# Efficient Fault-Tolerant Routing in Multihop Optical WDM Networks

Hong Shen, *Member, IEEE*, Francis Chin, *Fellow, IEEE*, and Yi Pan, *Senior Member, IEEE*

**Abstract**—This paper addresses the problem of efficient routing in unreliable multihop optical networks supported by Wavelength Division Multiplexing (WDM). We first define a new cost model for routing in (optical) WDM networks that is more general than the existing models. Our model takes into consideration not only the cost of wavelength access and conversion but also the delay for queuing signals arriving at different input channels that share the same output channel at the same node. We then propose a set of efficient algorithms in a reliable WDM network on the new cost model for each of the three most important communication patterns—multiple point-to-point routing, multicast, and multiple multicast. Finally, we show how to obtain a set of efficient algorithms in an unreliable WDM network with up to $f$ faulty optical channels and wavelength conversion gates. Our strategy is to first enhance the physical paths constructed by the algorithms for reliable networks to ensure success of fault-tolerant routing, and then to route among the enhanced paths to establish a set of fault-free physical routes to complete the corresponding routing request for each of the communication patterns.

**Index Terms**—Fault tolerance, multicast, point-to-point routing, queuing delay, WDM network

## 1 Introduction

Optical networks are emerging as a key technology in communication networks that provide capabilities far exceeding those of traditional electronic networks. The high bandwidth of fiber-optic cables enables data transmission rate in optic networks several orders of magnitude higher than electronic networks. The major applications of the network are video conferencing, scientific visualization, real-time medical imaging, supercomputing, and distributed computing [1], [29], [35]. A comprehensive overview of its physical theory and applications of this technology can be found in [13], [19], [24].

A popular approach to realizing optical networks is *Wavelength-Division Multiplexing* (WDM) [36]. WDM divides the optical spectrum in fiber-optic into many channels, each corresponding to a different optical wavelength, and thus allows multiple laser beams carrying different data streams to be transferred concurrently along a single fiber-optic link, provided that each beam uses a distinct wavelength. In a WDM network, all nodes (stations) are interconnected by point-to-point fiber-optic links, each supporting a certain number of wavelengths. At each node, a set of input ports receiving incoming data and output ports delivering outgoing data are attached to the optic links incident to the node. A wavelength coming in on an input port can be routed to one or more output ports. Multiple incoming signals cannot be routed on the same wavelength to the same output port at the same time.

WDM networks can in general be classified into two categories: *switchless* (also known as *nonreconfigurable*), in which the transmission from each node is broadcast to all nodes in the network and the desired signal is then extracted from all the signals at the receiver, and *switched* (also known as *reconfigurable*) in which signals on input ports at each node are routed to appropriate output ports directing to their destinations via a set of switches. Each of these networks can be further classified as either *single-hop* or *multihop*, according to whether wavelength conversion during the course of transmission is allowed or not. It has been shown that switchless networks suffer severe drawbacks and limitations, which make them difficult to use in wide area networks [1], [29]. We hereafter shall refer to optical networks always as switched networks. There have been numerous WDM network architectures and protocols proposed in the literature.

In single-hop networks, data transmission between each pair of source and destination uses the same wavelength throughout the whole communication path and, therefore, no wavelength conversion which requires conversion of optical signal to and from electronic form is needed. For this reason, single-hop networks are often referred to as *all-optical* networks. In single-hop networks, since there is no overhead due to optic-electronic conversion, signals are transmitted throughout the path in all optical form, which allows a very high data transmission rate in these networks. A communication path established between a source-destination pair in single-hop networks is called *lightpath* [8]. Single-hop (all-optical) networks are, in most cases, not practically feasible. Because optical networks are usually large in size—connecting at least hundreds of nodes—and the number of available wavelengths and tunability of

- *H. Shen is with the School of Computing and Information Technology, Griffith University, Nathan, QLD 4111, Australia. Email: hong@cit.gu.edu.au.*
- *F. Chin is with the Department of Computer Science and Information Systems, University of Hong Kong, Pokfulam Road, Hong Kong.*
- *Y. Pan is with the Department of Computer Science, University of Dayton, Dayton, OH 45469-2160.*

*For information on obtaining reprints of this article, please send e-mail to: tpds@computer.org, and reference IEEECS Log Number 110036.*

optical transceivers at each node are limited, it is very difficult, if not impossible, to establish a lightpath between each pair of communicating nodes in an optical network. Therefore, a more realistic approach is to allow wavelength conversion at some intermediate nodes on the communication path between a pair of nodes. At each of these intermediate nodes, the signal is converted from optic form to electronic form and then retransmitted on another wavelength (converted again from electronic to optic). This results in multihop networks. In multihop networks, a communication path between a source-destination pair is called *semilightpath* [7], which is obtained by establishing and chaining several lightpaths together, each undergoing an optic-electronic-optic conversion cycle.

As in electronic networks, a main research focus on WDM networks is the routing problem. Routing in a WDM network requires setting up a communication path between each pair of given nodes by chaining a set of optical channels together, with all channels on the path assigned a number of wavelengths (one wavelength) if the network is multihop (single-hop), and channels of different paths sharing the same optical link to have different wavelengths. In order to solve various application problems on a WDM network, mechanisms must be developed to handle not only point-to-point routing but also group (collective) communication involving transporting information from a group of nodes to another group of nodes in the network. A typical group communication is *multicast* that transports information from one source node to a set of destination nodes. A more general version of group communication is *multiple multicast* that contains more than one multicast group, each having its own source node and destination set [33]. If all the destination sets are identical, that is, messages from all sources are sent to the same set of destinations, the multiple multicast problem is also known as *concurrent multicast* [23], and *set-to-set broadcast* [18].

An important research topic on routing in WDM networks is to assign a minimal number of wavelengths for routing a given set of requests. This problem has been proven nontrivial even for the simplest network topologies like tree and ring [4], [10]. Another important topic of perhaps more practical significance is to set up minimum cost paths for fastest routing for a set of requests when the number of available wavelengths in a network is fixed and given by the network. There is a considerable body of work devoted to routing in single-hop (all-optical) networks [1], [2], [3], [6], [10], [28], [30], [31], particularly in single-hop tree-like networks [11], [14], [17], [25]. For routing in multihop networks, there has also been much work done recently [7], [20], [37]. Most of the above work is on point-to-point routing. On group communication, particularly multicast, research has become active recently [21], [22], [26], [27], [32], [34].

So far, most of the above work was done for reliable WDM networks. However, since WDM networks are usually larger and more complex than electronic networks, hardware faults may occur more frequently and in a greater variety than electronic networks. Two fundamental hardware faults are optical channel fault and wavelength conversion fault. Therefore, design of fault-tolerant routing mechanisms becomes not only theoretically important and challenging but also practically significant.

In this paper, we first propose a new cost model for routing in multihop WDM networks with limited wavelength conversion which is a general WDM network model and has already been adopted by various researchers in the literature [7], [20], [27], [37]. Our cost model takes the following three types of cost into consideration simultaneously: *channel cost* for using the assigned wavelength on the link, *wavelength conversion cost* for converting an incoming wavelength to a different outgoing wavelength at a sending node, and *queuing delay* for queuing multiple incoming signals sharing the same output channel (wavelength) at the same node. We then focus on the problem of efficient communication in an unreliable multihop WDM network in which wavelength channels may be faulty. We present efficient algorithms for the following major communication patterns in both reliable and unreliable WDM networks on this new cost model: multiple point-to-point routing, multicast, and multiple multicast.

Our proposed cost model is more general than existing models, since the latter are special cases of our model. Considering only the channel cost, routing is the same as that in electronic networks with each physical link being replaced with multiple optical channels and hence all existing routing algorithms for electronic networks can be applied. In general, routing algorithms for single-hop WDM networks belong to this category. Considering both channel cost and wavelength conversion cost, which is necessary for routing in multihop WDM networks, the routing problem is more complex than that for electronic networks because in addition to setting multiple channels for each link, a many-to-many crossbar-like switch must be included in each node to realize all wavelength conversions. Routing algorithms of [7], [20], [27], [37] are all along this line. Note that simply adding a node cost (degree) to the existing routing algorithms for electronic networks cannot make these algorithms directly applicable to multihop WDM networks, because converting different wavelengths at the same node may contribute to the routing cost with different costs. Moreover, in our model, allowing multiple incoming signals routed on the same wavelength to the same output port to share a common transmitter at a node through a queue makes a WDM network more scalable.

We would like to point out that queuing delay has been addressed implicitly in some recent work, for example [5], [12], mostly in the form of minimizing the degree at each node or maintaining the degree within the given constraint. To the best of our knowledge, we haven't seen any work reported on routing in WDM networks that considers all three cost factors. Moreover, most existing routing algorithms for WDM networks are not fault-tolerant in the sense that they cannot work correctly on an unreliable network. To our knowledge, little work has been reported on fault-tolerant routing in WDM networks.

Like most existing work on routing in WDM networks, our algorithms are *off-line* in the sense that all communication requests are prespecified and do not change during the course of communication. They use virtual *circuit switching* to establish all paths before data transmission actually takes

place, which uses *packet switching*. During data transmission, signal switching (scheduling) at each node ensures that each transmitter residing in the node is used to multiplex incoming signals from different channels to be relayed using their properly assigned output channel (wavelength).

## 2 The Optical Model and Cost Structure

Our optical model is the multihop WDM network with limited wavelength conversion, as stated in the previous section. For a network of $n$ nodes using $k$ wavelengths, a node $v$ contains a set of $n'$ input ports and $n''$ output ports. Each input port contains a dedicated electronic *receiver* with buffering capability for each incoming (signal) wavelength which converts the signal from optical to electronic, and each output port contains a dedicated laser *transmitter* for each outgoing wavelength that converts the signal from electronic to optical on that wavelength and transmits the optical signal. The traditional *converter* is composed of both a receiver at the input port and a transmitter at the output port. To handle the situation of simultaneous arrival of multiple incoming signals routed to the same optical channel (wavelength) in the same output port, we further assume that these signals are queued in a buffer (electronic) before they are sent to the corresponding transmitter at the output port one by one. There are $n''$ sets of buffers, where set $i$ is dedicated to output port $i$ and contains $o_i(v)$ buffers if there are $o_i(v)$ wavelengths available on output port $i$. Incoming signals are switched to the buffers through a cross-bar-like switch after the receivers. The switch is dynamically set according to the wavelength conversion cost from incoming wavelength to outgoing wavelength and the length of the queue on the outgoing wavelength for the purpose of minimizing the total waiting time of the signal at the output port. That is, signal with incoming wavelength $\lambda_i$ chooses outgoing wavelength $\lambda_j$ such that $\min_j\{c_v(\lambda_i,\lambda_j)+q_j(e)\}$, where $q_j(e)$ is the length of queue $j$ on channel (edge) $e$ in the output port. Fig. 1 shows the procedure of wavelength conversion at a node.

Our model uses a combination of *circuit switching* to reserve each communication path for each entire session when it is established, and *packet switching* to physically transmit the messages in each buffer along these paths. Clearly, the above model introduces a natural queuing delay that is the time required for an incoming signal before its wavelength conversion to wait for its turn in the buffer. The value of this queuing delay depends on the strategy used for scheduling signals. For simple round-robin scheduling, it is the length of the queue, which is the number of input ports on that node in the worst case.

Let $\Gamma=\{\lambda_1,\lambda_2,\ldots,\lambda_k\}$ be the set of available wavelengths in a WDM network. A WDM network can be represented by a directed graph $G=(V,E,\Gamma)$ with $|V|=n$ and $|E|=m$, where $\Gamma=\{\Gamma_{e_0},\Gamma_{e_1},\ldots,\Gamma_{e_{m-1}}\}$, $\Gamma_e\subseteq\Gamma$ is the set of wavelengths available at edge $e\in E$ with $w(e,\lambda)$ associated with wavelength $\lambda$ as the cost required to access $\lambda$. Converting a particular (incoming) wavelength ($\lambda_i$) to another (outgoing) wavelength ($\lambda_j$) at node $v$ causes a fixed cost $c_v(\lambda_i,\lambda_j)$ for all available $\lambda_j$ on all outgoing edges, where $c_v(\lambda_i,\lambda_i)=0$ indicates no wavelength conversion is incurred.

Let $\mathcal{P}$ be a semilightpath connecting a pair of nodes in the network to fulfill a request, and $C(\mathcal{P})$ be the total cost required for traveling path $\mathcal{P}$. Using the definition and notation of [7], $\mathcal{P}$ consists of a sequence of optical channels $e_1,e_2,\ldots,e_l$, where $e_i$ carries wavelength $\lambda_{p_i}$, $1\le p_i\le k$. All channels $e_1,e_2,\ldots,e_l$ are chained together such that the tail of $e_{i+1}$, $t(e_{i+1})$, coincides with the head of $e_i$, $h(e_i)$, for all $1\le i<l$. The following cost structure for $C(\mathcal{P})$ was defined in [7] and used in a number of papers such as [20], [21], [34].

$$C(\mathcal{P})=\sum_{i=1}^{l}w(e_i,\lambda_{p_i})+\sum_{i=1}^{l-1}c_{h(e_i)}(\lambda_{p_i},\lambda_{p_{i+1}}). \tag{1}$$

Due to the speed limitation of wavelength buffering, currently in electronic form, the signal queuing at output port described before may require a considerable amount of time, making the queuing delay a nonnegligible factor contributing to the total cost for transmission. We denote the queuing delay for transmitting any incoming signal using wavelength $\lambda$ on edge $e$ by $d_\lambda(e)$, which is proportional to the number of signals in the queue that buffers this signal, that is, the queue length. All signals in the same queue shall follow the same queuing delay because signals are transmitted in packet-switching along the optical channel of wavelength $\lambda$ on edge $e$. For the semilightpath $\mathcal{P}$ defined above, we have hence modified the above cost structure for $C(\mathcal{P})$ as follows:

$$C(\mathcal{P})=\sum_{i=1}^{l}w(e_i,\lambda_{p_i})+\sum_{i=1}^{l-1}(c_{h(e_i)}(\lambda_{p_i},\lambda_{p_{i+1}})+d_{\lambda_{p_{i+1}}}(e_i)). \tag{2}$$

In the case of multicast, we want to transport data from source node $s$ to a set of destination nodes $D=\{t_1,t_2,\ldots,t_g\}$. Multicast can be performed by first constructing a *multicast tree* $MT$ that is rooted at $s$ and connects all nodes in $D$, and then traveling all tree edges. Assume that $\{e_1,e_2,\ldots,e_{|MT|}\}$ is the sequence of edges obtained by left-first traversal (left-visit-right) on $MT$ that enumerates semilightpaths in $MT$, and $L$ is the set of leaf nodes in $MT$. The cost for traversing $MT$ is:

$$C(MT)=\sum_{i=1}^{|MT|}w(e_i,\lambda_{p_i})+\sum_{1\le i<|MT|,h(e_i)\notin L}(c_{h(e_i)}(\lambda_{p_i},\lambda_{p_{i+1}})+d_{\lambda_{p_{i+1}}}(e_i)). \tag{3}$$

In the above setting, $G=(V,E,\Gamma)$, modeling our WDM network carries three different types of weight (cost) which need to be combined dynamically during the course of any dedicated routing. This makes derivation of efficient routing algorithms directly on $G$ difficult and infeasible. Hence, it becomes necessary to devise effective ways to transform graph $G$ into another graph to support routing. In the case of only considering edge weight and conversion cost, there are different ways to transform graph $G$, for example those in [7], [20], [21]. But most of them are unnecessarily complicated, and hence, not suitable for direct use for our general optical model. We propose the following simple method to transform $G$.

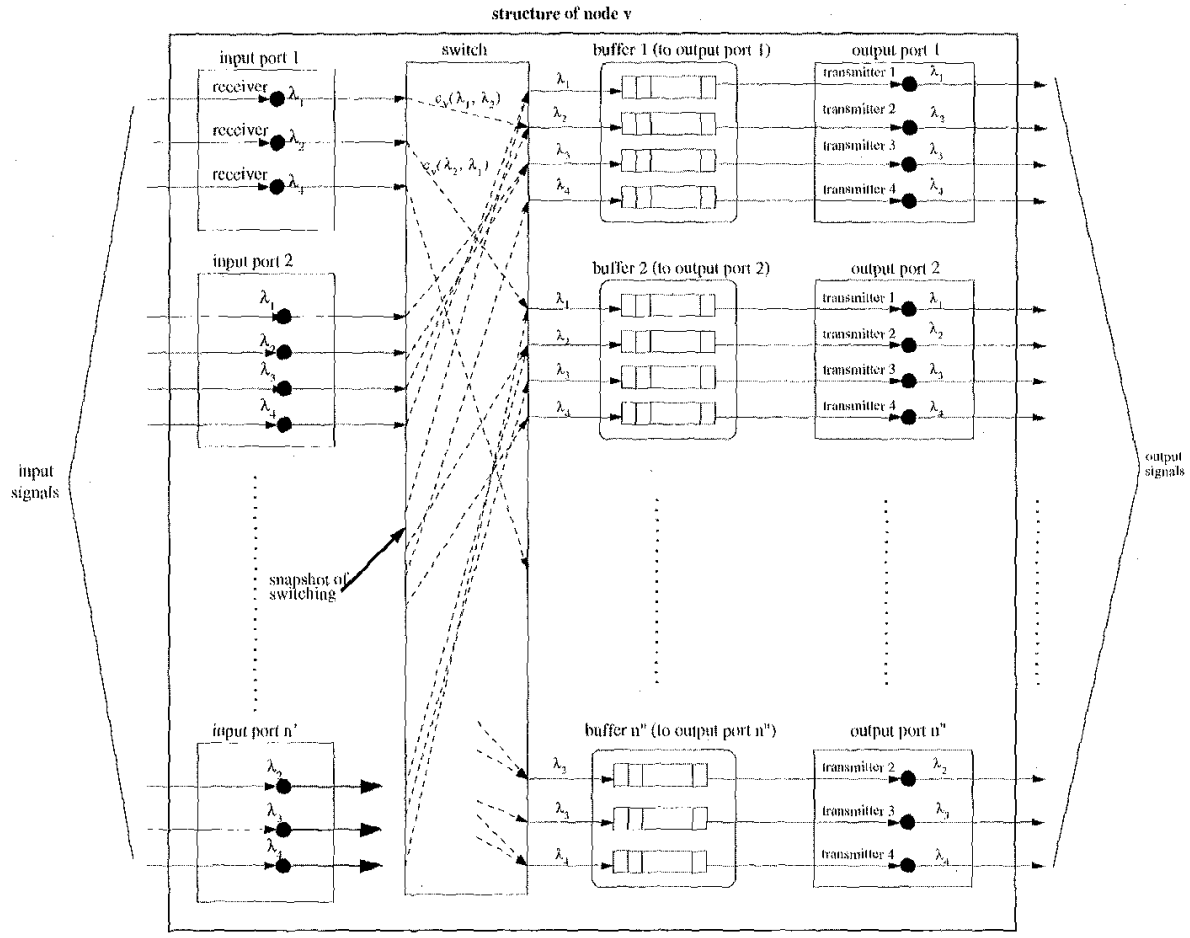



Fig. 1. Wavelength conversion at a node.

Let $\delta[1..k](e)$ represent the queue I/O delays (time) required for queuing all incoming signals on link $e$, where $\delta[i](e)$ is the queue I/O delay for incoming signals using output wavelength $\lambda_i$ on $e$, which is mainly defined by the speed of the underlying buffer of the queue. Here we consider the general case that different incoming wavelengths may have different queuing delays subject to the length of the queue and speed of the buffer for each wavelength.

We transform $G = (V, E, \Gamma)$ into an auxiliary graph $G_M = (V_M, E_M)$ as follows. We call all original nodes in $V$ *node*, all auxiliary nodes in $G_M$ *vertex*, optical channels on all links in $E$ and auxiliary edges in $G_M$ *edge*. $G_M$ is a directed and weighted graph with both fixed edge weights and dynamically changing edge weights with initial value zero.

1. For each $v \in V$, construct a bipartite graph $G_v = (A_v \cup B_v, E_v)$, where vertex sets $A_v$ and $B_v$ represent the input wavelengths and output wavelengths at $v \in V$, and $E_v$ represents all possible wavelength conversions at $v$—$(a \in A_v, b \in B_v) \in E_v$ iff wavelength $a$ can be converted to wavelength $b$ at $v$ (i.e., $c_v(a, b)$ exists). Set $c_v(a, b) = 0$ if $a = b$. Assign weight $c_v(a, b)$ to edge $(a, b)$. Connect all vertices in $A_v$ to $v$ through introducing $k$ new edges $E'_v$. Assign zero weight to each of these new edges. Vertices in $B_v$ are connected to the appropriate nodes in $V$ by edges transformed from links in $E$ described in the following step.
2. Replace each $e \in E$ with $|\Gamma_e| \leq k$ parallel edges (channels), $E_e$. For each $e' \in E_e$ carrying wavelength $\lambda_i$, assign edge weight $w(e, \lambda_i)$ to it. These edges connect vertices in $B_v$ to the corresponding vertices in $A_{v'}$ of $v$'s neighbor(s) $v'$.
3. Assign an edge weight $d_{\lambda_i}(e')$, initially zero, to edge $e'$ (representing outgoing wavelength $\lambda_i$) in $E_e$, indicating the queuing delay for sending messages from $h(e') \in B_v$ to $t(e') \in A'_v$ using wavelength $\lambda_i$. This edge weight is dynamically changing—it is increased by a queuing delay $\delta[i](e)$ when an incoming signal arrives at the queue for this wavelength.
4. Let $V_M = \cup_{v \in V} A_v \cup B_v$, and

$$E_M = (\cup_{v \in V}(E_v \cup E'_v)) \bigcup (\cup_{e \in E} E_e).$$

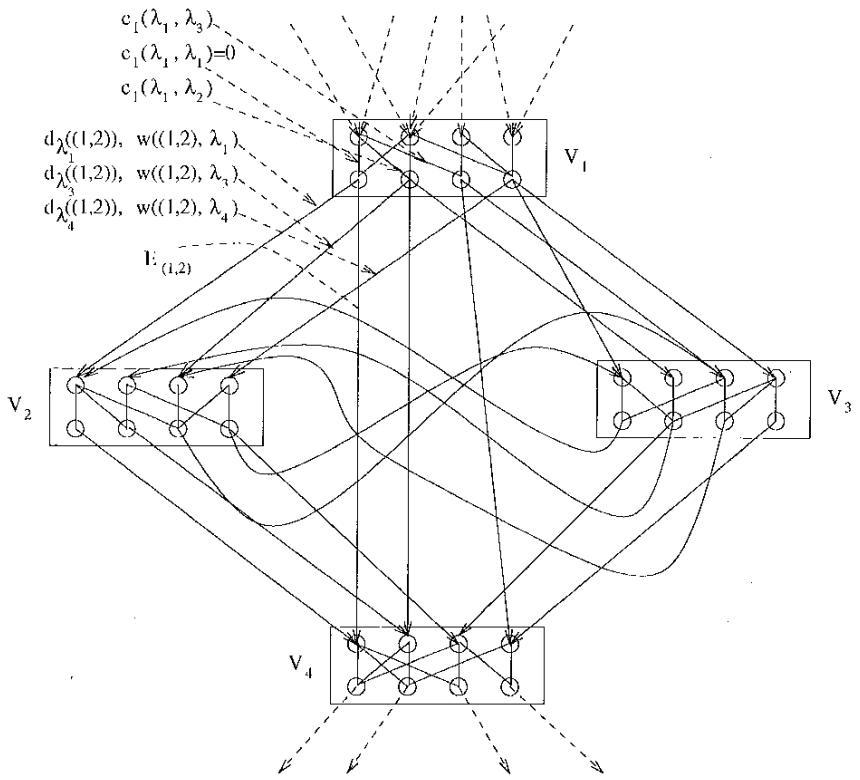



Fig. 2. An example of a component of $n = 4$, $m = 7$, and $k = 4$ in $G_M$.

Since $|A_v| = |B_v| = k$, $|E_v| \leq k^2$ and $|E_e| \leq k$, we can easily obtain the following equations:

$$|V_M| \leq 2kn, \tag{4}$$

$$|E_M| \leq k^2 n + km. \tag{5}$$

An example of $G_M$ is given in Fig. 2.

Because for each $v \in V$ all edges in $E'_v$ have been assigned a zero weight, the relative distance between any pair of vertices in $G$ remains unchanged in $G_M$. The relativity shall depend on the setting of edge weights in $G_M$, which increases proportionally as the the number of incoming signals to the same queue increases. Therefore, routing between any pair of vertices in $G$ corresponds to routing between them in $G_M$. In other words, $G_M$ and $G$ are equivalent for all routing problems. In the following sections, all our WDM networks will be modeled by $G_M$. Whenever appropriate, we use *weight* and *cost* interchangeably.

For general $G_M$, which is a directed graph, we define the (edge) *connectivity* of $G_M$ to be the minimal number of edge-disjoint directed paths from any node to any other node in $G_M$. We equivalently say that $G_M$ is $t$-edge connected if $G_M$ has a connectivity of $t$.

## 3 Routing in Reliable Networks

In this section, we study routing in a reliable WDM network. We present a set of efficient algorithms for point-to-point requests, multicast, and multiple multicast, respectively, on the proposed cost model of equations.

### 3.1 Multiple Point-to-Point Routing

Point-to-point routing is the most fundamental routing problem. A general form of point-to-point routing is routing for multiple requests: given $r$ requests $\mathcal{R} = \{(s_1, t_1), (s_2, t_2), \ldots, (s_r, t_r)\}$, we are required to establish a communication path from $s_i$ to $t_i$ for each request $(s_i, t_i)$, where $1 \leq i \leq r$ and $r$ is smaller than the connectivity of $G_M$. We call this problem *multiple point-to-point routing*. Being able to route for multiple requests is an essential ability of a communication network.

Let $SP_i$ be the shortest path for request $(s_i, t_i)$ in $G_M$ without considering the queuing delay caused by the existence of $SP_j$, $1 \leq i \neq j \leq r$. When queuing delay is taken into account, an optimal solution producing minimum total cost of all the paths needs to compute the alternative shortest paths of $SP_1, SP_2, \ldots, SP_r$ in $G_M$ without passing through any edge in $E(SP_{i_1}) \cap E(SP_{i_2}) \cap \ldots \cap E(SP_{i_q})$ for all different combinations of $1 \leq i_1 \neq i_2 \neq \ldots \neq i_q \leq r$ and $2 \leq q \leq r$, and then take those to replace their corresponding $SP$'s if such replacement results in the minimum total cost of all the $r$ paths. Clearly, the total number of all these alternative paths is prohibitively large for large $r$, so it is unrealistic to expect an optimal solution.

We use a greedy approach to design our algorithm that employs a modified shortest-path algorithm to construct paths one by one in their length increasing order, where the length of a path is the number of edges on the path. During each step of including an edge to a path, the weight of the edge is increased by an appropriate additive factor of $\delta$ to reflect the increment of queuing delay for all signals in the same queue when being transmitted.

Let the index of the $i$th path in length increasing order be $\pi_i$. For any edge $e$ on path $P$ under construction in $G_M$, we call the probability of $e$ falling on an occupied edge by other paths the *edge overlapping probability* of $e$. We define the *path overlapping probability* of $P$ to be the average edge overlapping that equals the summed edge overlapping of all edges on $P$ divided by the length of $P$. Path overlapping probability shows how likely edges on the path are to fall on occupied edges in $G_M$, and consequently gives an indication on the quality of the path that is the ratio of its total cost versus its optimal cost. We shall show that the path overlapping probability of $SP_1, SP_2, \ldots, SP_r$ is minimized if we establish paths in the order of $SP_{\pi_1}, SP_{\pi_2}, \ldots, SP_{\pi_r}$, and hence, the proposed heuristic is optimal in this regard in the expected case when every edge in $G_M$ has an equal probability of being used by all shortest paths.

Once all paths are established, signals are sent along the paths using packet switching, observing the delays calculated in path establishment. Our modified shortest path algorithm for path establishment takes into consideration dynamic edge weights for edges on the path under construction and uses *source routing* in which path establishment is initiated by the source node and path is extended step-by-step from one node to the next toward the destination. Source routing can be viewed as a limited version of centralized routing without global central control. Our multiple point-to-point routing algorithm has the following structure:

Algorithm **A.1**
{*Construct a shortest path for each request $(s_i, t_i)$, $1 \le i \le r$.*}
**for** $i := 1$ to $r$ do
  Use Dijkstra's algorithm to compute the length $L_i$ of the shortest path from $s_i$ to $t_i$;
  Insert $L_i$ into the sorted list of increasing order $\{L_{\pi_1}, L_{\pi_2}, \ldots, L_{\pi_i}\}$;
**for** $i := 1$ to $r$ do
  Find the shortest path from $s_{\pi_i}$ to $t_{\pi_i}$ using source routing, where for any edge (channel) $e$ using wavelength $\lambda_j$ the distance from $h(e)$ to $t(e)$ is $w(e, \lambda_j) + c_{h(e)}(\lambda'_j, \lambda_j) + d_{\lambda_j}(e)$;
{*$\lambda'_j$ is the wavelength on the preceding edge of the path from which $\lambda_j$ is converted.*}
  Add $\delta[j](e)$ to $d_{\lambda_j}(e)$ for each $e$ of wavelength $\lambda_j$ on the path constructed;
{*Increase the queuing delay of all signals in the same queue by a pre-specified I/O cost, where all queuing delays are initialized to 0.*}

**Lemma 1.** *When $SP_{\pi_1}, SP_{\pi_2}, \ldots, SP_{\pi_r}$ are constructed one-by-one in this order, the path overlapping probability of $SP_{\pi_i}$ at which $SP_{\pi_i}$ passes through any edge of $SP_{\pi_1}, SP_{\pi_2}, \ldots, SP_{\pi_{i-1}}$ is minimized if $L_{\pi_j} \le L_{\pi_i}$ in the expected case when each edge in $G_M$ has an equal probability to be used by all paths.*

**Proof.** If $SP_i$ is constructed after $SP_j$, the path overlapping probability of $SP_i$ that passes through edges of $SP_j$ is $\frac{L_j}{|E_M|}$. Assume that swapping the order of $SP_i$ and $SP_x$ for any $x < i$, that is, constructing paths one-by-one in the order of $SP_{\pi_1}, \ldots, SP_{\pi_{x-1}}, SP_{\pi_i}, SP_{\pi_{x+1}}, \ldots, SP_{\pi_x}$, will result in the minimal path overlapping probability for these paths. Since $\frac{L_j}{|E_M|} \le \frac{L_i}{|E_M|}$ for any $L_j \le L_i$, swapping back $SP_i$ and $SP_x$ will have a smaller path overlapping probability for each of the paths $SP_{\pi_{x+1}}, \ldots, SP_{\pi_i}$. Therefore, the assumption is false, and the minimal path overlapping probability must result in the order stated in the lemma. □

From the above lemma, it is clear that the heuristic used in Algorithm **A.1** is optimal with respect to the path overlapping probability for all paths, which determines the quality of the paths, in the expected case when each edge in $G_M$ has a equal probability to be used by all paths.

In Algorithm **A.1**, the shortest path from $s_{\pi_i}$ to $t_{\pi_i}$, when all paths from $s_{\pi_j}$ to $t_{\pi_j}$ have been established for all $j < i$, is constructed using a modified Dijkstra's algorithm [9] with the change in distance formula including dynamic edge weight, as previously stated. At the first step of routing, source ($s_{\pi_i}$) will pick up a channel (wavelength) to send the signal, so there is only channel cost at the source node, no wavelength conversion. Wavelength conversion may occur in all subsequent steps. Using adjacent list to represent $G_M$, Dijkstra's algorithms can be implemented in $O(|E_M| + |V_M| \log |V_M|)$ time. By (4) and (5), we have the following theorem:

**Theorem 1.** *The problem of multiple point-to-point routing for $r$ requests in a WDM network of $n$ nodes and $m$ links with $k$ available wavelengths can be solved in $O(rk(kn + m + n \log(kn)))$ time.*

### 3.2 Multicast

Multicast requires transporting information from source $s$ to a set of destinations $D = \{t_1, t_2, \cdots, t_g\}$. Multicast can be realized by first constructing a multicast tree $MT$ rooted at $s$, including all nodes $\{t_1, t_2, \cdots, t_g\}$ in $G$, and then transmitting information from the root to all destinations along the tree edges using appropriate wavelengths. We are interested in finding an optimal $MT$ in which the total cost for multicast is minimum. Putting the dynamic edge weight aside, when $G$ is transformed into $G_M$, it is clear that finding an optimal $MT$ is equivalent to finding a minimum directed Steiner tree in $G_M$ which is unfortunately NP-complete. Our generalized cost in (3) requires considering both static and dynamic edge weights in calculating the cost, which is even harder than the minimum directed Steiner tree problem on $G_M$. We therefore have to turn to approximate solutions that produce an $MT$, whose cost is close to that of an optimal $MT$.

We use an approach based on that of [16] to construct a *minimum spanning tree* (MST) rooted at $s$ on an induced graph $I(\{s\} \cup D)$ of $G_M$ to approximate the Steiner tree in $G_M$, where $I(\{s\} \cup D)$ is a complete graph on vertex set $\{s\} \cup D$ with the distance (length of shortest path) being the edge weight between any pair of vertices. Note that the algorithm of [16] works only for undirected graphs, and may not work for general directed graphs, as $I(\{s\} \cup D)$ may not exist for general directed graphs. However, due to the special properties of $G_M$, we can make this approach also work for our case. Since our graph $G_M$ is a special directed graph in which there is a path between any pair of

nodes $v_i, v_j \in V$ (not those in $A_v$ and $B_v$), the induced graph $I(\{s\} \cup D)$ exists and is a completely directed graph. We can therefore apply the approach of [16] for approximate multicast in our case as follows.

We construct $I(\{s\} \cup D)$ that is a completely directed graph with vertex set $\{s\} \cup D$ and edge weight $dist(u, v)$ in $G_M$ for all $u, v \in \{s\} \cup D$. After we have constructed $I(\{s\} \cup D)$, we find the directed MST rooted at $s$ instead of the undirected $MST$ in the undirected case [16]. Because we are seeking for a Steiner tree in $G_M$ rooted at $s$ covering $D$, we can use the approximation ratio of the undirected MST on $I(\{s\} \cup D)$ to the undirected Steiner tree on $\{s\} \cup D$ to estimate that of the directed $MST$ rooted at $s$ to the directed Steiner tree on $\{s\} \cup D$.

The directed $MST$ rooted at $s$ in $I(\{s\} \cup D)$ can be constructed as follows: Extend a most economic path from $s$ to every node in $D$ one by one, where the most economic path adds a least weight edge to the MST under construction. It expands the $MST$ from originally only containing $s$ to finally covering all nodes in $D$ by repeatedly adding an edge of direction outwards of the $MST$ with the least weight in the neighborhood of the $MST$. This construction can be completed in $O(|I(\{s\} \cup D)|^2) = O(g^2)$ time, because each step needs to consider at most $g$ neighbors.

**Lemma 2.** *The MST constructed by the above method is the minimum cost directed spanning tree that connects $s$ to every other node in $I(\{s\} \cup D)$.*

**Proof.** Assume that the MST constructed is not the minimum cost directed spanning tree rooted at $s$. Then there must exist at least one nontree edge $e^* \in E(I(\{s\} \cup D) - MST)$ whose replacement to a tree edge $e \in E(MST)$ will result in another directed spanning tree rooted at $s$, $MST^*$, that has a smaller cost than $MST$. Since $MST^*$ and $MST$ both connect $s$ to every other node in $I(\{s\} \cup D)$, $e^*$ and $e$ must point to the same node $v$. When the above method extends MST to include $v$, it would thus have chosen $e^*$ instead of $e$ to connect MST to $v$, since $w(e^*) < w(e)$. This contradicts the fact that $e \in E(MST)$ and $e^* \in E(I(\{s\} \cup D) - MST)$. Therefore, the assumption is false and hence, the lemma is true. □

With the help of the above algorithm, we can now present our algorithm for multicast in the WDM network. Let $dist(u, v)$ be the shortest distance from $u$ to $v$ in $G_M$ that is the summed edge weight on the shortest path from $u$ to $v$. We also keep the shortest path corresponding to $dist(u, v)$ in $P[u, v]$ accordingly. The induced graph $I(\{s\} \cup D)$ is a complete graph on $g + 1$ nodes $(\{s\} \cup D)$ with cost $dist(u, v)$ associated with edge $(u, v)$. The algorithm works as follows:

Algorithm **B.1**
{*Multicast routing for $\mathcal{M} = (s, D)$, where $D = \{t_1, t_2, \ldots, t_g\}$.*}

1. **for** each ordered pair of $u, v \in \{s, t_1, t_2, \cdots, t_g\}$ **do**
   Compute the shortest path from $u$ to $v$, $P[u \rightarrow v]$, and $dist(u, v)$ in $G_M$ using modified Dijkstra's algorithm to include dynamic edge weight updating as used for point-to-point source routing;
   For each $e \in P[u \rightarrow v]$ add $\delta[j](t(e))$ to $d_{\lambda_j}(e)$ if $e$ uses wavelength $\lambda_j$;
2. Construct $I(\{s\} \cup D)$;
3. Compute the $MST_I$ rooted at $s$ in $I(\{s\} \cup D)$ using the algorithm described before;
4. Replace each edge in $MST_I$ with the corresponding path in $G_M$, that is, $dist(u, v)$ with $P[u \rightarrow v]$, and break all cycles at their maximum weighted edges (removal) so that the resulting subgraph is a Steiner tree $ST$;
5. For each edge $e$ of wavelength $\lambda_j$ in $ST$, add $\delta[j](e)$ to $d_{\lambda_j}(e)$.
   {*Increase the queuing delay of all signals in the same queue by a pre-specified I/O cost.*}

For completely directed graph $G_I = I(\{s\} \cup D)$, let $ST$ be the Steiner tree constructed in $G_M$, $MST$ be the minimum spanning trees constructed by Algorithm **B.1**, both rooted at $s$. We have the following lemma showing their approximation ratio.

**Lemma 3.** *Approximating $ST$ on $g$ destination nodes by $MST$ using Algorithm **B.1** has the following approximation ratio in the expected case when all edge weights in $G_M$ are randomly and independently chosen from a fixed range. That is,*

$$\frac{Cost(MST)}{Cost(ST)} \leq 2 - \frac{2}{g+1}. \tag{6}$$

**Proof.** Using a similar proof method as [16] for undirected case, we construct a cycle $C$ that starts from and ends at $s$ and visits all nodes in $D$ once in the order as they are visited in the Steiner tree $ST$, but goes from one to the next on the shortest path between them in $G_M$, as shown in Fig. 3. When we embed $C$ in $I(\{s\} \cup D)$, clearly $C$ "traverses" each edge of $ST$ twice, where the traversal of opposite direction to the edge direction is achieved along a shortest path in $G_M$ that connects the tail to the head of the edge. Hence, $C$ corresponds to two trees, $ST$ itself and an "inverted" $ST$, $ST'$, that replaces each edge $(u, v)$ of $ST$ with an opposite direction shortest path from $v$ to $u$, as depicted in Fig. 3. Cutting off the most costly edge in $C$ makes $C$ become a spanning tree $T$ in $I(\{s\} \cup D)$. Let $Cost(ST)$ and $Cost(ST')$ be their accumulated edge weights. Since the most costly edge in $C$ carries at least weight $\frac{1}{g+1}Cost(C)$,

$$Cost(MST) \leq Cost(ST) \leq (1 - \frac{1}{g+1})Cost(C).$$

Since $ST$ and $ST'$ are directed, depending on the total edge weight difference of $ST$ and $ST'$, $Cost(ST)$ and $Cost(ST')$ may differ within range $(-\Delta, \Delta)$, where $\Delta$ is the total weight of the longest path in $G_M$. As all edge weights are randomly and independently chosen from a fixed range, values of the above difference are uniformly distributed over the range $(-\Delta, \Delta)$. Thus, the expected value of the difference of $Cost(ST)$ and $Cost(ST')$ is 0. That is, in the expected case $Cost(ST) = Cost(ST')$ and hence, $Cost(C) = 2Cost(ST)$. Removing the most costly edge from $C$ results in a spanning tree whose total cost is at least as that of the $MST$. Therefore, (6) holds in the expected case when all edge weights in $G_M$are randomly and independently chosen from a fixed range. □

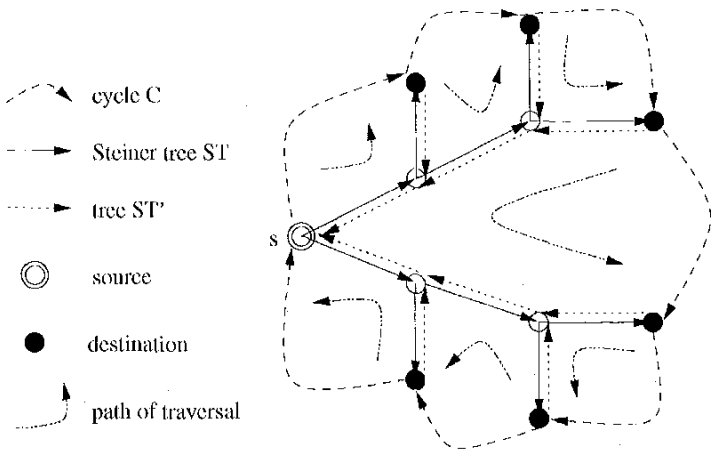



Fig. 3. A cycle connecting all destinations in a Steiner tree.

From Lemma 3, we know that the cost of $MST_I$ obtained in Algorithm **B.1** is no more than that of $2-\frac{2}{g+1}$ times of that of the desired Steiner tree in the expected case. That is, this $MST_I$ is $(2-\frac{2}{g})$-OPT (within factor $(2-\frac{2}{g})$ of the optimal cost).

Since $|V_M| = 2kn$ and $|E_M| = k^2n + kn$ by (4) and (5), Step 1 of the algorithm requires $O(g^2k^2n + g^2km + g^2kn\log(kn))$ time. Steps 2 and 3 can be done in $O(g^2)$ time. Step 4 requires $O(gkn)$ time. Therefore, we have the following theorem:

**Theorem 2.** *A $(2-\frac{2}{g})$-OPT approximate multicast tree for multicast of group size $g$ in a WDM network of $n$ nodes and $m$ links can be computed in $O(g^2k(kn + m + n\log(kn)))$ time in the expected case, where $k$ is the number of available wavelengths in the network.*

Note that after Step 4 replacement of each $dist(u,v)$ with its corresponding path $P[u \to v]$, $MST_I$ may contain $|V_M|$ nodes, because all these shortest paths may span over the entire $G_M$.

### 3.3 Multiple Multicast

When several groups of multicast wish to take place concurrently, a more general communication pattern multiple multicast is formed. Given $r$ groups of multicast $\mathcal{M}_i = (s_i, D_i)$, where $s_i$ is a source and $D_i = \{t_i^1, \ldots, t_i^{g_i}\}$ are the destinations, $1 \le i \le r$ and $r$ is smaller than the connectivity of $G_M$, assume that $\mathcal{M}_i$ alone (without considering the existence of other groups) can be realized by a multicast tree $MT_i$. Let multicast forest $MF = \cup MT_i$. It is clear that several edges of different $MT_i$ in $MF$ may fall onto the same edge of $G_M$ and hence, attempt to use the same wavelength at the same node in the network. This will possibly cause *contention* on a particular wavelength when these requests arrive simultaneously at a node. Fig. 4 shows an example of wavelength contention caused by three multicast trees.

As our WDM network model does not allow multiple requests using the same wavelength on the same physical link due to the current technology restriction, an important task in implementing multiple multicast is to construct a minimum cost $MF$, which is wavelength contention-free. To achieve this, we take a greedy approach to find an approximate optimal multicast tree for each multicast $MT_i$ one-by-one employing Algorithm **B.1**, where all edges of $MT_i$are marked with an infinitely large weight as soon as $MT_i$ is constructed, and before constructing the next tree so as to ensure that these edges will not be chosen again by the later trees and hence, no wavelength contention can possibly occur. Note that there can be $r!$ ways of ordering all $MT_i$'s, each possibly resulting in different total cost for the final $MF$. We take the heuristic of ordering them by size, which is the number of participating nodes in the multicast group, for the reason that the smaller size an $MT_i$ has, the more flexible alternations it can undergo. Our algorithm for multiple multicast is described as follows:

Algorithm **C.1**
{*Multiple multicast for $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_r$, where $\mathcal{M}_i = (s_i, D_i)$.*}

1. Sort $\{\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_r\}$ into increasing size order $\{\mathcal{M}_{\pi_1}, \mathcal{M}_{\pi_2}, \ldots, \mathcal{M}_{\pi_r}\}$.
2. **for** $i = 1$ to $r$ do
   Construct multicast tree $MT_{\pi_i}$ for $\mathcal{M}_i$ using Algorithm**B.1**.
   For each $e \in E(MT_i)$ mark $w(e)$ with weight $\infty$.

The correctness of the algorithm is seen clearly from the greedy approach. The time complexity of the algorithm is $O(r\log r + \sum_{i=1}^{r} t_{MT_i})$, where $t_{MT_i}$ is the time complexity required for constructing the multicast tree for $\mathcal{M}_i$. With our result for multicast in the previous section, we have the following theorem:

**Theorem 3.** *The problem of multiple multicast for $r$ groups of sizes $g_1, g_2, \ldots, g_r$, respectively, in a WDM network can be solved in $O((\sum_{i=1}^{r} g_i^2)k(kn + m + n\log(kn)))$ time, where $n$, $m$, and $k$ are the number of nodes, links, and available wavelengths in the network, respectively.*

The probability of edges of $MT_{\pi_j}$ falling to those of $MT_{\pi_i}$ is the probability of wavelength contention and hence, queuing delay increase caused by $MT_{\pi_i}$ and $MT_{\pi_j}$ both wanting to access the wavelength represented by this edge (channel). As proved in Lemma 3.1, in the expected case

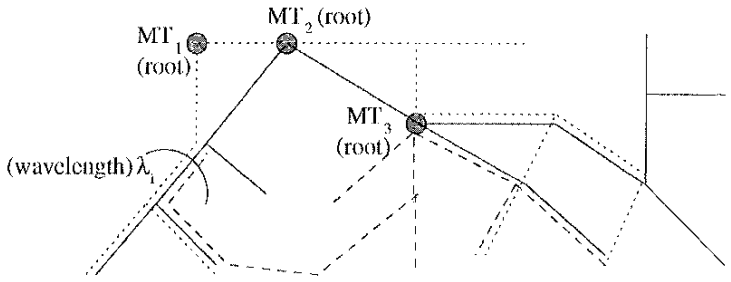



Fig. 4. Wavelength contention caused by three multicast trees.

when all edges in $G_M$ have an equal probability to be used by all multicast trees, our heuristic is optimal in the sense that it minimizes the average probability of edges of $MT_{\pi_j}$ falling to those of $MT_{\pi_i}$ for $j > i$. Therefore, Algorithm **C.1** has the same approximation ratio as Algorithm **B.1** in the above expected case.

## 4 Routing in Unreliable Networks

In this section, we consider the routing problem in an unreliable WDM network in which both optical channel (wavelength) and wavelength conversion faults may occur. The optical channel fault occurs in the cases such as the designated wavelength on the channel is accidentally lost, distorted, and insufficiently amplified. The wavelength conversion fault occurs when the corresponding wavelength conversion within a node cannot be completed correctly due to hardware fault in the receiver or switch. We call the faulty hardware blocking a single wavelength conversion a *faulty gate*. From our optical model described in Section 2, we know that a channel of any link and a wavelength conversion in the WDM network $G$ corresponds to an edge in $G_M$ respectively. By transforming $G$ into $G_M$, we have effectively converted the channel faults and wavelength conversion faults in the WDM network into only edge faults in $G_M$. We give in this section a set of efficient algorithms for point-to-point requests, multicast and multiple multicast on our cost model of Equations 2-3 in an unreliable WDM network. We assume that $G_M$ in this section is $(f+1)$-edge connected, so that any $f$ faulty edges of the same direction at one node will not disconnect $G_M$.

### 4.1 Point-to-Point Routing

We consider routing in an unreliable WDM network with $f$ edge faults in $G_M$. Let $F = \{e_1^*, e_2^*, \ldots, e_f^*\}$ be the set of edges that are faulty. The whole process of routing consists of the following consecutive three stages: 1. finding path; 2. establishing the found path; and 3. transmitting message along the the established path. $F$ can be known locally at each associated node in $G_M$ at different stages of routing, requiring different strategies for fault-tolerance. Note that we do not require global state consensus. We consider three cases respectively:

1. $F$ is known before routing stage;
2. $F$ is known after finding path, but before establishing the found path;
3. $F$ is known after establishing the found path, but before transmitting message along the established path.

For Case 1, since $F$ is known before path finding, simply assigning infinitely large weight to each faulty channel will convert the unreliable network to reliable network and hence, algorithms described in the previous section will apply.

For Case 2, which is more realistic and general and hence, of our interest, we establish multiple paths for each edge in $G_M$ such that for any portion of $F$ falling to a path, we are able to choose an available alternative path from them to skip the faculty edges. This approach is better and more practical than the two straightforward solutions. The first finds $f+1$ edge-disjoint shortest paths from $s_i$ to $t_i$ for each $i$. This, however, can generate paths of unpredictable large weight. The second computes and stores the shortest path from $s_i$ to $t_i$ for each of the cases when taking $f$ edges out from the $s_i$–$t_i$ shortest path computed by Algorithm **A.1**, in order to achieve a minimum cost. This will, however, require clearly exponential time, since the number of such cases is exponential to $f$.

For Case 3, different strategies can be applied to obtain a solution. One method is that routing for each request is realized by implementing a reliable communication protocol that requires the receiver to acknowledge receipt of message to the sender for each step of routing. This requires considerable overhead, but guarantees reliability. Another method requiring less overhead is to construct two or more edge-disjoint paths for each request and send message along these paths independently. This method is more efficient but does not guarantee reliability. Message is sent along the shortest path established by Algorithm **A.1**. At any step if a sender $u$ doesn't receive an acknowledgment from a receiver $v$, it should assume that there is an edge fault on the path from $u$ to $v$, and as a result an alternative path from $u$ to $v$ is sought and message is sent along that path.

Our purpose here is to present a simple and efficient solution with the above approach for fault-tolerant routing in Case 2. Given $r$ requests $\mathcal{R} = \{(s_1, t_1), (s_2, t_2), \ldots, (s_r, t_r)\}$, for each request $(s_i, t_i)$ we are required to establish a robust communication path from $s_i$ to $t_i$ that can tolerate any (up to) $f$ edge faults in each of the above two cases. Note that since our routing involves circuit switching, a physical path from source to destination must be established before the message transmission takes place. We propose the following approach.

Let $P_i = \{e_i^1, e_i^2, \ldots, e_i^{p_i}\}$ be the shortest path from $s_i$ to $t_i$, where $e_i^j = (h(e_i^j), t(e_i^j))$. Assume that the size of $F$, $f$, is known before routing stage (but not the content of $F$). We compute $f$ alternative paths across two endpoints after removal of each $e \in E_M$, that are edge-disjoint shortest paths in length increasing (non-decreasing) order and have the same orientation as edge $e$. This is done as preprocessing only once without knowing $F$. During the course of establishment of paths for the routing requests when $F$ is known before routing stage, we then chain several such paths together to skip all faulty edges in $F$ if they are encountered. The edge-disjointness of the $f$ redundant paths for each edge on $P_i$ guarantees that an alternative path can always be found in routing stage in the presence of any $f$ faulty edges.

We assume that both preprocessing (Algorithm **A.2**) and path establishment (Algorithm **A.3**) are executed in the way of source routing, where **A.2** finds alternative routes from $h(e)$ to $t(e)$ when $e$ is removed, and **A.3** establishes physical paths to skip all faulty edges on each path found by Algorithm **A.1** in the previous section.

Algorithm **A.2**
{*Construct alternative paths for every edge $e \in E_M$.*}

**for** every edge $e \in E_M$ **do**
Find $f$ shortest paths connecting $h(e)$ to $t(e)$ in $E_M - \{e\}$ that are edge-disjoint with each other;
Store these paths in $\mathcal{P}(e)$ according to length increasing order in $\mathcal{P}(e)[1, f]$.

When all redundant paths are computed by Algorithm **A.2**, path establishment from $s_i$ to $t_i$ along each path found by **B.1** in the presence of $F = \{e_1^*, e_2^*, \ldots, e_f^*\}$ is carried out as follows: For any path $P = \{e_1, e_2, \ldots, e_p\}$, let $t(P) = t(e_1)$ and $h(P) = h(e_p)$. We say that node $u$ precedes node $v$ in $P$, denoted by $u \prec v$, if there exist $e_i$ and $e_j$ in $P$, $i \le j$, such that $t(e_i) = u$ and $h(e_j) = v$. That is, node $u$ precedes node $v$ in $P$ if $u$ appears before $v$ on path $P$. We use $P[t(e_i), h(e_j)]$ to denote the segment path $\{e_i, e_{i+1}, \ldots, e_j\}$ in $P$, and $R \parallel P$ to denote the update to path $R$ with $P$ such that $R \parallel P$ contains the shortest path from $t(R)$ to $h(P)$. Below is the algorithm:

Algorithm **A.3**
{*Establish a physical fault-free path for every request $(s_i, t_i)$ along the already found path $P_i = \{e_i^1, e_i^2, \ldots, e_i^{p_i}\}$ by **A.1**, $1 \le i \le r$.*}
**for** $i = 1$ to $r$ **do**

1. $R_i := e_i^1$;
2. **for** $j = 1$ to $p_i$ **do**
   **if** $(e_i^j \in F) \wedge (h(R_i) \prec h(e_i^j))$ then
   {*If $h(e_i^j) \prec h(R_i)$ removal of $e_i^j$ will have no effect on $R_i$.*}
   $u := 1$;
   **while** $\mathcal{P}(e_i^j)[u] \cap F \neq \emptyset$ **do**
   $u := u + 1$;
   {*Choose a shortest path across $V^1[P_i - \{e_i^j\}]$ and $V^2[P_i - \{e_i^j\}]$ that contains no faulty edges.*}
   $R_i := R_i \parallel \mathcal{P}(e_i^j)[u]$;
   {*Include the chosen path into the route.*}
3. $R_i := R_i \parallel P_i[h(R_i), t_i]$;
   {*Include the last segment of $P_i$ that connects $h(R_i)$ to $h(e_i^{p_i}) = t_i$.*}
   Add $\delta[k](t(e))$ to $d_{\lambda_k}(e)$ for each $e \notin P_i$ using wavelength $\lambda_k$.
   {*Increase the queuing delay for each replacement edge.*}

Note that since we do not know which replacement path will be used for each edge in $P_i$, we do not assign any queuing delay for edges on replacement paths when we construct the redundant paths in Algorithm **A.2**. Instead, we assign each such edge a queuing delay when the replacement path is physically included in the path for routing message in Algorithm **A.3**. This is different from the queuing delay calculation for the edges in $P_i$ in **A.1**, for which we know beforehand that all edges on $P_i$ will be used for the $i$th request routing.

**Lemma 4.** *Algorithm A.3 finds a path correctly for each request $(s_i, t_i)$ when there are $f$ faulty edges in $G_M$.*

**Proof.** We show that Algorithm **A.3** always finds a path for request $(s_i, t_i)$ for any set $F$ of $f$ faulty edges. This can be seen from the fact that for any $e_i^j \in P_i$ there are $f$ edge-disjoint paths in $\mathcal{P}(e_i^j)$ and therefore at least one of them contains no faulty edges for any set of $f$ faulty edges. Without loss of generality, we assume that $F' = F \cap P_i = \{e_1, e_2, \ldots, e_{f_i'}\}$ and $\mathcal{P}(e_j)[r_j]$ is the path in $\mathcal{P}(e_i^j)$ that contains no faulty edges for $1 \le j \le f_i'$. Thus, chaining all these paths together with the last segment of $P_i$ that connects $h(\mathcal{P}(e_j)[r_{f_i'}])$ to $t_i$, that is, $(\bigcup_j \mathcal{P}(e_j)[r_j]) \bigcup P_i[h(\mathcal{P}(e_j)[r_{f_i'}]), t_i]$, constitutes a path from $s_i$ to $t_i$. □

Next we show the quality of the path found by Algorithm **A.3**.

**Lemma 5.** *If $F \subset P$, Algorithm A.3 finds a path connecting the first node to the last node of path $P$ whose length is no more than $f$ times of the optimal (shortest) one.*

**Proof.** Let $P = \{e_1, e_2, \ldots, e_p\}$, $p > f$. Assume that the optimal path connecting $t(e_1)$ to $h(e_{j_f})$ that skip all edges in $F \subset P$ has a length increase $\Delta_F$ over $P$. Suppose that for each edge $e_j$, $1 \le j \le p$, the alternative path that skips $e_j$ computed in Algorithm **A.2** has a length increase $\Delta_{1,j}$. Clearly $\Delta_{1,j} \le \Delta_F$ for all $j$, since the former has more candidate edges than the latter for the alternative path. In the case when all $f$ edges in $F = \{e_{j_1}, e_{j_2}, \ldots, e_{j_f}\}$ are simultaneously faulty, Algorithm 3 simply chains the alternative path (segment) skipping $e_{j_1}$ with that skipping $e_{j_2}, \cdots$, and with that skipping $e_{j_f}$ one-by-one. Thus, it introduces a total length increase $\sum_{q=1}^{f} \Delta_{1,j_q}$. Since $\sum_{q=1}^{f} \Delta_{1,j_q} \le f\Delta_F$, each path found by Algorithm **A.3** is at most $f$ times worse than the optimal one. □

In the general case $F \not\subseteq P$, the quality of the paths found by Algorithm **A.3** does not follow the above $f$-factor approximation. In fact, in order to achieve $f$-approximation, an algorithm needs to compute alternative paths that skip

every edge on the previously computed alternative path recursively. This will clearly require exponential time.

Using Dijkstra's algorithm to find a shortest path in $G_M$ in $t_{SP} = O(|E_M| + |V_M| \log |V_M|)$ time, we can complete Algorithm **A.2** in time

$$f|E_M|t_{SP} = O(f|E_M|(|E_M| + |V_M| \log |V_M|)).$$

For Algorithm **A.3**, we use a bit-vector $B$ to represent all edges in $F$—$B[i] = 1$ if $e_i \in F$ and $B[i] = 0$ otherwise, and record at each node its rank in path $P_i$ at the same time when $P_i$ is constructed in **A.2**. With this, we can check $(e_i^j \in F) \wedge (h(R_i) \prec h(e_i^j))$ in $O(1)$ time and $\mathcal{P}(e_i^j)[r] \cap F \neq \emptyset$ in $O(|E_M|/\log |V_M|)$ time. Consequently, we can complete Algorithm **A.3** in $O(\sum_{i=1}^{r} p_i f|E_M|/\log |E_M|) = O(fr|V_M||E_M|/\log |E_M|)$ time. By equations 4 and 5 we have the following theorem:

**Theorem 4.** *When $f$ redundant paths are precomputed for each edge in $G_M$, the problem of multiple point-to-point routing for $r$ requests in an unreliable WDM network with up to $f$ faulty optical channels and wavelength conversion gates can be solved in*

$$O(frk^2n(kn + m)/\log(kn))$$

*time, where $n$, $m$, and $k$ are the number of nodes, links, and available wavelengths in the network, respectively.*

Clearly, if multiple point-to-point routing in a reliable WDM network requires time $t_{MPP}$ (Algorithm **A.1**), the same problem in an unreliable WDM network with $f$ faulty edges can be solved in $O(fkn/\log(kn)t_{MPP})$ time. Precomputing all redundant paths by Algorithm **A.2**, executed only once, requires $O(k^2 f(kn + m)(kn + m + n \log(kn)))$ time. All redundant paths are stored permanently for use by different routing requests.

### 4.2 Multicast

Now we consider the problem of multicast in an unreliable WDM network modeled by $G_M$. For multicast $\mathcal{M} = (s, D)$ that transports information from source node $s$ to all destination nodes in set $D = \{t_1, t_2, \cdots, t_g\}$, we know that in a robust $G_M$, $\mathcal{M}$ can be accomplished through sending message along the tree edges of a multicast tree $MT$ rooted at $s$. From Section 3.2, we also know that there is an $MT$ that can be constructed efficiently whose cost is within an approximation ratio of less than 2 to that of the optimal Steiner tree in the expected case. Our task in this section is to construct a communication structure for $\mathcal{M}$ that can tolerate any $f$ edge faults in an unreliable $G_M$.

Let $F = \{e_1^*, e_2^*, \ldots, e_f^*\}$ be the set of faulty edges in $G_M$ which are known after routing stage 1 and before routing stage 2. For other cases, solutions are the same as given in the previous section. The basic idea to achieve fault-tolerant multicast is to enhance every edge in multicast tree $MT$ with multiple alternative paths such that $MT$ is always connected via at least one of these paths in the case that all edges in $F$ are broken for any $F$. To achieve this, a trivial solution is to compute $(f + 1)$ edge-disjoint minimum spanning trees of $G_M$. Another straightforward approach is to establish $k$ edge-disjoint alternative paths for each edge in $MT$ that connect the two endpoints of the edge such that the two endpoints are always connected via one of these paths in case of $k$ faulty edges. These two approaches, although both feasible, do not provide a low cost to the modified $MT$. In order to maintain the cost of $MT$ as small as possible, a better approach is to reconnect the two connected components, not necessarily the two endpoints of the faulty edge, when an edge in $MT$ is faulty. For a faulty edge $e = (u, v)$, let $MT^{(u)}$ and $MT^{(v)}$ be two connected components (trees) after removal of $e$, where $MT^{(u)}$ and $MT^{(v)}$ contain endpoints $u$ and $v$, respectively. Our approach first calls Algorithm **A.2** to enhance each edge in $G_M$ with $f$ replacement paths (redundant edges), so that an $MT$ constructed in $G_M$ can tolerate any $F$ edge faults. As in **A.2**, this is done only once as a preprocessing procedure. We then find a shortest replacement path connecting node $u$ and any node in $MT^{(v)}$ for any faulty $e = (u, v) \in E(MT)$ after $MT$ has been found by **B.1**.

When each edge in $G_M$ has been enhanced with $f$ alternative paths by Algorithm **A.2**, after the multicast tree $MT$ has been found by Algorithm **B.1** for multicast request $\mathcal{M}$, path establishment on $MT$ in the presence of any up to $f$ faulty edges $F = \{e_1^*, e_2^*, \ldots, e_f^*\}$ is carried out as follows: Let $\{e_1, e_2, \ldots, e_{|E(MT)|}\}$ be level-by-level ordered edges of $MT$. The multicast proceeds by sending message originated at root $s$ along edges $e_i$ for $i = 1$ to $|E(MT)|$ in $MT$. If edge $e$ is faulty, then an alternative path of the shortest length that does not contain any faulty edge is chosen from $\mathcal{P}(e)$ to deliver the message. To support faulty edge detection, each path in $\mathcal{P}(e)$ uses a bit-vector of $|E_M|$ bits to store the presence of each edge of $G_M$ in the path—"0" for nonpresence and "1" for presence. To facilitate alternative path selection, all paths in $\mathcal{P}(e)$ are stored in the order of their increasing lengths. We use an array of $f \times |E_M|$ bits for $\mathcal{P}(e)$ and let $F$ store the global indices of all faulty edges, that is, faulty edge $e*_i = e_{F[i]}$ for $1 \leq i \leq f$. Thus, we have immediately the following multicast path establishment algorithm, which is called for each multicast request after the multicast tree $MT$ has been found by **B.1** and executed in the way of source routing.

Algorithm **B.2**
{*Establish physical paths for message multicast from the root in $MT$ found by **B.1**.*}

**for** $i = 1$ to $|E(MT)|$ **do**
  **if** $e_i \in F$ **then**
    Deduct $\delta[k](e_i)$ from $d_{\lambda_k}(e_i)$ if $e_i$ uses wavelength $\lambda_k$;
{*Reduce its queue length by 1 to reflect release of channel $e_i$.*}
  $j := 1$; ~ $alt := FALSE$;
  **while** $(j \leq f) \wedge (alt = TRUE)$ **do**
    $q := 1$; ~ $alt := TRUE$;
    **while** $(q \leq f) \wedge (alt = TRUE)$ **do**
      **if** $\mathcal{P}(e_i)[j][F[q]] = 1$ then $alt := FALSE$;
      $q := q + 1$;
    $j := j + 1$;
{*Choose a shortest path in $\mathcal{P}(e)$ that contains no faulty edges.*}
    **if** the above replacement path contains a node $u' \in MT^{(u)}$ **then**

Delete the edge pointing to $u'$ in $MT^{(u)}$;
{*Eliminate 'loop' while maintaining the path connecting from $u$ to $MT^{(v)}$.*}
$MT = MT \parallel \mathcal{P}(e)[j]$:
Add $\delta[k](t(e'))$ to $d_{\lambda_k}(e')$ for each $e' \in \mathcal{P}(e)[j]$ using wavelength $\lambda_k$.
{*Update $MT$ and the edge weight for each edge on the new path.*}

Note that 'loop' in the above means more than one incoming edges to a tree node. It is not a loop in the directed sense.

**Lemma 6.** *Algorithm* **B.2** *correctly implements multicast $\mathcal{M}$ in an unreliable WDM network with up to $f$ faulty edges in $G_M$.*

**Proof.** The correctness of the lemma can be seen easily from the fact that for each $e \in E(MT)$ in $G_M$ that models the WDM network, there are $f$ edge-disjoint paths in $\mathcal{P}(e)$ constructed by Algorithm **A.2**, and therefore, at least one of these paths will pass messages through for any set of $f$ faulty edges. □

We know that Algorithm **A.2** can be completed in time $O(f|F_M|(|E_M| + |V_M|\log|V_M|))$, which is the preprocessing time. Using Algorithm **B.1** to construct $MT$ in time $t_{MT}$ and Dijkstra's algorithm to find a shortest path, Algorithm **B.2** requires $O(|E_{MT}|f^2 + |V_M|\log|V_M|)$ time. With $|V_M| = 2kn$ and $|E_M| = k^2n + km$, we have the following theorem:

**Theorem 5.** *The problem of multicast of group size $g$ in an unreliable WDM network with up to $f$ faulty optical channels and wavelength conversion gates can be solved in $O(kf^2(kn + m) + kn\log(kn))$ time, with preprocessing support of $O(k^2f(kn + m)(kn + m + n\log(kn)))$ time, where $n$, $m$, and $k$ are the number of nodes, links, and available wavelengths in the network, respectively.*

The preprocessing here is the same as for multiple point-to-point routing implemented by Algorithm **A.2**, and runs only once for all routing requests. Let multicast in a reliable WDM network require time $t_{MC}$ (Algorithm **B.1**). From the above discussion, it is clear that multicast in an unreliable WDM network with $f$ faulty edges requires $O(f^2/g^2 t_{MC})$ time.

### 4.3 Multiple Multicast

We now consider the general group communication pattern of multiple multicast. Let $\mathcal{M}_1, \mathcal{M}_2, \ldots, \mathcal{M}_r$ be $r$ groups of multicast, and $\mathcal{M}_i = (s_i, D_i)$, $1 \le i \le r$, where $s_i$ is a source and $D_i = \{t_i^1, \ldots, t_i^{g_i}\}$ are the destinations, $G_M$ must be at least $(f + r + 1)$-edge connected. In an unreliable WDM network with up to $f$ faulty edges in $G_M$ that are known after routing stage 1 and before routing stage 2, $\mathcal{M}_i$ alone can be realized by a multicast tree $MT_i$ constructed by Algorithm **B.2**. As we stated in Section 2.3, since all $MT_i$'s are constructed concurrently and independently, edges of different $MT_i$ in $MF = \cup MT_i$ may fall onto the same edge of $G_M$, and hence, possibly cause wavelength contention on the same optical link of the network. Thus, our task here is to construct all $MT_i$'s in such a way that results in a minimal wavelength contention for all the trees in $MF$. We use the same greedy approach as in Section 2.3 to achieve the above: construct an edge-enhanced $G_M$ for fault-tolerance by Algorithm **A.2** as preprocessing; then, after approximate multicast tree $MT_i$ for each multicast $\mathcal{M}_i$ has been found by Algorithm **C.1**, establish physical paths in each $MT_i$ one-by-one in size, increasing order in the presence of any $F$ applying Algorithm **B.2**, where edges of all physical paths in $MT_i$ are marked with an infinitely large weight as soon as they are established before establishing paths for the next tree. This will ensure that once physical paths in $MT_i$ are established, they will not be chosen again by paths in $MT_i$ for all $j > i$, and hence, no wavelength contention can occur on any link. Our algorithm for multiple multicast in an unreliable WDM network is described as follows:

Algorithm **C.2**
{*Establish physical paths for multicast trees $MT_{\pi_1}, MT_{\pi_2}, \ldots, MT_{\pi_r}$ sorted in size increasing order found by **C.1** in an unreliable WDM network with up to $f$ faulty edges.*}

**for** $i = 1$ to $r$ **do**
Call Algorithm **B.2** to establish a set of physical routes $\mathcal{R}(MT_{\pi_i})$ for $MT_{\pi_i}$ that skips all faulty edges in $MT_{\pi_i}$;
For each $e \in \mathcal{R}(MT_{\pi_i}) - MT_{\pi_i}$ mark $w(e)$ with weight $\infty$.

The correctness of the above algorithm is obvious. The time complexity of the algorithm can be directly obtained from that of algorithms **A.2** and **B.2**. This results in the following theorem:

**Theorem 6.** *The problem of multiple multicast of $r$ groups of maximum size $g$ in an unreliable WDM network with up to $f$ faculty optical channels and wavelength conversion gates can be solved in $O(rk(f^2(kn + m) + n\log(kn)))$ time under the same preprocessing as for multiple point-to-point and multicast routing, where $n$, $m$, and $k$ are the number of nodes, links, and available wavelengths in the network, respectively.*

Let the time for multiple multicast routing in a reliable WDM network be $t_{MMC}$ (Algorithm **C.1**). It is clear that multiple multicast in an unreliable WDM network with $f$ faulty edges would require $O(f^2/\sum_{i=1}^{r} g_i^2 t_{MMC})$ time.

## 5 Concluding Remarks

We have proposed a set of efficient algorithms for fault-tolerant routing in WDM networks on a new model of routing cost. Our new cost model considers not only the costs for wavelength access and conversion as defined in the existing model [7], [20], but also the cost for queuing signals of the same output wavelength at the same node, which is an analogy of link multiplexing cost in traditional electronic networks that require link sharing. The communication patterns covered in our algorithms include three general types governing almost all current communication patterns: multiple point-to-point routing, multicast, and multiple-multicast. For each of these communication patterns, our algorithms on the new cost model are more complex than those on the existing ones because of the requirement of considering the queuing delay. For finding good routing solutions, we introduced an auxiliary graph

which can be used to minimize the total cost by considering three different costs simultaneously. Based on such an auxiliary graph, we obtained fault-tolerant communication algorithms by first developing a set of algorithms for the above communication patterns in a reliable WDM network on the new cost model, and then enhancing the physical routes constructed by these algorithms accordingly, such that routing among the enhanced routes can tolerate any $f$ faulty optical channels and wavelength conversion gates in the network. For each of the communication patterns, our algorithms run efficiently in time polynomial to the network size and the number of wavelengths. The algorithms for multiple point-to-point routing is optimal in the regard of minimizing path overlapping probability in the expected case. The algorithms for multicast and multiple multicast in reliable WDM network produce suboptimal outcome with an approximate ratio less than two in the expected case. It easily can be seen that all our algorithms are deadlock-free.

In the optical model used in this paper, we assumed that there is exactly one receiver at the input port for each available input wavelength, and one transmitter at the output port for each available output wavelength. Our approach can be extended to handle the case when there are more or fewer receivers/transmitters than the number of available wavelengths at each node by simply using more or less queues (buffers), and modifying the switching structure in Fig. 1. This will result in a modified auxiliary graph reflecting the above changes, making our proposed algorithms still applicable with minor changes.

In our routing algorithms, the existence of multiple routes for multiple point-to-point and multiple multicast problems is guaranteed by sufficient network connectivity. In case the connectivity is relatively small, it is not always possible to succeed in finding all routes. In such a case, some routing requests may be rejected, which is common in circuit-switched networks. Hence, routing with QoS according to the priority of request will be imposed for this case. Moreover, QoS routing on multiple resource constraints, such as priority, bandwidth and timing, will also become necessary in order to make WDM networks suitable for applications of real-time traffic with QoS, such as VoD. We believe that the work presented in this paper also provides a framework for development of new models and algorithms for QoS routing in future WDM networks. This shall remain as a focus of our future research.

## ACKNOWLEDGMENTS

This research project was supported by the following research grants: The Australian Research Council under its Large Grants Scheme (1996-98) A849602031 and Small Grants Scheme (1998) (for Hong Shen); RGC Grant Project # HKU 7024/98E (for Francis Chin); the U.S. National Science Foundation under Grants CCR-9211621, OSR-9350540, and CCR-950388 (for Yi Pan). The authors wish to express their sincere thanks to their colleagues and anonymous reviewers who read the manuscript carefully and provided many constructive comments and suggestions, which greatly helped to shape this paper.

## REFERENCES

[1] A. Agrawal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan, "Efficient Routing in Optical Networks," *J. ACM*, vol. 46, pp. 973–1,001, 1996.

[2] Y. Aumann and Y. Rabani, "Improved Bounds for All Optical Routing," *Proc. Sixth Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '95)*, pp. 567–576, 1995.

[3] R.A. Barry and P.A. Humblet, "On the Number of Wavelengths and Switches in All-Optical Networks," *IEEE Trans. Comm. (Part I)*, pp. 583–591, 1994.

[4] B. Beauqier, J.C. Gargano, S. Perenees, P. Hell, and U. Vaccaro, "Graph Problems Arising from Wavelength-Routing in All-Optical Networks," *Proc. Second Workshop Optics and Computer Science (WOCS)*, 1997.

[5] K.-M. Chan and T.-S. Yum, "Analysis of Least Congested Path Routing in WDM Lightwave Networks," *Globecom*, pp. 962–969, 1994.

[6] K.W. Cheung, "Scalable, Fault-Tolerant 1-Hop Wavelength Routing," *Globecom*, pp. 1,240–1,244, 1991.

[7] I. Chlamtac, A. Farag, and T. Zhang, "Lightpath (Wavelength) Routing in Large WDM Networks," *IEEE J. Selected Areas Comm.*, vol. 14, pp. 909–913, 1996.

[8] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath Communications: A Novel Approach to High Bandwidth Optical WAN's," *IEEE Trans. Comm.*, vol. 40, pp. 1,171–1,182, 1992.

[9] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[10] T. Erlebach and K. Jansen, "Scheduling of Virtual Connections in Fast Networks," *Proc. Fourth Workshop Parallel Systems and Algorithms (PASA '96)*, pp. 13–32, 1996.

[11] J.C. Gargano, P. Hell, and S. Perences, "Colouring All Directed Paths in a Symmetric Tree with Applications to WDM Routing," *Proc. ICALP '97*, pp. 505–515, 1997.

[12] L. Gargano, "Limited Wavelength Conversion in All-Optical Networks," *Proc. 25th Int'l Colloquium Automata, Languages and Programming*, pp. 544-555, 1998.

[13] P.E. Green, *Fiber-Optic Communication Networks*. Prentice Hall, 1992.

[14] K. Kaklamanis, G. Persiano, T. Erlebach, and K. Jansen, "Constrained Bipartite Edge Coloring with Applications to Wavelength Routing," *Proc. ICALP '97*, pp. 460–470, 1997.

[15] L. Kou, G. Markowsky, and L. Berman, "A Fast Algorithm for Steiner Trees," *Acta Informatica*, vol. 15, pp.141–145, 1981.

[16] K. Bharath-Kumar and J. M. Jaffe, "Routing to Multiple Destinations in Computer Networks," *IEEE Trans. Comm.*, vol. 31, pp. 343–351, 1983.

[17] E. Kumar and E. Schwabe, "Improved Access to Optical Bandwidth in Trees," *Proc. Eighth Ann. ACM-SIAM Symp. Discrete Algorithms (SODA '97)*, pp. 437–44, 1997.

[18] H.-M. Lee and G.J. Chang, "Set-to-Set Broadcasting in Communication Networks," *Discrete Applied Math.*, vol. 40, pp. 411–421, 1992.

[19] *Parallel Computing Using Optical Interconnections*, K. Li, Y. Pan, and S.Q. Zheng, eds., Kluwer Academic Publishers, 1998.

[20] W. Liang, G. Havas, and X. Shen, "Improved Lightpath Routing in Large WDM Networks," *Proc. 18th Int'l Conf. Distributed Computing Systems*, pp. 516–523, 1998.

[21] W. Liang and H. Shen, "Multicast and Broadcast in Large WDM Networks," *Proc. 12th Int'l Parallel Processing Symp (IPPS/SPDP)*, pp. 365–369, 1998.

[22] R. Malli, X. Zhang, C. Qiao, "Benefit of Multicasting in All-Optical WDM Networks," *Conf. All-Optical Networks (SPIE)*, vol. 3531, pp. 209-220, 1998.

[23] G. De Marco, L. Gargano, and U. Vaccaro, "Concurrent Multicast in Weighted Networks," manuscript.

[24] A.D. McAulay, *Optical Computer Architectures*. John Wiley, 1991.

[25] M. Mihail, K. Kaklamanis, and S. Rao, "Efficient Access to Optical Bandwidth," *Proc. FOCS '95*, pp. 548–557, 1995.

[26] B. Mukherjee, *IEEE Comm.*, Jan./Feb. 1999.

[27] Y. Ofek and B. Yener, "Reliable Concurrent Multicast from Bursty Sources," *Proc. IEEE INFOCOM '96*, pp.1,433–1,441, 1996.

[28] P. Raghavan and E. Upfal, "Efficient Routing in All-Optical Networks," *Proc. STOC '94*, pp. 133–143, 1994.

[29] R. Ramaswami, "Multi-Wavelength Lightwave Networks for Computer Communication," *IEEE Comm.*, vol. 31, pp. 78–88, 1993.

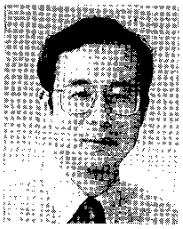
[30] G.N. Rouskas and M.H. Ammar, "Analysis and Optimization of Transmission Schedules for Single-Hop WDM Networks," *Infocom '93*, pp. 1342–49, 1993.
[31] G.N. Rouskas and M.H. Ammar, "Multi-Destination Communication over Tunable-Receiver Single-Hop WDM Networks," Technical Report, TR-96-12, Department of Computer Science, North Carolina State University.
[32] L.H. Sahasrabuddhe and B. Mukherjee, "Light-Trees: Optical Multicasting for Improved Performance in Wavelength-Routed Networks," *IEEE Comm.*, vol. 37, no. 2, pp. 67-73, 1999.
[33] H. Shen, "Efficient Multiple Multicasting in Hypercubes," *J. System Architectures* vol. 43, no. 9, pp. 655–662, 1997.
[34] H. Shen and W. Liang, "Efficient Multiple Multicast in WDM Networks," *Proc. 1998 Int'l Conf. Parallel and Distributed Processing Techniques and Applications*, pp. 1,028–1,033, 1998.
[35] R.J. Vitter and D.H.C. Du, "Distributed Computing with High-Speed Optical Networks," *Computer*, vol. 26, pp. 8–18, 1993.
[36] S.S. Wagner and H. Kobrinski, "WDM Applications in Broadband Telecommunication Networks," *IEEE Comm.*, vol. 27, no. 3, pp. 22–30, 1989.
[37] Z. Zhang and A.S. Acampora, "A Heuristic Wavelength Assignment Algorithm for Multihop WDM Networks with Wavelength Routing and Wavelength Reuse," *IEEE J. Networking*, vol. 3, pp. 281–288, 1995.

**Hong Shen** received the BS degree from Beijing University of Science. and Technology (previously known as Iron and Steel Technology) in 1982, the MS degree from the University of Science and Technology of China in 1987, and the PhLic and PhD degrees from Abo Akademi University, Finland, in 1990 and 1991, respectively, all in computer science. He became an assistant professor in the Department of Computer Science, Abo Akademi University, in 1991. He joined the School of Computing and Information Technology at Griffith University as a lecturer in 1992, where he was then promoted to senior lecturer and associate professor (reader). Dr. Shen is the research director for the Parallel Computing Unit at Griffith University. With main research interests in algorithms, parallel and distributed computing, interconnection networks, parallel databases and data mining, multimedia systems, and networking, he has authored more than 120 technical papers.

Dr. Shen has served as an editor of the *Journal of Parallel and Distributed Computing Practice*, associate editor of the *International Journal of Parallel and Distributed Systems and Networks*, and on the editorial board of the *Journal of Parallel Algorithms and Applications*, the *Journal of Supercomputing*, and the *International Journal of Computer Mathematics*.

**Francis Chin** received the BSc degree in engineering science from the University of Toronto, Toronto, Canada, in 1972, and the MS, MA, and PhD degrees in electrical engineering and computer science from Princeton University, New Jersey, in 1974, 1975, and 1976, respectively. He is a fellow of the IEEE, HKIE, and HKACE.

Since 1975, Prof. Chin has taught at the University of Maryland, University of California San Diego, University of Alberta, and Chinese University of Hong Kong. He is currently head of the Department of Computer Science at the University of Hong Kong. He has served on the editorial boards for *Computer Processing of Chinese and Oriental Languages*, *Information Processing Letters*, *HKIE Transactions*, and the *Chinese Journal of Advanced Software Research*. He has also served on the IEEE Computer Society Fellow Evaluation Committee, and a number of Hong Kong government IT-related committees. His current research interests include algorithm design and analysis and parallel and distributed computing.



**Yi Pan** received his BEng degree in computer engineering from Tsinghua University, China, in 1982, and his Ph.D. degree in computer science from the University of Pittsburgh, USA, in 1991. He joined the Department of Computer Science at the University of Dayton, Ohio, in 1991, and has been an associate professor since 1996. He has published more than 70 papers in international journals and conference proceedings. He is the co-recipient of the Best Paper Award at the Second International Conference on Parallel and Distributed Processing Techniques and Applications in 1996. His research has been supported by the NSF, AFOSR, U.S. Air Force, and the state of Ohio.

Dr. Pan has served as a guest editor of special issues for *Parallel Processing Letters*, *International Journal of Parallel and Distributed Systems and Networks*, and *Informatica*. He is currently an editor of the *Journal of Parallel and Distributed Computing Practices*, associate editor of *International Journal of Parallel and Distributed Systems and Networks*, and on the editorial board of the *Journal of Supercomputing* and the *Journal of Information*. He was the program chair of the 10th International Conference on Parallel and Distributed Computing and Systems.