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 4-fiber system) or may be carried over a single pair of fibers (for a 2-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 routing. 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 differs from that of other schemes in the following way: while most schemes are concerned with making use of unreserved bandwidth, we propose to make use of both

unreserved bandwidth and of *reserved bandwidth which 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.

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 routing, 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 which 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 which is accessed for the collection routing. 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, then 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 re-used, 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 which 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, then 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, then 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, then 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 routing.

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 routing).

4 Routing for recovery from link or node failures.

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

4.1 Routing for recovery from link failures.

An undirected graph $G = (N, E)$ is a set of nodes N and edges E. With each edge $[x, y]$ of an undirected graph, we associate two directed arcs (x, y) and (y, x). Each arc corresponds to one of the fibers forming the link. We assume that if an edge $[x, y]$ fails, then arcs (x, y) and (y, x) fail. Let us consider that we have a two edge-connected graph, or redundant graph $G = (N, E)$, i.e. removal of an edge leaves the graph connected.

We describe an algorithm for routing using access nodes in such a way that recovery is possible even in the event of any link failure. Our routing consists of two parts: a "collection" portion of the routing 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.

The collection portion is constructed as follows. We select a root node and build a depth-first search (DFS) numbering beginning at the root node. The routing is a walk which traverses nodes as they are considered by the DFS numbering algorithm. Figure 2 illustrates how our collection route is built. The full lines indicate edges used by our DFS tree and the

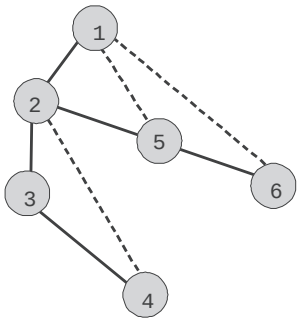

Figure 2: Example of a DFS tree.

nodes in dashed lines indicate edges which are not used for the DFS tree. Node 1 is selected to be the root. Next, nodes 2, 3 and 4, in order, are explored. Node 4 is a leaf of the DFS tree. The DFS algorithm next returns to node 3, from where it explores nodes 5 and 6, in order. Node 6 is a leaf of the DFS tree. The DFS then returns to node 5, 3, 2 in order and, finally, to node 1. The collection route is thus (1, 2, 3, 4, 3, 5, 6, 5, 3, 2, 1). We can prune the collection route by removing nodes in the following fashion. If the last i nodes of the collection routing were included earlier in the collection routing, then those i nodes may be removed. The pruned collection routing for our example is (1, 2, 3, 4, 3, 5, 6). In the sequel, we shall not make the distinction between a pruned and unpruned collection routing.

A collection routing will traverse every node at least once. Moreover, it will traverse each edge at most twice. If an edge is traversed twice, it is in opposite directions at each time. Thus, each arc is traversed at most once and so a single wavelength, in both directions, is sufficient to establish the collection portion of the routing.

The distribution portion of the routing is simply a directed spanning tree rooted at a root which is the last node traversed by the collection routing. We call this tree the primary tree. Robustness is afforded in the distribution section by constructing a secondary tree, which shares the root but no arcs with the primary tree, and such that removal of any edge (and its two associated arcs)leaves the root connected to every node on at least one of the trees. A single wavelength, used in both directions, is sufficient to construct the primary and secondary trees. Figure 3 shows a primary and secondary tree, in thick and thin lines, respectively . The root of the primary and secondary tree must therefore be able to perform wavelength conversion, by placing the traffic from the collection routing 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 routings sharing fibers, or as a four-fiber systems, where each fiber carries either one direction of the collection routing, or one direction of the distribution trees. Figure 4 shows the collection routing corresponding to the primary and secondary trees in 3.

We may now address how we ensure robustness against link failures. The crux of our

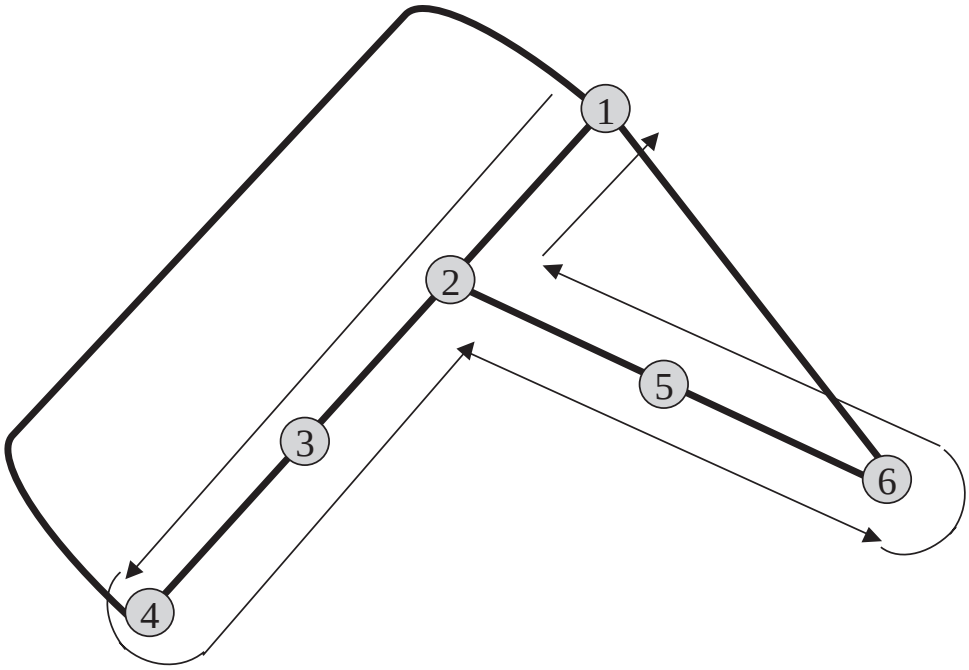

Figure 3: Example of two distribution trees.

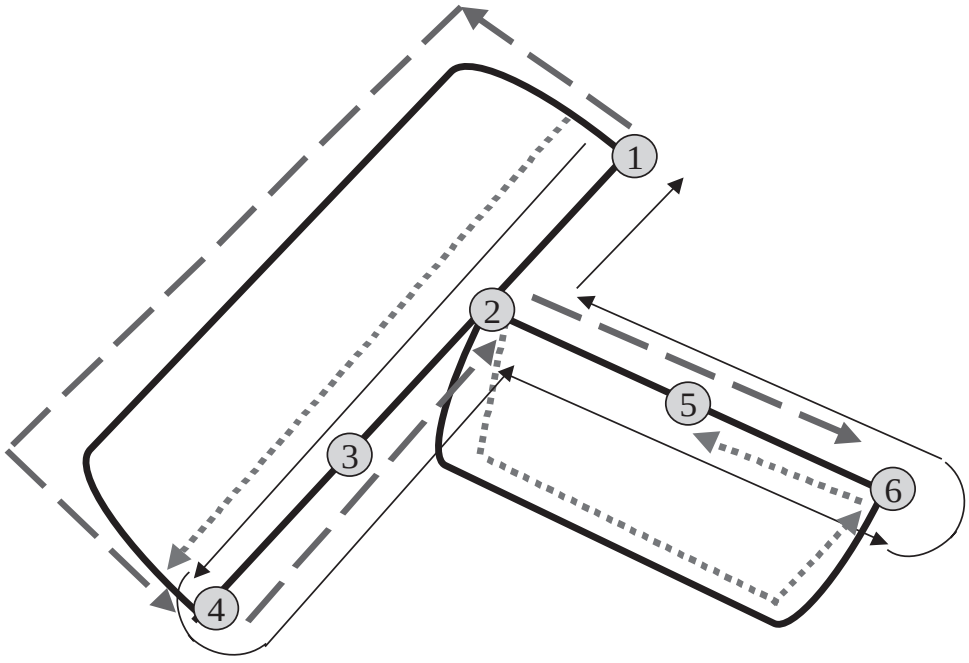

Figure 4: Example of collection routing.

algorithm lies in our method of performing link recovery in the collection portion of the routing. 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, then its failure leaves the collection routing unaffected. We therefore need only consider the case where link $[i, j]$ is not included in the DFS tree. We assume wlog that the node i is the ancestor of j. Failure of $[i, j]$ entails disconnection of all the descendants of j. From the DFS construction and the fact that that we have a 2-edge connected graph, some descendant of j must have an edge connecting it to some ancestor of j. Let k be the descendant of j 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. Figure shows the construction on which our argument is based.

We may now describe how recovery is effected. A new collection routing is constructed using the original collection routing. The original (unpruned) collection routing 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 routing is

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

The two routings are shown in figures 5 and 6, respectively. In 6, portions of the routings which do not use links used by the DFS tree are shown in dashed lines (the two dashed lines in our illustration are for traversing the link $[k, l]$ in both directions. For the routing after failure of $[i, j]$, we keep the old routing 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 which are descendants of j can be thought as being done in two parts. First, we explore the nodes which are not descendants of k in the DFS numbering. Next, we explore the nodes which 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 routing.

We may give an interpretation of the above routing in terms of the switching that needs to be done at nodes. For the distribution portion, each node which is downstream of the link failure in the primary tree switches to receiving on the secondary tree. On the collection portion, node 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). Node j connects (j_1, j) to (j, j_1). 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 node would be followed by connections ensuring explorations of those branchings. Then, the connections would resume as above.

4.2 Routing for recovery from node failures.

The construction of our routes for node failures has a similar flavor to our construction of routes for link failure recovery. We now assume that the graph G is two-vertex redundant.

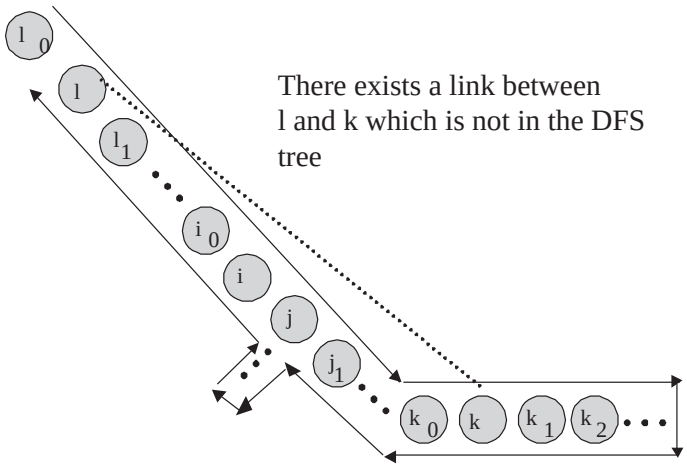

Figure 5: Collection routing before link failure.

We still maintain a collection and a distribution portion of our route. The collection portion is still built by traversing all nodes through a DFS search. A single wavelength in both directions may be used for the collection portion. The distribution portion of the network still relies on two trees. Again, a single wavelength in both directions is required for the distribution portion. Wavelength changing need only occur when transferring traffic from the collection portion to the distribution portion. Alternatively, no wavelength conversion need occur and, instead, we can have a four fiber system, with two fibers devoted to the collection routing, one fiber devoted to the primary distribution routing and one fiber devoted the secondary distribution routing. The major difference lies in the fact that we cannot rely on a single node to connect the collection portion of the routing 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.

Let us assume that the collection portion is constructed by visiting every node twice (a pruned version may easily be considered instead). Let node n_1 be the root of the DFS and let node n_2 be the node first visited by the DFS. Thus, the last arc of the collection routing is arc (n_2, n_1). Since we have two-node connected graph, we can construct a DFS so that a single arc originates at n_1 in the DFS tree. We assume that we conduct our DFS so that a single arc originates at n_1 in the DFS tree. In case of failure of a node other than n_1, then we perform recovery in a manner akin to that given for link failures, with some crucial

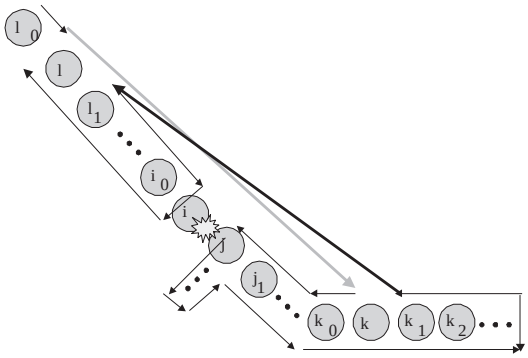

Figure 6: Collection routing after failure of link $[i, j]$.

modifications.

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

For each branching from j, the new collection routing 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 routing is constructed using the original collection routing. Figure 7 illustrates the original routings, with only two branchings shown from j . The original (unpruned) collection routing included, without loss of generality,

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

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 routing is

$$\Big(l_0^1, l^1, k^1, k_0^1, \ldots j^1, \ldots, k_0^1, k^1, k_1^1, k_2^1, \ldots, k^1, l^1, l_1^1, \ldots,$$

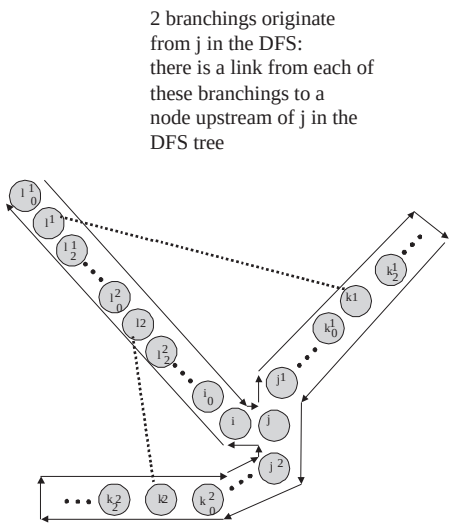

Figure 7: Collection routing before node failure.

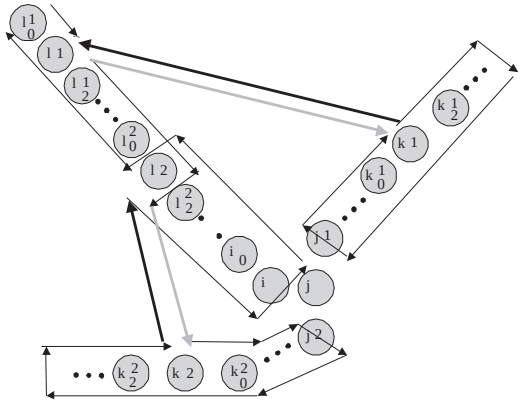

Figure 8: Collection routing after failure of node j.

$$l_0^2, l^2, k^2, k_0^2, \ldots j^2, \ldots, k_0^2, k^2, k_1^2, k_2^2, \ldots, k^2, l^2, l_1^2, \ldots$$
$$l_0^b, l^b, k^b, k_0^b, \ldots j^b, \ldots, k_0^b, k^b, k_1^b, k_2^b, \ldots, k^b, l^b, l_1^b, \ldots$$
$$l_1^2, l^2, l_0^2, \ldots, l_1^1, l^1, l_0^1, \ldots, n_2, n_1\Big) .$$

Figure 8 shows the routing after failure of node j for the example shown in figure 7.

We may now describe how we may modify the above routing when the l^xs are not all distinct. Suppose that l^x and l^{x+1} are the same. Then, after having visited the x^{th} 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 routing 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, then our collection routing can be effected by making n_2 the root of the DFS tree.

The distribution routing 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. A means of constructing such a pair of trees is given in [MFBG99]. From the algorithm in [MFBG99], 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 node 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 which traverses arc (n_2, n_1), then arc (n_1, n_2) will be included in the secondary tree. New nodes are explored by searching nodes which 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 node and ending at another covered node and such that all intermediate nodes are uncovered. Using the numberings given to the nodes, the path is traversed (except for the last node in the path) in one direction for the primary tree and in the reverse direction for the secondary tree. The root node, in this case n_1, would always be the starting point of a path inclusion in the primary tree. For all nodes adjacent to n_1, we may select to explore them from n_1. Thus, except for (n_1, n_2), no other arc originating at n_1 will be included in the secondary tree.

If any node n other than n_1 fails, then in the distribution portion of the routing, each node downstream of n in the primary distribution tree switches to receiving on the backup tree. All other nodes are unaffected by the failure. The difficulty arises when node n_1 itself if affected. Figure illustrates our discussion. In the collection portion of the routing, n_2 becomes the new root. Thus, at the end of the collection routing, n_2 has all collected packets. Node n_2 broadcasts on the secondary tree of the distribution routing. Since all nodes are downstream of n_1 in the distribution portion of the routing, all nodes switch to receiving on the backup tree. Thus, n_2 acts as the root on the backup tree. Thus, while in the case of routing for link failure recovery only we required a single wavelength changer at n_1, in the case of routing for node failure we require a wavelength changer at n_1 and one at n_2.

5 Recovery protocols and implementation issues.

This section demonstrates the feasibility of the protocol by outlining a simple yet flexible implementation. We consider that each node houses a switch, whose functionality we discuss in this Section. We use the numbering and spanning tree defined by the DFS to implement the protocol through actions and timing completely local to the switches. In particular, we describe a dynamically configurable link device that manages recovery on the access collection tree.

We begin by labeling links in terms of the numbering defined by the DFS. Each link from a switch is labeled as either an ancestor (A) or a descendant (D) link depending on the relative order of the switch 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 Figure 9. In the implementation, these labels

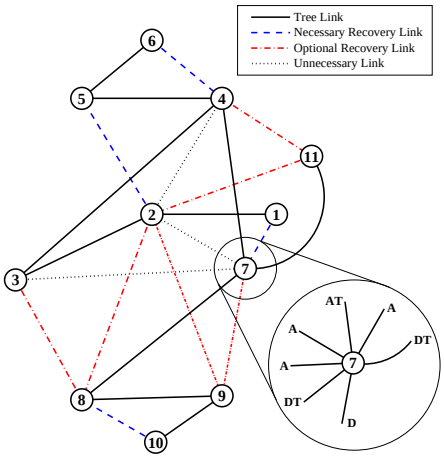

Figure 9: Example DFS tree and labeling. The graph shown is the NJ LATA network. An expanded version of node 7 is labeled with ancestor (A), descendant (D), and tree (T) markings. Link styles indicate their use in the access collection protocol.

are used to configure the access recovery device.

Now consider the case of a link failure. A failed link divides the 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 switches at either end of the link detect the absence of the pilot tone (P) 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 the existing collision avoidance protocol as necessary.

As shown in the discussion of our routing, 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 switches 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 switch at the end of the failed link and propagates along the collection route through a reserved failure marker (F) similar to that used for pilot tones (either in-band or sub-carrier multiplexed).

As detection of a failure propagates from switch to switch, each switch must decide whether or not it can effect recovery. In order to prevent switches from attempting to recover through ancestors in the lower part of the tree, switches are required to suppress such recovery when they detect a failure. This suppression is accomplished by asserting a suppression marker (S) over all non-tree descendant links. Owing to the triangle inequality, these suppression markers typically arrive at a switch before the switch 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 switch has

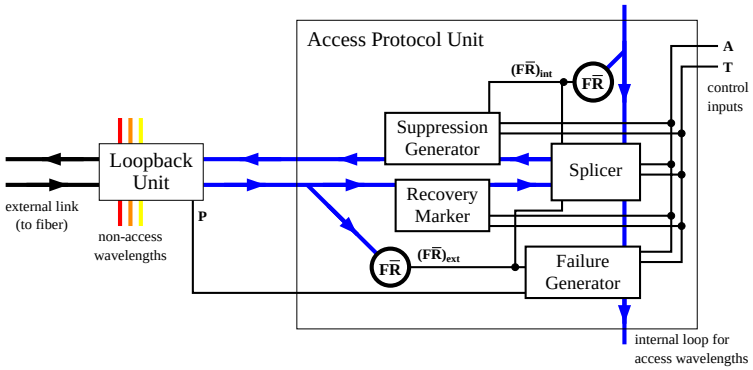

Figure 10: Block diagram of link control hardware. The access protocol unit implements the collection tree, while a slightly modified loopback unit handles loopback and wavelength splitting.

decided that it cannot realize recovery itself. Consider a series of switches below a link (or node) failure. Before deciding to splice in a non-tree ancestor link, a switch must wait to guarantee that suppression of such links has been asserted and must also wait to ensure that no switch upstream has already recovered. As the recovery decision process requires at least some input from the switch immediately upstream, part of this delay cannot be overlapped with the delay at that switch. This component of the delay thus accumulates along the path below a cut, requiring that switches 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 switches have attempted recovery, a switch must wait only for suppression, the time for which depends only on the propagation delays on the switches' links. Such a scheme can be realized by requiring a switch to mask failures until deciding that it cannot recover. Our implementation makes use of a recovery (R) marker for this purpose (again either in-band or sub-carrier multiplexed). As this masking serializes the parallelizable component of the delays at each switch, however, global recovery times are longer. The recovery marker is also used when recovery has been effected.

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 markers. A block diagram of the device appears in Figure 10. At this level, the device consists of a loopback and multiplexing unit and an access protocol unit. The access protocol unit consists of four components that implement aspects of protocol behavior.

The collection route is formed by using the switching fabric to loop a fiber through all devices in the switch. Figure 11 shows an example based on the DFS labeling of the NJ LATA network from Figure 9. The internal loop connects access wavelengths in a single cycle beginning with the tree ancestor, passing through all other ancestor links, then through non-tree descendant links, and finally looping through links to descendants in the collection tree before returning to the tree ancestor link. The ordering supports the operation of the protocol: failures detected upstream or immediately above the switch are detected at the

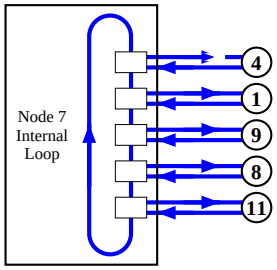

Figure 11: Example internal loop. The loop shown corresponds to that of node 7 from the example of Figure 9.

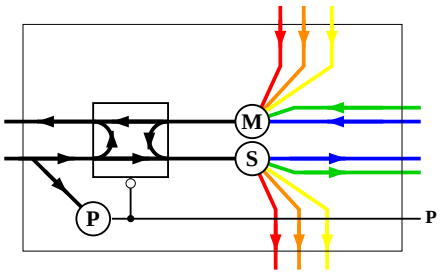

Figure 12: Loopback and multiplexing unit. This unit provides the standard pilot-tone-based loopback control and wavelength multiplexing and demultiplexing. Access wavelengths and the pilot tone marker are passed to the access protocol unit.

device attached to the tree ancestor link. After a delay to ensure the arrival of supression markers, failures are then passed through each non-tree ancestor link and propagated only if a link has been suppressed. If no ancestor link is available, the failure marker passes through the non-tree 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 non-tree ancestor link to be included for recovery from failure of the link to the tree ancestor (or of the ancestor switch itself). However, one can often select an ancestor link that also provides coverage for other link node failures. In Figure 9, 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 Figure 11.

A common loopback and multiplexing unit, shown in Figure 12 sits between the external link and the access protocol unit. This device monitors the pilot tone on the link and loops the signal back in the event of link failure. It also demultiplexes the incoming signal and selects those wavelengths configured for access to the access protocol unit. Finally, it multiplexes the outgoing signals onto the fiber. The result of pilot tone detection is passed

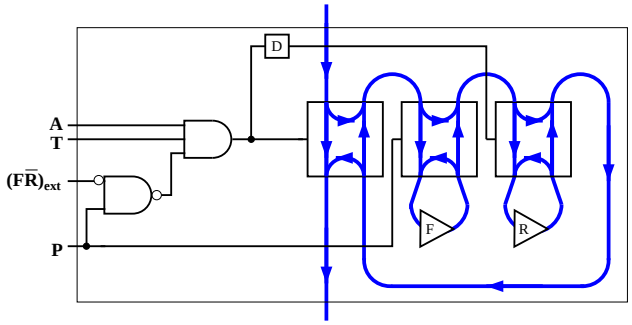

Figure 13: Failure generator. This component monitors a tree ancestor's pilot tone and failure marker and asserts failure to downstream access protocol units. The delay box ensures that suppression for non-tree descendant links has been processed before propagating a failure.

into the access protocol unit for use by the failure generator component.

The four components of the access protocol unit implement protocol behavior. In addition to the four components, the unit detects unrecovered failures on both the external and internal loop fibers through optical correlation with the combined marker signals. Neither of the markers need be recovered individually. Link labeling is encoded as two inputs to the unit, A and T; the logical D label translates to a low A input in the physical implementation. In the absence of failure, only the splicer affects optical signal propagation; the splicer must integrate tree links into the collection route. Tree descendants do not in fact require any functionality beyond that provided by the loopback and multiplexer unit; failure of the link to a tree descendant is resolved by simply looping the collection traffic back and waiting for the lower part of the tree to initiate recovery.

The failure generator component of the access protocol unit appears in Figure 13. For all but the tree ancestor link, this unit has no effect. For the tree ancestor, it monitors both the (unrecovered) failure signal and the pilot tone on the incoming link to detect failures. When either type of failure is detected, the 2x2 switch on the left flips, rerouting the traffic through the loop on the right. The middle 2x2 switch generates a failure marker in the case of pilot tone absence (otherwise, the signal already contains the failure marker). The right 2x2 switch asserts recovery until the failure has propagated through a tunable (electronic) delay element that serves to ensure that suppression markers from non-tree ancestor switches have arrived. Once the delay expires, this switch flips, removing the recovery marker from the signal and allowing the failure to propagate downstream in the internal loop.

The two components shown in Figure 14 implement suppression and failure masking once recovery has been realized. Although the suppression marker S is logically distinct from the failure marker F, mapping the two to the same physical signal simplifies the implementation and reduces the number of signals that must be generated and detected by the optical devices. For all but non-tree descendant links, both components in the figure do nothing. In the absence of a failure, tree descendant links are looped back onto themselves at either end by the splicer and do not carry data. When the descendant initiates recovery through the link, the device must avoid timing errors caused by delays between detection of the failure and insertion of the recovery marker. Unmasked failures can neither be allowed to propagate

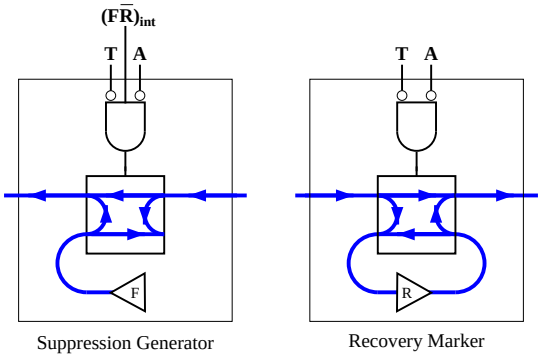

Figure 14: Components pertaining to non-tree descendant links. The suppression generator prevents attempts to recover within a failed circuit. The recovery marker logically removes the failure marker when a repair is effected.

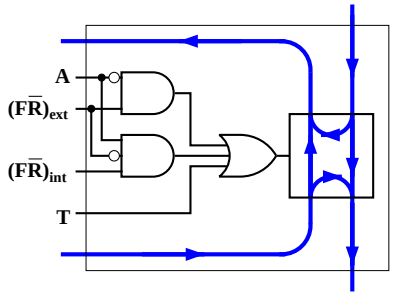

Figure 15: Link splicer. This component of the access protocol unit controls splicing a link into the access loop. In the absence of failure, the tree input T dictates the decision. The loopback unit overrides the splicing decision for failed links in the collection tree.

downstream in the internal loop, which might cause inappropriate recovery attempts by other switches, nor to return to the descendant, where they must be interpreted as suppression and may cause recovery to fail (or put the system into an unstable state). To avoid these scenarios, the recovery marker is not inserted dynamically on failure detection, but is instead added to the signal continuously by all units configured as non-tree descendants. As a consequence, the suppression generator must not use the signal from the recovery marker when asserting suppression. As the signal only contains data when a descendant initiates recovery (assuming one failure, this event cannot co-occur with suppression), the fact that it is dropped is inconsequential.

The last component in the access protocol unit is the splicer, which controls inclusion of the associated link into the collection route. Clearly, any link in the tree must be included. Two other types of links can be spliced into the route. First, when a failure is detected on the internal loop, a unit configured as a non-tree ancestor splices in its link provided that the link has not been suppressed. Second, when a non-tree descendant passes traffic upwards with failure asserted, that traffic is spliced into the collection route.

6 Conclusions.

We have described an architecture for optical local and metropolitan access in a way which allows for efficient and fair sharing of an access wavelength and which 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 trade-off 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

[BCF99] C.S. Baw, R.D. Chamberlain, and M.A. Franklin. Fair scheduling in an optical interconnection network. In *7th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems*, pages 56 –65, 1999.

[BDF+] A. Bianco, V. Distefano, A. Fumagalli, E. Leonardi, and F. Neri. A-posteriori access strategies in all-optical slotted WDM rings. In *Global Telecommunications Conference*.

[Bis90] C. Bisdikian. Waiting time analysis in a single buffer DQDB (802.6) network. In *Proceedings IEEE INFOCOM*, pages 610–616, 1990.

[BM95] M.S. Borella and B. Mukherjee. A reservation-based multicasting protocol for WDM local lightwave networks. In *Proceedings of the IEEE International Conference on Communications*, pages 1277–1281, 1995.

[BSD93] K. Bogineni, K.M. Sivalingam, and P.W. Dowd. Low-complexity multiple access protocols for wavelength-division multiplexed photonic networks. In *Jsac*, volume 11, pages 509–604, 1993.

[CDR90] M.-S. Chen, N.R. Dono, and R. Ramaswami. A media-access protocol for packet-switched wavelength division multiaccess metropolitan area networks. In *IEEE Journal on Selected Areas in Communications*, volume 8, pages 1048 –1057, 1990.

[CG89] I. Chlamtac and A. Ganz. Design alternatices of asynchronous WDM star networks. In *IEEE International Conference on Communications*, pages 23.4.1–23.4.5, 1989.

[CG99] D. Callahan and G. Grimes. An intelligent hub protocol for local area lightwave networks. In *Conference on Local Computer Networks*, pages 260–261, 1999.

[CGL90] M. Conti, E. Gregori, and L. Lenzini. DQDB under heavy load: performance evaluation and fairness analysis. In *Proceedings IEEE INFOCOM*, pages 133–145, 1990.

[CGL91] M. Conti, E. Gregori, and L. Lenzini. A methodological approach to an extensive analysis of DQDB performance and fairness. In *IEEE Journal on Selected Areas in Communications*, volume 9, pages 76–87, January 1991.

[CO90] I. Cidon and Y. Ofek. Metaring - a full duplex ring with fairness and spatial reuse. In *Proceedings IEEE INFOCOM*, pages 969–981, 1990.

[CZA93] R. Chipalkatti, Z. Zhang, and A.S. Acampora. Protocols for optical star-coupler network using WDM: performance and complexity study. In *IEEE Journal on Selected Areas in Communications*, volume 11, May 1993.

[DG99] E.H. Dinnan and M. Gagnaire. An efficient media access protocol for packet switched WDM photonic networks. In *Symposium on Performance Evaluation of Computer and Telecommunication Systems*, July 1999.

[Dow91] P.W. Dowd. Random access protocols for high-speed interprocessor communication-based on an optical passive star topology. In *Journal of Lightwave Technology*, volume 9, pages 799 –808, June 1991.

[GCJ+93] P.E. Green, L.A. Coldren, K.M. Johnson, J.G. Lewis, C.M. Miller, J.F. Morrision, R. Olshansky, R. Ramaswami, and E.H. Smith. All-optical packet-switched metropolitan-area network proposal. In *Journal of Lightwave Technology*, page 754, May 1993.

[GG94] A. Ganz and Y. Gao. Time-wavelength assignment algorithms for high performance WDM star based systems. In *IEEE Transactions on Communications*, pages 1827 –1836, May 1994.

[GK91] A. Ganz and Z. Koren. WDM passive star - protocols and performance analysis. In *Global Telecommunications Conference*, pages 9A.2.1–9A.2.10, 1991.

[Gui97] M. Guizani. High speed protocol for all optical packet switched metropolitan area networks,. In *International Journal of Network Management*, volume 8, pages 9–17, 1997.

[Ham97] M. Hamdi. ORMA: a high-performance MAC protocol for fiber-optic LANs/MANs. In *IEEE Communications Magazine*, volume 35, pages 110–119, March 1997.

[HC98] E.J. Harder and H.-A. Choi. Scheduling file transfers in WDM optical networks. In *Fifth International Conference on Massively Parallel Processing*, pages 186 –193, 1998.

[HKR+96] E. Hall, J. Kravitz, R. Ramaswami, M. Halvorson, S. Tenbrink, and R. Thomsen. The rainbow-ii gigabit optical network. In *IEEE Journal on Selected Areas in Communications*, volume 14, pages 814–823, June 1996.

[HKS87] 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. volume LT-5, pages 1782–1794, December 1987.

[HM90] E.Y. Huang and L.F. Merakos. On the access fairness of the DQDB MAN protocol. In *Proceedings of IPCC*, pages 325–329, 1990.

[HRS93] P.A. Humblet, R. Ramaswami, and K.N. Sivarajan. An efficient communication protocol for high-speed packet-switched multichannel networks. In *IEEE Journal on Selected Areas in Communications*, volume 11, pages 568 –578, May 1993.

[Hua94] E.Y. Huang. Analysis of cyclic reservation multiple access protocol. In *19th Conference on Local Computer Networks*, volume 2, pages 102–107, 1994.

[IEE] Distributed queue dual bus (DQDB) - subnetwork of a metropolitan are network (MAN). In *IEEE Standard 820.6*.

[IR88] A. Itai and M. Rodeh. The multi-tree approach to reliability in distributed networks. Number 79, 1988.

[JL98] T.S. Jones and A. Louri. Media access protocols for a scalable optical interconnection network, May 1998.

[JU92] H.B. Jeon and C.K. Un. Contention based reservation protocols in multiwavelength protocols in multiwavelength protocols with passive star topology. In *Proceedings of the IEEE International Conference on Communications*, June 1992.

[Kam91] A.E. Kamal. Efficient multi-segment message transmission with slot reuse on DQDB. In *Proceedings IEEE INFOCOM*, pages 869–878, 1991.

[KFG92] B. Kannan, S. Fotedar, and M. Gerla. A protocol for WDM star coupler networks. In *IEEE Transactions on Communications*, volume 40, pages 730–737, April 1992.

[KJ95] S. Kumar and A.P. Jayasumana. Request based channel access protocol on folded bus topology. In *20th Conference on Local Computer Networks*, pages 174 –183, 1995.

[LA95] D.A. Levine and I.F. Akyildiz. PROTON:a media access control protocol for optical networks with star topology. In *Proceedings of the 20th Annual Computer Science Conference*, volume 3, pages 158 –168, April 1995.

[LaM91] R.O. LaMaire. FDDI performance at 1 Gbit/s. In *IEEE International Conference on Communications*, pages 174 –183, 1991.

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

[LEC66] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of Graphs International Symposium*, pages 215–232, July 1966.

[LF82] J. Limb and C. Flores. Description of Fasnet - a unidirectional local area communication network. In *Bell System Technical Journal*, volume 61, pages 1413–1440, September 1982.

[LG97] A. Louri and R. Gupta. Hierarchical optical interconnection network HORN: scalable interconnection network for multiprocessors and multicomputers, January 1997.

[LGK96] B. Li, A. Ganz, and C.M. Krishna. A novel transmission scheme for single hop lightwave networks. In *Global Telecommunications Conference*, pages 1784 –1788, June 1996.

[Lim90] J. Limb. A simple multiple access protocol for metropolitan area networks. In *Proceedings of SIGCOMM*, pages 67–79, 1990.

[LK93] J.H. Laarhuis and A.M.J. Koonen. An efficient medium access control strategy for high-speed WDMmultiaccess networks. In *Journal of Lightwave Technology*, page 1078, May 1993.

[LVB00] V.P. Lang, E.A. Varvarigos, and D.J. Blumenthal. The λ-scheduler: A multiwavelength scheduling switch. In *Journal of Lightwave Technology*, volume 18, pages 1049 –1063, August 2000.

[MB] B. Mukherjee and S. Banerjee. Incorporating continutaion-of-message (COM) information, slot reuse, and fairness in DQDB networks. In *Division of Computer Science, University of California, Davis, CA, Technical Report CSE-90-42.*

[MB99] E. Modiano and R.A. Barry. Design and analysis of an asynchronous WDMlocal area network using a master/slave scheduler. In *INFOCOM '99*, volume 2, pages 900 –907, May 1999.

[MBL+97a] M.A. Marsan, A. Bianco, E. Leonardi, A. A. Morabito, and F. Neri. SR/sup 3/:a bandwidth-reservation MACprotocol for multimedia applications over all-optical WDM multi-rings. In *INFOCOM '97*, volume 2, 1997.

[MBL+97b] 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 *IEEE International Conference on Communications*, volume 1, pages 437 – 442, May 1997.

[MBL+99] M.A. Marsan, A. Bianco, E. Leonardi, A. Morabito, and F. Neri. All-optical WDMmulti-rings with differentiated qos. In *IEEE Communications Magazine*, volume 37, pages 58 –66, Feb. 1999.

[Meh90] N. Mehravari. Performance and protocol improvements for very high speed optical fiber local area networks using a passive star topology. volume 8, pages 520–530, April 1990.

[MFBG99] M. Medard, S. G. Finn, R. A. Barry, and R. G. Gallager. Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. October 1999.

[MHH98] M. Maode, B. Hamidzadeh, and M. Hamdi. A receiver-oriented message scheduling algorithm for WDM lightwave networks. In *Global Telecommunications Conference*, volume 4, pages 2333 –2338, May 1998.

[MJS00] M. Mishra, E.L. Johnson, and K.L. Sivalingam. Scheduling in optical WDMnetworks using hidden markov chain-based traffic predictors. In *IEEE International Conference on Networks*, pages 380 –384, 2000.

[Nas90] M.M. Nassehi. CRMA:an access scheme for high-speed LANs and MANs. In *Proceedings of the IEEE International Conference on Communications*, volume 4, pages 1697 –1702, 1990.

[NT90] K. Nosu and H. Toba. An optical multiaccess network with optical collision detection and optical frequency addressing. In *IEEE International Conference on Communications*, volume 3, pages 968 –975, March 1990.

[Rod90] M.A. Rodrigues. Erasure nodes: performance improvements for the IEEE 802.6 MAN. In *Proceedings IEEE INFOCOM*, pages 636–643, 1990.

[Ros89] F.E. Ross. An overview of FDDI: the fiber distributed data interface. In *IEEE Journal on Selected Areas in Communications*, volume 7, pages 1043 –1051, Sept. 1989.

[Ros90] F.E. Ross. Fiber distributed data interface: an overview. In *15th Conference on Local Computer Networks*, pages 6–11, 1990.

[RZ92] I. Rubin and Z. Zhang. Message delay analysis for TDMA schemes using contiguous-slot assignments. In *IEEE Transactions on Communications*, volume 40, pages 730–737, April 1992.

[SG00] T. Strosslin and M. Gagnaire. A flexible MACprotocol for all-optical WDM metropolitan area networks. In *Proceeding of the IEEE International Performance, Computing, and Communications Conference*, pages 567 –573, 2000.

[SGK87] G.N.M. Sudhakar, N.D. Georganas, and M. Kavehrad. A multichannel optical star LAN and its application as a broadband switch. volume 5, December 1987.

[SR96] S. Selvakennedy and A.K. Ramani. Analysis of piggybacked token-passing mac protocol with variable buffer size for WDM starcoupled photonic network. In *3rd International Conference on High Performance Computing*, pages 307 –312, 1996.

[SS93] O. Sharon and A. Segall. A simple scheme for slot reuse without latency in dual bus. In *IEEE/ACM Transactions on Networking*, volume 1, pages 96–104, February 1993.

[SS94a] O. Sharon and A. Segall. On the efficiency of slot reuse in the dual bus configuration. In *IEEE/ACM Transactions on Networking*, volume 2, pages 89–100, June 1994.

[SS94b] O. Sharon and A. Segall. Schemes for slot reuse in CRMA. In *IEEE/ACM Transactions on Networking*, volume 2, pages 269–278, June 1994.

[TBF83] F.A. Tobagi, F. Borgonovo, and L. Fratta. Expressnet: a high performance integrated-services local area network. In *IEEE Journal on Selected Areas in Communications*, volume SAC-1, pages 898–913, November 1983.

[TC] W.C. Tseng and B.U. Chen. D-Net: a new scheme for high data rate optical local area network.

[vA90] H.R. van As. Performance evaluation of bandwidth balancing in the DQDB MAC protocol. In *Proceedings of EFOC/LAN Conference*, 1990.

[vALZZ91] 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 *Proceedings of the EFOC/LAN Conference*, pages 262–272, 1991.

[Var00] E.A. Varvarigos. The "packing" and the "scheduling packet" switch architecture for almost all-optical lossless networks. In *Journal of Lightwave Technology*, volume 18, pages 1049 –1063, AUGUST 2000.

[WH98] L. Wang and M. Hamdi. Efficient protocols for multimedia streams on WDMAnetworks. In *Twelfth International Conference on Information Networking*, volume 1, pages 241 –246, 1998.

[WO90] G. Watson and S. Ooi. What *should* a Gbits/s network interface lool like. In North-Holland, editor, *Protocols for High-Speed Networks*, volume II, pages 237–250, Amsterdam, 1990. M.J. Johnson, editor.

[Won89] J.W. Wong. Throughput of DQDB networks under heavy load. In *Proceedings of EFOC/LAN Conference*, pages 146–151, 1989.

[WOS92] H.-T. Wu, Y. Ofek, and K. Sohraby. Integration of synchronous and asynchronous traffic on the MetaRing architecture and its analysis. In *Proceedings of the IEEE International Conference on Communications*, 1992.

[WOSC92] G. Watson, S. Ooi, D. Skellen, and D. Cunningham. HANGMAN Gbit/s network. In *IEEE Network Magazine*, July 1992.

[WT93] G.C. Watson and S. Tohme. S++-anew MAC protocol for Gb/slocal area networks. In *IEEE Journal on Selected Areas in Communications*, volume 11, pages 531–539, may 1993.

[YC92] M.C. Yuang and M.C. Chen. A high performance LAN/MANusing a distributed dual mode control protocol. In *IEEE International Conference on Communications*, volume 1, pages 11 –15, 1992.

[YGK96] A. Yan, A. Ganz, and C.M. Krishna. A distributed adaptive protocol providing real-time services on WDM-based LANs. In *Global Telecommunications Conference*, volume 14, pages 1245 –1254, June 1996.

[ZI89] A. Zehavi and A. Itai. Three tree-paths. In *Journal of Graph Theory*, volume 13, pages 175–188, 1989.

[ZQ97] X. Zhang and C. Qiao. Pipelined transmission scheduling in all-optical TDM/WDMrings. In *Sixth International Proceedings*, volume 37, pages 144 –149, 1997.

[ZQ99] X. Zhang and C. Qiao. On scheduling all-to-all personalized connection and cost-effective designs in WDMrings. In *IEEE/ACM Transactions on Networking*, volume 7, pages 435–445, 1999.