1 POINT-TO-MULTIPOINT PATH ESTABLISHMENT SCHEMES TO SUPPORT MULTICASTING IN WDM NETWORKS

Taieb F. Znati[†,§]
Tawfig Alrabiah[†]
Rami Melhem[†]

[†]Department of Computer Science
[§]Telecommunications Program
University of Pittsburgh
Pittsburgh, PA 15260
(znati, tawfig, melhem)@cs.pitt.edu

Abstract:
Next generation Internet multimedia applications will increase the need for high throughput and low latency. *Wavelength Division Multiplexing* (WDM) optical networks have the potential for achieving these two goals by offering unprecedented high bandwidth. One very important aspect of new Internet services is the support of multicasting. As such, support for multicasting in WDM networks becomes crucial if these networks were to play an efficient role in the next generation Internet. Multicasting in WDM networks supporting multimedia applications can be viewed as the process of taking a group communication request and selecting a multicast tree that satisfies the quality of

service (QoS) requirements, in terms of bandwidth and end-to-end delay, of the underlying application.

In this paper, we present a new class of low-cost, bounded-delay multicast heuristics for WDM networks. The heuristics use various techniques to establish a tree of semilightpaths between a source and a group of destination nodes. The unique feature of these heuristics is that they decouple the cost of establishing the multicast tree from the delay incurred by data transmission due to lightwave conversion and processing at intermediate nodes along the transmission path.

INTRODUCTION

The Next Generation Internet (NGI) technology will, for the most part, be driven by the increasing need for high throughput and low latency. Optical networks have the potential for achieving these two goals by offering unprecedented bandwidth in a medium that is free from inductive and capacitive loadings, thus relaxing the limitations imposed on the bandwidth-distance product [10, 15]. Optical fibers may be used for transmitting terahertz signals in low attenuation passbands while maintaining low error rates and low sensitivity to noise [21, 11].

Although different types of photonic switching networks have been reported and demonstrated, Wavelength Division Multiplexing (WDM) has emerged as one of the most attractive approaches for data transfer in interconnection networks [5, 11, 7, 19]. WDM divides the optical fiber link spectrum into several channels, each corresponding to a different wavelength[1]. An incoming wavelength in one input port can be routed to one or more output ports. Due to electromagnetic interference, however, the same wavelength coming from two separate input ports cannot be routed to the same output port.

In *single-hop* WDM networks, the optical layer provides a *lightpath* to the higher layers. A lightpath is an all-optical transmission path between a source and a destination which uses the same wavelength between the intermediate nodes along the path. Each lightpath provides a full wavelength's worth of bandwidth to the higher layer. Lightpaths enable an efficient utilization of the optical bandwidth since data transmission through a lightpath does not require wavelength conversion or electronic processing at intermediate nodes. Therefore, an all-optical path between two access nodes can exploit the large bandwidth of optics without the overhead of buffering and processing at intermediate nodes.

In optical networks, however, the number of wavelengths available at the intermediate nodes and the tunability of the optical transceivers at these nodes are both limited. These constraints, combined with the fact that optical networks are usually large in size, make all-optical networks practically infeasible [4]. To increase the probability of successfully establishing a connection between a source and a destination, *multihop* WDM networks allow conversion between different wavelengths to take place at the intermediate nodes[2]. The resulting path is referred to as *semilightpath*.

Despite the renewed popularity of WDM, it has not been easy to bridge the gap between the Internet and optical networks. Much of the focus of optical communications research has been on transmission, a link property, rather than on networking. The feasibility of economical all-optical wideband amplifiers and multi-channel links has been demonstrated, thus proving WDM to be an effective transmission solution. However, optical networking, a property of the overall system and its dynamics, still remains a challenge to the research community. Addressing this challenge requires the identification of the synergies between the two technologies that will enable the wide deployment of Internet protocols and services on advanced optical networks. One very important aspect of the new Internet services is multicasting.

Multicasting, which is currently supported in most common commercial operating systems and routers, is expected to play a major role in the Internet. It provides the means to deliver messages to multiple destinations as a network service without requiring message replication by the end systems. As such, support for *native* WDM multicast is essential if WDM networks are to play an effective role in the next generation Internet.

Multicasting requires building a *point-to-multipoint tree* of paths between the source node and a *group* of destination nodes. The focus of the early research work on multicasting aimed at extending the LAN-style multicasting across a wide area network by developing heuristics for approximate solutions to the Steiner problem. In this context, the objective was to find a minimum cost tree that connects all the multicast nodes. The heuristics developed aimed at finding a low cost multicast tree, given a connectivity graph and single cost metric; the goal was to minimize the sum of link metrics in the tree.

The primary limitation of using the previous approaches for multimedia applications rests with their sole consideration of low cost. In addition to low cost, multimedia applications have different demands in terms of bandwidth, reliability, delay and jitter. A key property of multimedia data is its time dependency. The support of sustained streams of multimedia objects, over a period of time, requires the establishment of reliable, low delay and low cost source to destination routes. Nevertheless, the objective is not to develop a strategy which produces the lowest possible end-to-end delay, but a strategy to ensure that the data traffic arrives within its delay bound, thereby allowing a tradeoff between delay and cost. Thus the objective is to produce a minimal cost tree which guarantees bounded end-to-end delay between all source-destination pairs.

Finding the minimum delay paths from a source to a set of destination nodes can be achieved in polynomial time using one of the well-known shortest path algorithms. However, finding a delay-bound, minimum cost multicast tree is at least as complicated as the classical Steiner Minimum Tree (SMT), which is known to be NP-hard. Finding an approximation solution for the SMT problem has been the focus of research in multicasting and several algorithms for constructing multicast trees have been developed for traditional networks.

It must be mentioned, however, that there is a fundamental difference between the traditional networks and the WDM networks regarding the multicasting issue. This difference reflects in their optimization objectives. Multicasting in traditional networks has focused on bounding the end-to-end delay while keeping the cost minimum. In WDM networks, wavelengths constitute critical resources and the cost of wavelength conversion plays an important factor in the optimization objectives. Furthermore, the optimization objectives in WDM networks involve more than one cost metric. The tradeoff between the desire to reduce wavelength convergence to a minimum and the need to increase the likelihood of successfully establishing a multicast tree brings about challenges that are unique to WDM networks.

This paper focuses on the algorithmic issues related to establishing trees of semilightpaths between a source node and a group of destinations nodes. More specifically, we propose a set of heuristics that can be used to build minimum-cost, delay-bounded multicast trees in WDM networks. Contrary to many solutions proposed in the literature, our approach decouples the cost of building the multicast tree from the data transmission delay due to wavelength conversion and processing at intermediate nodes along the semilightpath.

The rest of this paper is organized as follows: We first review the research work related to multicasting in traditional and WDM networks. We then formalize the multicast problem in WDM, in terms of minimizing the cost and bounding the delay, and describe a graph model to characterize this problem. In the following section, we describe three heuristics that can be used to build minimum-cost, bounded-delay multicast trees. In the last section, we summarize the contributions of this paper and discuss future work.

RELATED WORK

The minimum-cost, delay-bounded multicast tree problem, thereof referred to as *Delay-Bounded Steiner Minimum Tree* (DBSMT), can be viewed as a derivative of the Steiner Minimal Tree (SMT) problem in which the delay from the source to the destinations is bounded [12]. Both problems are NP-hard and several heuristics have been proposed to address these problems.

Based on their design objectives, the multicast heuristics proposed in the literature can be viewed as members of one of three possible classes. The first class includes heuristics which are designed to accommodate the Internet environment. The second class includes heuristics which aim at reducing the cost of the multicast tree, while bounding the end-to-end delay. The third class of heuristics focuses on multicasting in WDM networks.

The Internet community proposed different algorithms to create multicast trees, including Distance Vector Multicast Routing Protocol (DVMRP), Multicast Open Shortest Path First (MOSPF), Protocol Independent Multicasting (PIM), and Core Based Trees (CBT) [22, 18, 9, 1]. These protocols are designed to work specifically with the IP environment and take advantage of the IP routing protocols such as RIP and OSPF. They focus on issues related to scalability and reduced communication overhead. These protocols do not address the performance requirements of the underlying applications and may not be directly applicable to WDM networks.

The second class of multicast algorithms aims at achieving good approximations to DBSMT problem. These heuristics include Least Delay Heuristic (LDH), Least Cost Heuristic (LCH), Dynamic Programming Heuristic (DPH), and Bounded Shortest Multicast Algorithm (BSMA). They differ in terms of their objectives and the techniques used to achieve them.

LDH builds the shortest delay path tree, using Dijkstra's sink shortest path tree algorithm, without any consideration of the cost [6]. Therefore, the heuristic's time complexity is $O(n^2)$. LDH finds a solution to the multicast problem, if one exists. The tree total cost, however, is not optimal.

LCH is similar to LDH except that it tries to reduce the tree cost by building the least cost tree instead of the least delay tree [25]. In the worst case, this heuristic requires finding the minimum cost tree as well as the minimum delay tree. If there are no delay violations then in the worst case the cost of the tree produced by LDH does not exceed $|D|$ times the cost of the optimal tree, where D is the size of the multicast group. The time complexity in this case is $O(n^2)$.

DPH finds the minimum cost constrained paths between any two multicast nodes [12]. DPH assumes that the link delay, $\delta(u, v)$, between any two nodes u and v, and the delay bounds, Δ_i, are integers. The algorithm first computes the minimum cost constrained delay path between any two multicast nodes using a dynamic programming approach. Using the least constrained paths, the algorithm then builds the closure graph, a fully connected graph consisting of the multicast nodes. The cost and delay of an edge in the closure graph are the same as the cost and delay of the minimum cost constrained path. Using the closure graph, the algorithm computes the minimum spanning tree to produce the final delay-constrained tree. The approach used to build the spanning tree is similar to the one used by the Distance Network Heuristic [13]. The algorithm accuracy and complexity depend on the granularity of the delay values. If the granularity of the delay values is very small, DPH complexity is high. On the other hand, a large delay value granularity reduces the computation time but results in poor performance.

BSMA starts initially with a minimum delay path tree and incrementally reduces the tree cost by iteratively replacing high cost relay-paths with lower cost paths [25]. A relay-path is a path on which all internal nodes are relay-nodes. Relay-nodes are nodes of degree two which are not multicast nodes, excluding the end nodes. This iterative process terminates when no relay-paths can be replaced. The resulting tree represents the final delay-bounded multicast tree. The time complexity of the heuristic is $O(n\ log(n) * k\, n^2) = O(k\, n^3\ log(n))$, where k is the number of shortest paths to be explored when a relay-path is removed.

The above heuristics deal with DBSMT from different perspectives. The simple heuristics, LDH and LCH, use the shortest path tree algorithm to build the multicast tree. Due to the fact that they perform very limited cost optimization, these heuristics do not produce "good" approximations. On the other hand, DPH and BSMA use more elaborate approaches to find a low cost tree but incur a much higher time complexity. DPH may lead to a good approximation if the granularity of the delay bound is large. However, if the granularity of the delay bound is fine, the time complexity of the algorithm may be very high, thereby limiting its practical implementation. BSMA results in a feasible least-cost, delay-bounded tree, if one exists. The algorithm also has high complexity making its implementation impractical for large networks. Most importantly, these heuristics do not consider the cost of wavelength conversion which limits their applicability to WDM networks.

The third class of heuristics focuses on routing and channel assignment in WDM networks. For a given number of routing requests, the objective is to minimize the number of wavelengths used to satisfy the routing requests [8, 3]. Most of this work focused on point-to-point routing in single-hop and multi-hop WDM networks. More recently, the focuse has shifted to point-to-multipoint routing [20, 17, 16]. The metric used in building point-to-multipoint paths must take into consideration the cost of using a specific wavelength on a given link, the cost of converting an incoming wavelength into one or more outgoing wavelengths, and the time required to convert an incoming wavelength into one or more outgoing wavelengths. Most of the proposed solutions that address WDM multicast communication combine these costs to produce a weighted cost structure which is then used to produce a multicast tree. The DBSMT problem, however, consists of two sub-goals, namely minimizing the multicast tree cost and bounding the delay due to switching and wavelength conversion. Our approach decouples the cost optimization from bounding the delay. Furthermore, given that in WDM networks, the need for wavelength conversion is expected to be low and must be avoided as much as possible, our approach is first to build a close-to-optimal minimal cost tree and deal with delay violations whenever they occur.

DBSMT PROBLEM FORMULATION

What makes the problem of building WDM multicast trees fundamentally different from the multicasting problem in traditional networks is the cost structure associated with the underlying network. The metrics used in building point-to-multipoint paths in WDM networks must take into consideration the cost of using a specific wavelength on a given link, the cost of converting an incoming wavelength into one or more outgoing wavelengths, and the delay required for wavelength conversions.

Consider a WDM optical network with N nodes, M optical links, and K wavelengths. This network can be modeled by a graph $G = (V, E, \Omega)$ where:

- V denotes a set of vertices, corresponding to the network nodes,
- E denotes a set of edges, corresponding to the network optical links. A link incident from node $u \in V$ onto node $v \in V$ is denoted as (u, v).

- $\Omega = \{\nu_1, \nu_2, \cdots, \nu_K\}$ represents the set of all available wavelengths in the network.

Each link $l = (u, v) \in E$, connecting node $u \in V$ to $v \in V$, is associated with a set $\Omega_l \subseteq \Omega$ of available wavelengths. Furthermore, each link $l = (u, v) \in E$, is associated with the following costs and delays:

- a *wavelength usage cost*, $\omega_l(\nu)$, for each $\nu \in \Omega_l$, which represents the cost of using ν on link l.

- a *wavelength conversion cost*, $c_l(\nu_i, \nu_j)$, representing the cost of converting a wavelength, ν_i, on some incoming link $e = (x, u)$ into a wavelength ν_j on the outgoing link $l = (u, v)$, where $\nu_i \in \Omega_e$ and $\nu_j \in \Omega_l$. This cost is infinity if such a conversion is not possible at u , and is reduced to zero if $\nu_j = \nu_i$.

- a *wavelength conversion delay*, $\delta_l(\nu_i, \nu_j)$, representing the time required to convert a wavelength, ν_i, on incoming link $e = (x, u)$ into a wavelength, ν_j, on outgoing link $l = (u, v)$, where $\nu_i \in \Omega_e$ and $\nu_j \in \Omega_l$. This delay is reduced to zero if $\nu_j = \ nu_i$.

The wavelength usage cost can be used to enforce a specific wavelength assignment strategy. The First-Fit policy, for example, selects the wavelength with the lowest index to assign to a new connection request [14]. The MaxSum policy, on the other hand, assigns the wavelength that maximizes the wavelength path capacity (WPC), defined as the capacity of the most congested link along the path, over all potential paths and all wavelengths [2]. The wavelength usage cost can be used to enforce either of these two policies. In the case of the First-Fit policy, the cost of available wavelengths on a given link can be assigned in a way such that the wavelength with the lowest index bears the lowest cost. Similarly, setting the cost of the wavelength that maximizes the WPC to be the lowest enforces the MaxSum wavelength assignment policy.

The wavelength conversion cost can be used to minimize the number of wavelength conversions along a path. In some cases, it can also be used to discourage wavelength conversion at nodes that perform critical functionalities within the network, such as nodes interconnecting different networks, or at nodes that are heavily loaded. The wavelength conversion cost can be also used to adapt to changing network environments and to reflect the availability of wavelengths at a given node. Links incident from interconnection nodes, heavily loaded nodes, or nodes with a reduced number of available wavelengths incur higher wavelength conversion costs.

The wavelength convergence delay reflects the significance of the time required to convert an incoming wavelength into an outgoing wave length in comparison to the time required to transmit the signal over the optical link. Furthermore, in multicast configuration, wavelength conversion can take place between an incoming wavelength on a given input port and a number of outgoing wavelengths on several output ports, as required by the underlying multicast tree. Unless the wavelength conversions proceed in parallel, the required wavelength convergence delay can become even more significant in comparison with the transmission delay.

The cost structure proposed in this paper decouples the conversion cost from the conversion delay. This is necessary to account for the fact that in a multicast tree two connections from the source to two different destinations which share some nodes along the path face conversion jointly at the shared node but suffer the conversion delays separately. Therefore, the conversion cost must be accounted for only once, while the end-to-end delay requirement must be verified for each connection separately. Consequently, the DBSMT problem can be viewed as consisting of two sub-goals, namely minimizing the multicast tree cost and bounding the delay. Given a source and a set of predefined multicast nodes, the DBSMT problem consists of finding the minimal cost tree which contains all selected multicast nodes. In addition to finding the optimal multicast tree that satisfies the delay requirements, a procedure to assign a specific wavelength $\nu_i \in \Omega_l$ on each link $l = (u, v)$ along the path is required. More specifically, the DBSMT problem can be stated as follows:

Let $\mathcal{G}$ be a multicast group composed of a source $s \in V$ and a set of multicast nodes $D \subseteq V$. Every destination node $d \in D$ has a delay bound Δ_d. The objective is to find:

1. a multicast tree $T = (V', E')$, $V' \subseteq V$ and $E' \subseteq E$, connecting s to all $d \in D$, and

2. a wavelength assignment function, $\nabla_P \; : \; E' \longrightarrow \Omega$, which assigns a wavelength, $\nabla_P(l) = \nu \in \Omega_l$ to link, l for each path, $P(s, d)$, connecting source s to a given destination $d \in D$,

such that the total cost of the tree is minimum and the delay from the source s to any multicast node $d \in D$ does not exceed Δ_d.

In order to formalize the cost of the multicast tree and the delay on a given path, define $P(s, d)$ as the sequence of links $(s, v_1), \cdots, (v_i, v_{i+1}), \cdots, (v_k, d)$ forming the multicast tree path from source s to node d. Furthermore, let $Pred(l) = (v_{i-1}, v_i)$, be the predecessor of link $l = (v_i, v_{i+1})$ on path $P(s, d)$. Note that if two paths, $P(s, d_1)$ and $P(s, d_2)$, share the same link, l, the $Pred(l)$ on $P(s, d_1)$ is the same as $Pred(l)$ on $P(s, d_2)$. This follows directly from the tree property of the multicast graph.

Further, define $\nabla(l) = \{\nabla_P(l) | l \in P(s, d) \; \forall d \in D\}$ to be the set of all wavelengths currently used on link l by all the source to destination paths which include l. Finally, let $\mathcal{W}_l = \{ \; (\mu, \nu) \mid \exists P \supseteq l \; : \; \nabla_P(Pred(l)) = \mu \; and \; \nabla_P(l) = \nu \; \}$ be the set wavelength pairs used on link l and its predecessor, respectively, along the same path. The following observations can be made regarding paths of a multicast tree in WDM networks:

- Two paths, $P(s, d_1)$ and $P(s, d_2)$, can share some link, l, without necessarily sharing the same wavelengths on l. This may be required if at one node along either one of the two paths, further conversions are no longer possible without violating the delay requirements.

- If two paths share the same wavelength on a given link, l, then they share the same wavelength on $Pred(l)$. This is required to eliminate unnecessary wavelength conversions which can only increase the cost and delays of the final multicast tree.

Based on the above, the DBSMT problem can be formalized as follows: Find a multicast tree $T(V', E')$ and a wavelength assignment function, ∇_P for each each path P, such that:

$$Cost(T) = \sum_{l \in E'} \sum_{\nu \in \nabla(l)} \omega(l, \nu) \; + \sum_{l \in E'} \sum_{(\mu,\nu) \in \mathcal{W}_l} c_l(\mu, \nu) \quad \text{is minimum} \tag{1.1}$$

subject to:

$$Delay(P(s,d)) = \sum_{l \in P(s,d)} \delta_l(\nabla_P(Pred(l)), \nabla_P(l)) < \Delta_d \quad \forall d \in D \tag{1.2}$$

The DBSMT problem can be viewed as a derivative of the Steiner Minimal Tree (SMT) problem in which the delay from the source to the destinations is bounded. Minimizing the tree cost is inherently NP-hard, while bounding the delay can be achieved in polynomial time. When the two parameters are combined, the problem is NP-hard and, therefore, has to be approached by a heuristic method. What makes the DBSMT problem in WDM networks inherently more complex than in traditional networks is that path selection in WDM networks is compound with wavelength assignment along the path. Heuristics for WDM networks, which take delay, cost and wavelength assignments into consideration, are discussed next.

MULTICAST HEURISTICS FOR WDM NETWORKS

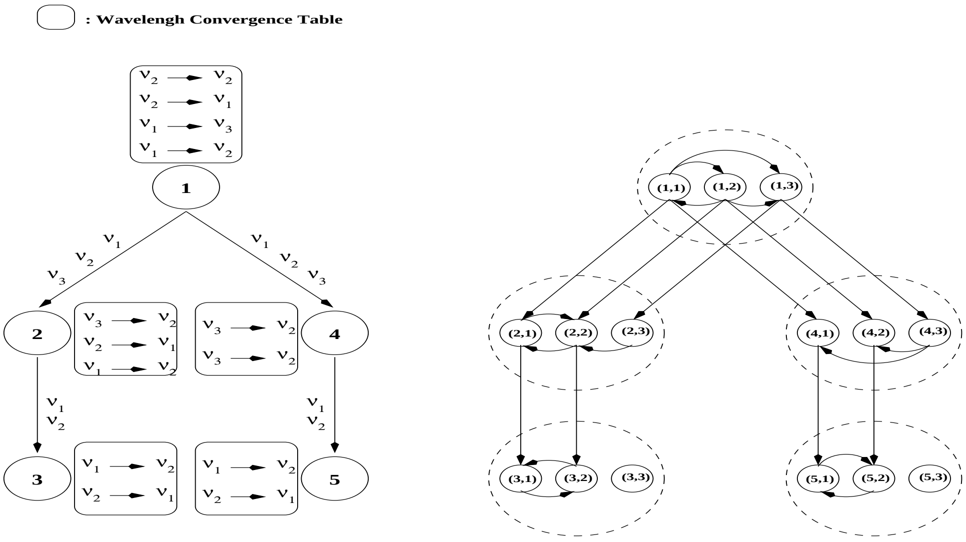
Given that in a WDM networks, the number of delay violations is expected to be low, an efficient approximation approach to the DBSMT problem is to first build a close-to-optimal least-cost tree and deal with delay violations as a special case. In the following, we show how we transform the original graph representing the network into an expanded graph to facilitate route computation and wavelength assignment. We then describe three new heuristics which can be used to build a low-cost, bounded-delay multicast tree for WDM networks.

Graph Transformation

Route selection and wavelength assignment are difficult to achieve based on the original graph. This is mostly due to the fact that the cost metrics and the delay computation must be considered dynamically during the path selection process. To make path selection and delay computation more tractable, an expanded directed and weighted graph, $\mathcal{A}(\mathcal{M},\mathcal{L},\Omega)$, derived from the original graph is used [4]. The expanded graph, $\mathcal{A}(\mathcal{M},\mathcal{L},\Omega)$, is defined as follows:

- $\mathcal{M}$ represents a set of *nodes* , (m,ν), where $m \in V$ is a node in the original graph, and $\nu \in \Omega$ is a wavelength in the network.
- $\mathcal{L}$ represents a set of links connecting pairs of nodes in M.
- Ω represents a set of all available wavelenghs in the network.
- an edge between two nodes (m_1,ν) and (m_2,ν) exists in the expanded graph, $\mathcal{A}(\mathcal{M},\mathcal{L})$, only if there exists a link l from node m_1 to node m_2 in the original graph, $G(V,E,\Omega)$, and the wavelength ν is available on link l.
- an edge between (m,ν_1) and (m,ν_2) exists only if wavelength conversion from ν_1 to ν_2 is available at node m.

The costs and delay associated with the links of the original graph, namely wavelength usage cost, wavelength conversion cost and wavelength conversion delay, are directly mapped into the corresponding costs and delay of the links of the expanded graph. Figure 1.1(a) shows the expanded graph resulting from the original graph in Figure 1.1(b).



(a) Original graph

(b) Auxiliary graph

Figure 1.1 Graph transformation

Using the expanded graph, we describe three new heuristics, *Least Delay Replacement*, *Least Cost First* and *Least Delay First*. The first heuristic decouples the cost optimization from bounding the delay by first building a low cost tree and then handling any delay violations that may occur in the tree. The other two heuristics incorporate delay bound checking into the process of building the minimum cost tree. That is, if any delay violation occurs, it is dealt with immediately. This is achieved by searching for a new delay-bounded path with the lowest cost. The two heuristics differ in the way they select low-cost paths which may result in different low-cost multicast trees.

Least Delay Replacement (LDR)

The basic idea of the LDR is to first build a least-cost tree using a Shortest Path Heuristic (SPH) based approach [26]. The link cost takes into consideration both the cost of wavelength conversion and the cost of using a specific wavelength on the link. Using this least-cost tree, delay bounds are checked and violations are removed accordingly.

The SPH based procedure to compute the least-cost tree can be described as follows:

Shortest Path Heuristic() {

Let $PATH(d, T)$ represent the least-cost path from a multicast node $d \in D$ to a tree T.

Step 1 : Set $i = 1$ and $M_1 = \{\emptyset\}$. Construct a subtree, $T_1 = \{s\}$, consisting of the source node s.

Step 2 : Set $i = i + 1$. Find the least-cost node, d_i, to T_{i-1} such that $d_i \in (D - M_{i-1})$ (ties are broken arbitrarily). Construct a new subtree, T_i, by adding all edges and nodes in $PATH(d_i, T_{i-1})$ to T_{i-1}. Set $M_i = M_{i-1} \cup \{d_i\}$.

Step 3 : If $|M_i| < |D|$ then go to **Step 2** else DONE: T_i is the final least-cost multicast tree.

}

The above algorithm is similar to Prim's Minimum Spanning Tree algorithm. The time complexity of SPH is $O(n^2\, log(n))$, where n is the number of nodes in the network. However, the need to compute the least-cost path at each stage raises the total complexity of the algorithm to $O(|Z| \times n^2)$, where $Z = D \cup \{s\}$, s is the source node and D is the set of multicast nodes. It was shown by simulation that on average the additional cost of the trees produced by this algorithm is no more than 5% above the cost of the corresponding optimal Steiner trees [24]. However, in the worst case, the cost of this algorithm is $(2 - 2/|Z|) \times C(T_{opt})$, where $C(T_{opt})$ is the cost of the corresponding optimal Steiner tree[3].

The LDR heuristic uses the expanded graph, $\mathcal{A}(\mathcal{M}, \mathcal{L})$, and invokes SPH to produce a least-cost multicast tree. The SPH, however, does not guarantee that the end-to-end delays on all paths of the least-cost tree meet the delay bounds required by the multicast nodes. To address potential delay bound violations, LDR identifies each node, v, whose delay exceeds its end-to-end delay bound, and replaces its current path with a new least-cost path that meets the end-to-end delay requirement of node v, if such a path exists. Notice, however, that adding the new path to the tree may cause nodes to have two incoming paths. This is undesirable as it causes the same information to be transmitted twice to the same node over different paths[4]. To illustrate this case, consider Figure 1.2, where S is a source node and M, R and L are multicast nodes, which depicts the least-cost tree produced by SPH. Assume, however, that in this tree the path from S to R, namely $((S,E), (E,H), (H,R))$ does not satisfy the end-to-end delay requirement of R. LDR finds a new least-cost path, from S to R, which meets the delay-bound requirement of node R. The addition of this path to the tree causes R to have two incoming paths. LDR removes the path with the higher delay up to the closest intermediate node of degree three or greater, as depicted in Figure 1.3 (Link (H,R) is removed in this case). The resulting tree satisfies the delay requirements of all multicast nodes. The addition of this new path to the tree, however, causes an intermediate node, N in this case, to have two incoming paths. This is again undesirable as node N receives the same information over two different paths. Furthermore, the presence of two incoming paths may unnecessarily increase the overall cost of the tree.

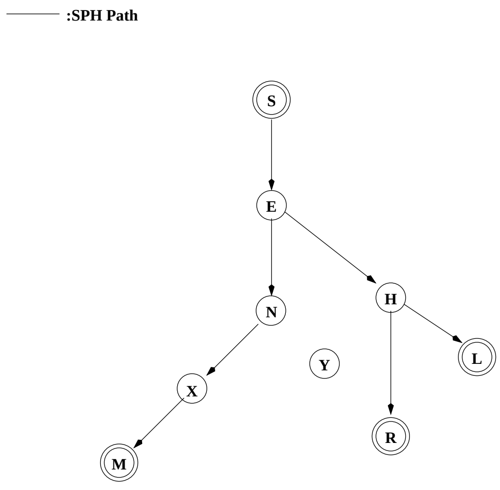


Figure 1.2 SPH low-cost multicast tree

In general, the problem of intermediate or multicast nodes with two incoming paths can be resolved by removing the *relay-path* on the larger-delay path. A relay-path is characterized by the following two properties: (i) all internal nodes

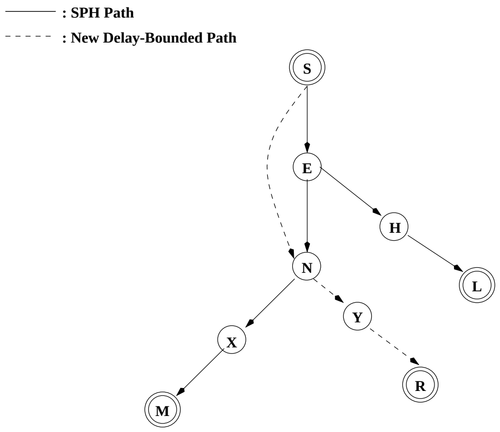


Figure 1.3 Resulting tree after removal of the path with larger delay

on the path are neither multicast nor source nodes and are exactly of degree two; (ii) the end nodes of the relay-path are either a source node, a multicast node or a node of degree three or higher[5]. In Figure 1.3, the relay-path, namely link (E,N), is redundant and may be removed. Notice, however, that because of potential wavelength conversions that can take place between the incoming link to node N on the new path and the outgoing links from node N along the paths to the multicast nodes attached to N, the removal of link (E,N) may not be feasible. For example, the removal of relay-path, (E,N), is only feasible if the end-to-end delay on the new path to multicast node M does not exceed M's delay bound

Based on the above, one possible way to verify the feasibility of removing a *redundant relay-path* at a given node, N, is by checking that its removal does not cause the end-to-end delay requirement of any multicast node attached to N to be violated. This, however, may increase the time complexity of the algorithm since it requires revisiting all nodes in the subtree attached to N. In the following, we derive a *sufficient* condition which can be used to verify the feasibility of removing a *redundant relay-path* without causing any multicast node to violate its end-to-end delay requirements. The condition involves quantities that can be obtained locally at the node and its verification can be achieved in a constant amount of time.

Relay-path removal feasibility condition. Consider an intermediate node, N, which has two incoming paths, namely $P(S,N)$ and $P'(S,N)$, and let $D_N = \{d \in D \mid \exists P(N,d)\}$ represent the set of multicast nodes of the subtree rooted at node N, as depicted in Figure 1.4. Let l_N, and l'_N represent the incoming link to node N on $P(S,N)$ and $P'(S,N)$, respectively. We use ν_N and ν'_N to denote the respective wavelengths associated with l_N and l'_N. Further, denote the wavelength associated with the outgoing link, l_R, at node N along $P(S,R)$ through node X as ν_R. Finally, let ν_d, $\forall\ d \in D$, denote the wavelength associated with the outgoing link, l_d, at node N along $P(N,d)$, to multicast node d.

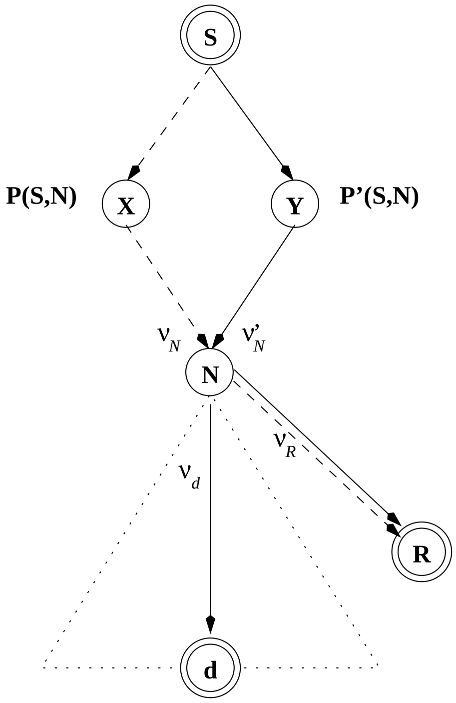


Figure 1.4 Redundant Relay Paths

Assume, without loss of generality, that $P(S,R)$ meets the delay requirements of node R, while the delay on $P'(S,R))$, denoted as $Delay(P'(S,R))$, exceeds the delay bound of node R. Then the following expression holds:

$$Delay(P(S,N)) + c_{l_R}(\nu_N, \nu_R) < Delay(P'(S,N)) + c_{l_R}(\nu'_N, \nu_R) \qquad (1.3)$$

which can be rewritten as:

$$Delay(P(S,N)) - Delay(P'(S,N)) < c_{l_R}(\nu'_N, \nu_R) - c_{l_R}(\nu_N, \nu_R) \qquad (1.4)$$

where $c_{l_R}(\nu_N, \nu_R)$ represents the conversion cost on link l_R along $P(S,R)$, and $c_{l_R}(\nu'_N, \nu_R)$ represent the conversion cost on link l_R along the concatenated path $P'(S,N) \cup P(N,R)$. Notice, however, that the relay-path on $P'(S,R))$ can be removed only if :

$$Delay(P(S,N)) + c_{l_d}(\nu_N, \nu_d) < Delay(P'(S,N)) + c_{l_d}(\nu'_N, \nu_d) \tag{1.5}$$

which can be rewritten as:

$$Delay(P(S,N)) - Delay(P'(S,N)) < c_{l_d}(\nu'_N, \nu_d) - c_{l_d}(\nu_N, \nu_d) \tag{1.6}$$

where $c_{l_d}(\nu_N, \nu_d)$ and $c_{l_d}(\nu'_N, \nu_d)$ represent the conversion cost on link l_d along $P(S,d)$, and $P'(S,d)$, respectively. Combining Equation 1.4 and Equation 1.6, a sufficient condition for a relay-path removal can be expressed as follows:

Relay-path $P'(S,N)$ *can be removed if:*

$$c_{l_R}(\nu'_N, \nu_R) - c_{l_R}(\nu_N, \nu_R) < c_{l_d}(\nu'_N, \nu_d) - c_{l_d}(\nu_N, \nu_d) \tag{1.7}$$

LDR heuristic description. Based on the above observations, the basic steps of the LDR heuristics can be described as follows:

Least Delay Replacement() {

Step 1 : Find the minimum cost multicast tree, T, using SPH.

Step 2 : Let $P_\delta(s,d)$ be the end-to-end delay on $Path(s,d)$ connecting node s to d on the tree. Find $W = \{\, d \in D, \;\mid\; P_\delta(s,d) \;>\; \Delta_d\}$.

Step 3 : For each node $v \in W$ {

Invoke the *Least Cost Constrained Path* procedure, $LCCP(\mathcal{A},\, T,\, v)$, to find the least-cost, delay-bounded path from s to v, whose addition to the tree and the removal of the resulting redundant relay-path do not cause any tree node to violate its delay bound.

If such a path exists, attach node v to the tree using this new path and remove the redundant relay-path to node v.

Else node v cannot be attached to tree. Exit.

}

Step 4 : Output T_i as the final minimal-cost, delay-bound multicast tree.

}

The $LCCP$ procedure searches for the least-cost, delay-bounded path that connects the delay violating node, $v \in D$, to the current tree, T. This is achieved by building a new graph G', obtained by reversing the links of G while keeping all its nodes. $LCCP$ then builds a set of paths composed of the

Least-Cost Constrained Path (G, T, v)

Create a new graph G' consisting of G's nodes and G's edges reversed.

Using G', create, P, the set of the least-cost delay paths from v to each node in T in G'.

Remove the edges of T from G'.

Using G', find the set of the least-cost paths from v to each node in T and add them to P.

Out of P, select path, p, such that:

- $delay(p) < \Delta_v$,
- $cost(p)$ is minimum, and
- addition of p to the tree only creates redundant relay-path (i.e., the removal of the redundant relay-path does not cause any other node in T to violate its delay requirements).

If no such path exists then return "no solution" else return p.

Figure 1.5 Least-Cost Constrained Path Procedure

shortest delay paths from v to the tree nodes and the shortest cost paths from v to nodes in T' in that order. Prior to computing the shortest cost paths, T's links are removed from G' to create different independent least-cost paths. This is necessary to avoid creating paths that are mutually derived from each other. Using this set of paths, $LCCP$ selects the path which meets the following three criteria (i) the end-to-end delay on the path meets the delay bound of the multicast node, and (ii) the addition of this path to the tree and the removal of the redundant relay-path associated with v do not cause any other multicast node currently attached to the tree to violate its delay bounds, and (iii) the path is the leat-cost path that meets criteria (i) and (ii). The basic steps of the $LCCP$ procedure are described in Figure 1.5.

The LDR heuristic uses the shortest delay paths, therefore, it finds a solution if one exists. Furthermore, if there is no violation, or the number of violations is low, then the cost of the resulting tree is close to the cost of the SPH tree. The complexity of the algorithm is bounded by the complexity of SPH which is $O(|Z| * n^2)$, where n is the total number of nodes in the graph. When there is a delay violation, then for each violating node the shortest delay path can be found in $O(n^2)$. In the worst case, we have $|D|$ violating nodes. Hence, the total time complexity of LDR is $O(|Z| * n^2)$.

Least Cost First (LCF)

The LCF heuristic incorporates the delay bound check as an integral part of the algorithm and performs a more elaborate search to find the least constrained path to any delay-bounded violating node. The algorithm builds the tree starting from the source node. Tree construction is performed by iteratively finding the node, $v \in D$, with the least-cost path. If the path to v satisfies the delay bound then that path is added to the tree. On the other hand, if the path fails to satisfy the delay bound, the Least-Cost Constrained Path ($LCCP$) procedure, described in Figure 1.5, is used to find the least-cost, delay-bounded path to v. The basic steps of the LCF heuristic can be described as follows:

Least Cost First () {

Let $PATH(d, T)$ represent the least-cost path from a multicast node $d \in D$ to a tree T.

Step 1 : Initially, set $i = 1$. Construct a subtree $T_1 = (s, \emptyset)$, consisting of the source node. Set $M_1 = \emptyset$.

Step 2 : Set $i = i + 1$. Find the closest node, d, to T_{i-1} such that $d \in (D - M_{i-1})$ (ties are broken by choosing the node with the smallest delay path).

Step 3 : If the delay from s to d on $PATH(T_{i-1}, d)$ does not exceed Δ_d, construct a new subtree, T_i, by adding all edges and nodes in $PATH(T_{i-1}, d)$. Set $M_i = M_{i-1} \cup \{d\}$. Go to Step 6.

Step 4 : If the delay on the least-cost path from s to d exceeds Δ_d then find a different path $PATH(T_{i-1}, d)$ using the $LCCP$ procedure described in Figure 1.5. Construct a new subtree, T_i, by adding all edges and nodes in $PATH(T_{i-1}, d)$. Set $M_i = M_{i-1} \cup \{d\}$.

Step 5 : Find the set, $\mathcal{V}$, of nodes, $v \in T_i$, such the addition of the new least cost path causes v to have two incoming paths. For each node $v \in \mathcal{V}$, remove the relay-path of the maximum delay path.

Step 6 : If $|M_i| < |D|$ then go to step 2 else DONE: T_i is the final least-cost, delay-bounded tree.

}

The time complexity of LCF is similar to LDR, which is dominated by the need to compute the least-cost path to all multicast nodes. Therefore the time complexity is $O(|Z| * n^2)$. LCF, however, does not take into consideration the delay when it tries to select the next node to add to the tree. Such an approach may unnecessarily increase the number of delay bound violations. To address this problem, we propose a different heuristic, referred to as Least Delay First, which takes the delay into consideration when selecting the next destination node to add to the tree. This heuristic is described next.

Least Delay First (LDF)

The algorithm is similar to LCF algorithm except that it uses a more efficient method to add a node to the tree. The algorithm first finds the least-cost paths to the destination nodes. Then out of these paths it selects the path whose delay is the minimum. This approach takes both constraints, cost and delay, into consideration when selecting the next node to add to the tree. The basic steps of the heuristic can be described as follows:

Least Delay First () {

Let $PATH(d, T)$ represent the least-cost path from a multicast node $d \in D$ to a tree T.

Step 1 : Initially, set $i = 1$. Construct a subtree $T_1 = (s, \emptyset)$, consisting of the source node. Set $M_1 = \emptyset$.

Step 2 : Set $i = i + 1$. Find the least-cost paths from T_{i-1} to the nodes in $D - M_{i-1}$. Select the node $d \in (D - M_{i-1})$ whose least-cost path has the minimum delay.

step 3 : If the delay from s to d on $PATH(T_{i-1}, d)$ does not exceeds Δ_d, construct a new subtree, T_i, by adding all edges and nodes in $PATH(T_{i-1}, d)$. Set $M_i = M_{i-1} \cup \{d\}$. Go to Step 6.

Step 4 : If the delay on the least-cost path from s to d exceeds Δ_d then find a different path $PATH(T_{i-1}, d)$ using the $LCCP$ procedure described in Figure 1.5. Construct a new subtree, T_i, by adding all edges and nodes in $PATH(T_{i-1}, d)$. Set $M_i = M_{i-1} \cup \{d\}$.

Step 5 : Find the set, $\mathcal{V}$, of nodes, $v \in T_i$, such the addition of the new least cost path causes v to have two incoming paths. For each node $v \in \mathcal{V}$, remove the relay-path of the maximum delay path.

Step 6 : If $|M_i| < |D|$ then go to step 2 else DONE: T_i is the final least-cost, delay-bounded tree.

}

The time complexity of LDF is similar to the time complexity of LCF. In this case, again the complexity is dominated by the need to compute the least-cost path to all multicast nodes. Therefore the time complexity is also $O(|Z| * n^2)$. Table 1.1 summarizes the time complexity of the heuristics discussed in this paper.

CONCLUSION

In recent years, WDM networks have been the subject of research and development. Dense WDM technology, which uses minimum spacing between channels, can accommodate up to one hundred optical channels per fiber without coherent detection techniques. Furthermore, the ability to use tunable transmitters

Heuristic	Time Complexity	Cost optimization approach
LDH	$O(n^2)$	No cost optimization
LCH	$O(n^2)$	Union of all shortest cost paths from source to destination
DPH	$O(max(\Delta_d) * n^3)$	Similar to SPH
BSMA	$O(k\, n^3\, log(n))$	Reducing cost by switching paths
LDR	$O(\lvert Z\rvert * n^2)$	Similar to SPH if no delay violation
LDF	$O(\lvert Z\rvert * n^2)$	Similar to SPH if no delay violation
LCF	$O(\lvert Z\rvert * n^2)$	Similar to SPH if no delay violation

Table 1.1 Time Complexity and bounds of the DBSMT heuristics

and receivers in WDM networks allows greater network flexibility as channels can be allocated dynamically according to traffic requirements. Both of these advantages make WDM an attractive technology to support the anticipated traffic of the next generation Internet. One specific aspect of future Internet services is multicasting.

High speed networking research efforts have concentrated on developing frameworks to support QoS guarantees. However, the success of these frameworks is strongly dependent on their ability to support routing and path establishment both for unicast and multicast communication. As the networks grow in size and user requirements increase in complexity the determination of the "best" route or multicast tree for the required level of QoS becomes a difficult task. Further, it is no longer sufficient to select routes with minimum cost; what is required is efficient mechanisms for building multicast trees that are not only of minimal cost but also support the application required level of QoS.

Given that WDM networks are fundamentally different from traditional networks when it comes to optimization objectives, it is imperative to explore the impact of the WDM networks' characteristics, in terms of wavelength availability, lightwave conversion cost and link delays, on the path selection process, both for unicast and multicast communication, and to understand the impact of the inaccuracy in the available network state information on the efficiency of the route establishment schemes. This paper addresses the first issue and proposes cost structure which takes into consideration both link cost and delay. Based on this cost structure, an optimization problem is formulated and three heuristics to this problem, which can be used to construct a low-cost, delay-bound multicast tree in WDM networks, are proposed.

The first heuristic, LDR, builds the least-cost tree using SPH, and then replaces the nodes whose delay were violated with the least-cost, delay-bounded path. If there is no delay violation, this heuristic gives the same approximation as SPH. The second heuristic, LCF, builds the multicast tree in a way similar to SPH. However, it incorporates the delay bound checking within the construction process of the minimum cost tree. When there is a delay violation, it uses $LCCP$ to find the least-cost bounded path. The last proposed heuristic, LDF, is similar to LCF. However, when it adds a node to the multicast trees, the heuristic selects the node whose least-cost path has the minimum delay.

Future work in this field may focus on adaptive path establishment in WDM networks. A good probabilistic model with up-to-date information about resource availability may select routes that can, with high probability, lead to the successful establishment of a multicast tree that satisfies the specified QoS requirements of the multicast nodes. However, when the rate of change in resource availability in the network is high, a distributed signaling protocol may fail to establish a connection with the specified QoS requirements along the selected route. Therefore, further research investigating multicast tree establishment protocols that can adapt to resource availability is required and different options for dynamically adapting to resource availability need to be developed. This includes allowing wavelength conversion when permitted by the QoS requirement, switching to overflow routes and selecting alternate routes dynamically.

Another aspect of multicasting that need to be addressed is on-line multicast and group management. The capability of on-line joining and leaving multicast groups is very important in today's multimedia systems. At the connection level, this requires the ability of individual nodes to set-up a connection to an already established point-to-multipoint connection. The choice of the path from the node to the tree should not only depend on the length of that path, but also on the availability of wavelengths and the number of wavelength conversions along that path. The problem is further complicated by the fact that, not only wavelength availability may dynamically change in the network, but also the multicast group membership may vary over time.

Notes

1. Current commercial WDM systems support up to 16 channels, at 2.5 Gbps per channel for an aggregate capacity of 40 Gbps per fiber, and are expected to scale to 32 channels at 10 Gbps per channel within the next few years.
2. The physical realization of wavelength conversion can be either opto-electronic or fully optical.
3. This bound only applies to graphs with symmetric, undirected edges. For directed graphs, the bound can be worse [23].
4. It may be acceptable to have a node with two incoming paths if the main objective is to find a path from the source to all multicast nodes. The final multicast graph, however, will not be a tree
5. Notice that according to the above definition, a relay-path can be composed only of one edge

References

[1] T. Ballardie, P. Francis, and J. Crowcroft. Core-based trees (cgt) an architecture for scalable inter-domain multicast routing. *Computer Com-*

muncation Review, 23(4):85–95, 1993.

[2] A. Barry and S. Subarmanian. The max_sum wavelength assignment algorithm for wdm ring networks. In Proceedings of OFC'97, February 1997.

[3] B. Beauquier, J.C. Gargano, S. Pereness, P. Hall, and U. Vaccaro. Graph problems arising from wave-length rounting in all. Proceedings of the 2nd Workshop on Optics and Computer Science (WOCS'97, 1997.

[4] I. Chlamtac, A. Faragó, and T. Zhang. Lightpath (wavelength) routing in large wdm networks. IEEE/ACM Journal on Selected Areas in Communications, 14:909–913, 1996.

[5] H. Choi, H.A. Choi, and M. Azizoglu. Efficient scheduling transmissions in optical broadcast networks. IEEE/ACM Trans. on Networking, December 1996.

[6] E. W. Dijkstra. A note on two problems in connection with graphs. *Numeriskche Mathematik*, 1:269–271, 1959.

[7] P. Dowd and K. Sivalingam. A multi-level wdm access protocol for an optically interconnectred parallel computer. Proceedings of the IEEE InfoCom, pages 400–408, 1994.

[8] T. Erlebach and K. Janson. Scheduling of virtual connections in fast networks. Proc. of the 4th Workshop on Parallel Systems and Algorithms (PASA'96), pages 13–32, 1996.

[9] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei. Protocol independent multicast-sparse mode (PIM-SM): Protocol specification. *Internet RFC 2117*, http://ds.internic.net/rfc/rfc2117.txt, June 1997.

[10] M. Feldman, S Esener, C. Guest, and S. Lee. Comparison between optical and electrical interconnects based on power and speed considerations. Applied Optics, 29:1742–1751, 1990.

[11] G. Gravenstreter, R.G. Melhem, D.M. Chiarulli, S.P. Levitan, and J.P. Teza. The partitioned optical passive stars (POPS) topology. IEEE Proceedings of the Ninth International Parallel Processing Symposium, April 1995.

[12] V. Kompella, J. Pasquale, and G. Polyzos. Multicast routing for multimedia communication. *IEEE/ACM Transactions on Networking*, 1(3):286–292, June 1993.

[13] L. Kou, G. Markowsky, and L. Berman. A fast algorithm for steiner trees. *Acta Informatica*, 15:141–145, 1981.

[14] M. Kovaceivc and A. S. Acampora. Benefits of wavelength translation in all-optical clear channel networks. IEEE Journal on Selected Areas in Communication, 14(5):868–880, June 1996.

[15] P. Lalwaney, L. Zenou, A. Ganz, and I. Koren. Optical interconnects for multiprocessors: Cost performance analysis. Proceedings of the Symposium on Frontiers of Massively Parallel Computation, pages 278–285, 1992.

[16] W. Liang and H. Shen. Efficient multiple multicast in wdm networks. Proceedings of the 1998 Intern. Conf. on Parallel and Distributed Processing Techniques and Applications, 1998.

[17] W. Liang and H. Shen. Multicast broadcasting in large wdm networks. Proceedings of the 12th Intern. Conf. on Parallel Processing Symposium (IPPS/SPDP), 1998.

[18] J. Moy. Multicast routing extensions for OSPF. *Communications of the ACM*, 37(8):61–66, August 1994.

[19] B. Mukherjee. WDM-Based local lightwave networks, Part I: Single-hop systems. IEEE Networks, pages 12–27, May 1992.

[20] Y. Ofek and B. Yener. Reliable concurrent multicast from busrty sources. Proceedings of IEEE Infocom'96, pages 1433–1441, 1996.

[21] P. Prucnal, I. Glesk, and J. Sokoloff. Demonstration of all-optical self-clocked demultiplexing of tdm data at 250 gb/s. Proceedings of the Int. Workshop on Massively Parallel Processing Using Optical Interconnections, pages 106–117, 1994.

[22] T. Pusateri. Distance vector multicast routing protocol. *Internet Draft*, February 1997.

[23] S. Ramanathan. Multicast tree generation in networks with asymmetric links. *IEEE/ACM Transactions on Networking*, 4(4):558–568, November 1996.

[24] V.J. Rayward-Smith and A. Clare. On finding steiner vertices. *Networks*, 16:283–294, 1986.

[25] Q. Sun and H. Langendoerfer. Efficient multicast routing for delay sensitive applications. *Proceedings of Second Workshop Protocols Multimedia Systems (PROMS'95)*, pages 452–458, April 1995.

[26] H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Mathematica Japonica*, 24:573–577, 1980.