INVITED SURVEY PAPER *Special Invited Survey*

Multicasting in Multihop Optical WDM Networks with Limited Wavelength Conversion

Hong SHEN[†], **Yi PAN**[††], **John SUM**[†††], ***Nonmembers***, ***and*** **Susumu HORIGUCHI**[†], ***Regular Member***

SUMMARY This paper provides an overview on efficient algorithms for multicasting in optical networks supported by Wavelength Division Multiplexing (WDM) with limited wavelength conversion. We classify the multicast problems according to off-line and on-line in both reliable and unreliable networks. In each problem class, we present efficient algorithms for multicast and multiple multicast and show their performance. We also present efficient schemes for dynamic multicast group membership updating. We conclude the paper by showing possible extension of the presented algorithms for QoS provision.

key words: *multicast, optical networks, routing, wavelength conversion, wavelength division multiplexing*

1. Introduction

Optical networking delivers promises for various applications that require ultra-high data transmission rates [1], [30], [33], [41]. A key technology to implement optical networks is *Wavelength-Division Multiplexing* (WDM) [42] that divides the optical spectrum in fiber-optic into many channels, each corresponding to an optical wavelength, and thus allows multiple laser beams carrying different data streams to be transferred concurrently along a single fiber-optic provided that each beam uses a distinct wavelength. All nodes in an (optical) WDM network are interconnected by point-to-point fiber-optic links, where each link can support a set of wavelengths. Attached to each node are a set of input ports receiving incoming data and output ports delivering out-going data. A WDM network allows an incoming signal to be routed to one or more output ports, but not multiple signals simultaneously to the same output port on the same wavelength. Multiple incoming signals are allowed to use the same output wavelength in the same output port at different times through a queue (buffer) [35]. In *switched* (also known as *reconfigurable*) *multihop* WDM networks, signals on input ports at each node are routed to appropriate output ports directing to their destinations via a set of switches, and wavelength conversion during the course of transmission may happen at some intermediate nodes on the communication path. At each of these intermediate nodes, the signal is converted from optic form to electrical form and then retransmitted on another wavelength (converted back from electronic to optic). In multi-hop networks, a communication path between a source-destination pair is called *semilightpath* [7] which is obtained by establishing and chaining several lightpaths together.

Multicast as an important communication pattern of great practical significance requires to transport information from a given 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 [34]. Multicasting in a WDM network requires to set up a communication path from source to each destination node by chaining a set of optical channels together, with all channels on each path being assigned a number of wavelengths, and channels of different paths sharing the same optical link having different wavelengths. *Off-line* multicasting constructs all paths before message routing actually takes place, whereas *on-line* multicasting routes messages simultaneously while the underlying paths are being constructed. Routing can be carried out in a WDM network that is either *reliable* where no faults can occur, or *unreliable* if hardware faults including optical channel and wavelength conversion faults may exist. There is an extensive literature for routing in both single-hop (all-optical) and multihop optical networks [1]–[3], [6], [7], [11], [12], [15], [18], [21], [26], [29], [31], [32], [43], [44]. Recently research on multicast in WDM networks has also become active [10], [22], [23], [27], [28], [33], [35], [36], [38], [39], [45].

This paper gives an overview on some recent results on multicast in multihop optical WDM networks with limited wavelngth conversion. Section 2 shows a general cost model for multicast in WDM networks that 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. Section 3 presents algorithms for efficient off-line multicating

Manuscript received August 2, 2002.

[†]The authors are with the Graduate School of Computer Science, Japan Advanced Institute of Science and Technology, Ishikawa-ken, 923–1292 Japan.

[††]The author is with the Department of Computer Science, Georgia State University, Atlanta, Georgia 30303, USA.

[†††]The author is with the Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.

in reliable WDM networks. Section 4 presents algorithms for efficient off-line multicasting in unreliable WDM networks. Sections 5 and 6 present algorithms for on-line multicasting in reliable and unreliable networks respectively. Section 7 addresses dynamic group membership maintenance for on-line multicast. This paper is an enhanced version of [37] and based on the results of [35], [38], [39].

2. A General Cost Model for Multicast

Let $\Gamma = \{\lambda_1, \lambda_2, \ldots, \lambda_k\}$ be the set of available wavelengths in a WDM network of n nodes. A node contains multiple input ports and multiple output ports. Each input port is equipped with a dedicated electronic *receiver* with buffering capability that converts the incoming signal from optical to electronic. Likewise, each output port is equipped with a dedicated laser *transmitter* that converts the outgoing signal from electronic to optical. Connected to the receivers is a cross-bar like switch that switches incoming signals to a set of buffers, each queuing all the signals that are routed simultaneously to the same optical channel (wavelength) of an output port so that they can be directed to the corresponding transmitter one by one. In the following we shall give a cost model for multicast in WDM networks under the above assumptions.

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

Let P be a semilightpath connecting a pair of nodes in the network to fulfill a routing request. Clearly P consists of a sequence of optical channels $e_1, e_2, \ldots, e_l$, where e_i carries wavelength λ_{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 $1 \le i < l$.

For multicast, we are required to construct a *multicast tree* MT rooted at s that connects all destination nodes within the multicast group. 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. We denote the queuing delay for transmitting any incoming signal using wavelength λ 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 λ on edge e. The following cost model $C(MT)$ for traversing MT was defined in [35].

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

To support routing, the following method was given in [35] that transforms $G = (V, E, \Gamma)$ into another auxiliary graph $G_M = (V_M, E_M)$.

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 λ_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.

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, and E_v represent 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 $e = (u, v) \in E$ with $|\Gamma_e| \le k$ parallel edges (channels), E_e. For each $e' \in E_e$ carrying wavelength λ_i, assign edge weight $w(e, \lambda_i)$ to it. These edges connect vertices in B_u to the corresponding vertices in A_v.
3. Assign an edge weight $d_{\lambda_i}(e')$, initially zero, to edge e' (representing outgoing wavelength λ_i) in E_e, indicating the queuing delay for sending message from $h(e') \in B_v$ to $t(e') \in A'_v$ using wavelength λ_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 $E_M = (\cup_{v \in V}(E_v \cup E'_v)) \bigcup (\cup_{e \in E} E_e)$.

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

$$|V_M| \le 2kn, \quad (2)$$

$$|E_M| \leq k^2 n + km. \quad (3)$$

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. Whenever appropriate, we use *weight* and *cost* interchangeably.

3. Off-Line Multicasting in Reliable Networks

In this section we present a set of efficient algorithms for multicast and multiple multicast in a reliable WDM network on the proposed cost model (1).

3.1 Multicast

Multicast requires to transport information from source s to a set of destinations $D = \{t_1, t_2, \cdots, t_g\}$ and 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. Finding an optimal MT is equivalent to finding a minimum directed Steiner tree in G_M which is unfortunately NP-complete even when only static edge weights are considered. We therefore use an approach based on that of [17] to find an approximate solution to the Steiner tree in G_M: We first 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 $u, v \in \{s\} \cup D$ and then find the directed MST rooted at s instead of the undirected MST in the undirected case [17]. 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 cover all nodes in D by repeatedly adding an edge of direction outwards 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. It was shown [35] that the MST constructed by the above method is the minimum cost directed spanning tree that connects s to every other node in $I(\{s\} \cup D)$.

With the help of the above algorithm, we can present an algorithm for multicast in the WDM network as follows. 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 MC
{*Multicasting 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 λ_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 λ_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.*}

It has been shown in [35] that the MST_I obtained by Algorithm MC is $(2-\frac{2}{g})$-OPT. Since $|V_M| = 2kn$ and $|E_M| = k^2 n + km$ by Equations (2) and (3), Step 1 of the algorithm requires $O(g^2 k^2 n + g^2\, km + g^2 kn \log(kn))$ 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 1: 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 $O(g^2 k(kn + m + n \log(kn)))$ 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 $P[u \rightarrow v]$, MST_I may contain $|V_M|$ nodes because all these shortest paths may span over the entire G_M.

3.2 Multiple Multicast

When several groups of multicast wish to take place concurrently, a more general communication pattern, namely *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, $1 \leq i \leq r$ 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. Figure 1 shows an example of wavelength contention caused by 3 multicast trees.

While wavelength contention is forbidden in most conventional optical models, the optical model [35] we use does allow it to happen by buffering all signals using the same wavelength on the same physical link in a queue and then transmitting them out in packet switching in different time slots. In order to produce a minimal cost MF, we need to minimize the aggregated wavelength contention probability on all optical channels. Clearly wavelength contention probability on an optical channel in G is the edge overlapping probability on that channel's corresponding edge in G_M. We take a greedy approach to find an approximate optimal multicast tree for each multicast MT_i one by one employing Algorithm MC in size increasing order. This approach will minimize the tree overlapping probability, which is the average edge overlapping probability over all edges in the tree, for all trees in MF in the expected case when every edge in G_M has an equal probability to be used by all the trees. The algorithm for multiple multicast is described as follows:

Algorithm MMC
{*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_{π_i} for $\mathcal{M}_i$ using Algorithm MC.

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 the result for multicast in the previous section we have the following theorem.

Theorem 2: 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.

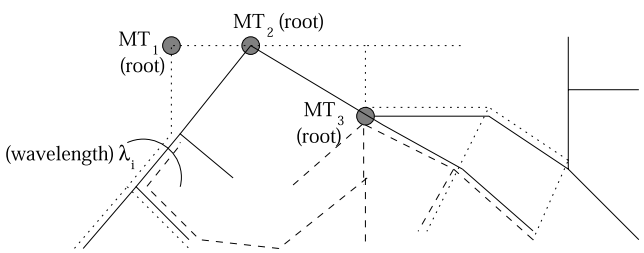


Fig. 1 Wavelength contention caused by 3 multicast trees.

The probability of edges of MT_{π_j} falling to those of MT_{π_i} is the probability of wavelength contention and hence queuing delay increase caused by MT_{π_i} and MT_{π_j} both wanting to access the wavelength represented by this edge (channel). In the expected case when all edges in G_M have an equal probability to be used by all multicast trees, the above heuristic is optimal in the sense that it minimizes the average probability of edges of MT_{π_j} falling to those of MT_{π_i} for $j > i$. Therefore Algorithm MMC has the same approximation ratio as Algorithm MC in this expected case.

4. Off-Line Multicasting in Unreliable Networks

Consider 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. By transforming G into G_M, we can convert the channel faults and wavelength conversion faults in the WDM network into only edge faults in G_M. We describe in this section efficient algorithms for multicast and multiple multicast on the cost model (1) 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. Let $F = \{e_1^*, e_2^*, \ldots, e_f^*\}$ be the set of edges that are faulty.

4.1 Multicast

Routing in an unreliable WDM network consists of the following consecutive three stages: (a) finding path, (b) establishing the found path, and (c) transmitting message along 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 we do not require global state consensus. We consider three cases respectively:

Case 1: F is known before routing stage (a);
Case 2: F is known after (a) but before (b);
Case 3: F is known after (b) but before (c).

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 that by finding f edge-disjoint shortest paths from s_i to t_i for each i or by storing all shortest paths from s_i to t_i when excluding different f edges.

For Case 3, different strategies can be applied to obtain a solution. We apply the following approach: Message is sent along the shortest path established originally. 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.

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 necessary 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 the following algorithm f-PATH 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. 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 Algorithm MC.

Algorithm f-PATH
{*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 each edge in G_M is enhanced with f alternative paths by Algorithm f-Paths, after the multicast tree MT has been found by Algorithm MC 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 non-presence 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 $1\le i\le f$. 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 Algorithm MC and executed in the way of source routing.

Algorithm FMC
{*Establish physical paths for message multicast from the root in MT found by Algorithm MC.*}
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 λ_k;
{*Reduce its queue length by 1 to reflect release of channel e_i.*}
$j:=1;\quad alt:=FALSE;$
while $(j\le f)\wedge(alt=TRUE)$ **do**
$q:=1;\quad alt:=TRUE;$
while $(q\le f)\wedge(alt=TRUE)$ **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 λ_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.

Algorithm f-PATH can be completed in time

$O(f|E_M|(|E_M|+|V_M|\log|V_M|))$ as preprocessing which is done only once for all multicast requests. Using Algorithm MC to construct MT in time t_{MT} and Dijkstra's algorithm to find a shortest path, Algorithm FMC requires $O(|E_{MT}|f^2+|V_M|\log|V_M|)$ time. With $|V_M|=2kn$ and $|E_M|=k^2n+km$, the following theorem was given in [35]:

Theorem 3: 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 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.

Let multicast in a reliable WDM network require time t_{MC} (Algorithm MC). From the above discussion it is clear that multicast in an unreliable WDM network with f faulty edges requires $O(f^2/g^2t_{MC})$ time.

4.2 Multiple Multicast

For r groups of multicast, $\mathcal{M}_i=(s_i,D_i)$, $1\le i\le r$, where s_i is source and $D_i=\{t_i^1,\ldots,t_i^{g_i}\}$ is destination set, 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 (a) and before routing stage (b), $\mathcal{M}_i$ alone can be realized by a multicast tree MT_i constructed by Algorithm FMC. As we stated in Sect. 3.3, 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. So 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 Sect. 3.3 to achieve the above: construct an edge-enhanced G_M for fault-tolerance by Algorithm f-PATH as preprocessing; then, after approximate multicast tree MT_i for each multicast $\mathcal{M}_i$ has been found by Algorithm MMC, establish physical paths in each MT_i one by on in size increasing order in the presence of any F applying Algorithm FMC. This will ensure that the tree overlapping probability is minimum for all trees in MF, and hence the probability of wavelength contention is minimal. Our algorithm for multiple multicast in an unreliable WDM network is described as follows:

Algorithm FMMC
{*Establish physical paths for multicast trees $MT_{\pi_1},MT_{\pi_2},\ldots,MT_{\pi_r}$ sorted in size increasing order found by Algorithm MMC in an unreliable WDM network with up to f faulty edges.*}
for $i=1$ **to** r **do**
Call Algorithm FMC to establish a set of physical routes $\mathcal{R}(MT_{\pi_i})$ for MT_{π_i}
that skips all faulty edges in MT_{π_i}.

The correctness of the above algorithm is obvious. The time complexity of the algorithm can be directly obtained from that of algorithms f-PATH and FMC. This results in the following theorem [35]:

Theorem 4: 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, with the same preprocessing as for multicast, where n, m and k are the number of nodes, links and available wavelengths in the network respectively.

Let the time for multiple multicast in a reliable WDM network be t_{MMC} (Algorithm MMC). 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. On-Line Multiple Multicast in Reliable Networks

In this section we present efficient algorithms for on-line multiple multicast in a reliable WDM network. We say that a routing problem is solvable if physical path(s) to realize the routing can be found and established. In off-line multiple multicast, all MT_i's are constructed one by one as described in Sect. 3.3. In the case of on-line multiple multicast, all groups of multicast are carried out concurrently, that is, all MT_i for $1\le i\le r$ are constructed concurrently, and multiple MT_i's may want to update the edge weight of each common edge they are using simultaneously. This can be resolved by imposing mutual exclusion to edge weight updating. However, doing so will make the above method for the off-line case not directly usable. The reason is that each step of extending each MT_i will update the edge weights at common edges and hence change the distances of many pairs of nodes in V_M, resulting in different edge weights of many edges in $I(\{s\}\cup D)$. Thus our main task here is to find an effective way to update the edge weights in $I(\{s\}\cup D)$ in correspondence to each edge weight update in G_M.

We observe that it is difficult to accomplish the above task if we use the same data structure as used in the off-line case because each edge weight in $I(\{s\}\cup D)$ corresponds to the accumulated path weight in G_M which is difficult to update with respect to an edge weight change. We therefore use an auxiliary graph G_I to represent $I(\{s\}\cup D)$. G_I is resulted by replacing each edge in $I(\{s\}\cup D)$ with its corresponding path in G_M, and its edge weight with the edge weight on the path in G_M. Let G_I^i be the G_I used for constructing MT_i, $1\le i\le r$. With the above replacement, any edge weight update resulted by path extension of MT_i in G_I^j will be immediately reflected in G_I^j for all $j\ne i$ and thus affect other paths extension of MT_j.

Our algorithm for constructing MT_i is given as follows [38].

Algorithm OLMMC
{*Multicast tree construction for $\mathcal{M}_i = (s_i, D_i)$, $D_i = \{t_i^1, \ldots, t_i^{g_i}\}$ in multiple multicast of r groups.*}

1. **for** each pair of $u, v \in \{s_i, t_i^1, t_i^2, \cdots, t_i^{g_i}\}$ **do**
   Compute the shortest path from u to v, $P[u \rightarrow v]$, in G_M;
   {*This shortest path will be used to connect u to v in G_I^i.*}
2. Construct "complete" graph G_I^i, where edge (u, v) is the path $P[u \rightarrow v]$ (with all edge weights preserved);
3. Compute a shortest path tree MT_i rooted at s_i reaching all destinations (D_i) in G_I^i, where for each new edge e added to ST_i do the following:
   {*Update the corresponding edge weight in all $G_I^{i'}$ mutual-exclusively.*}
   $wait(mutex)$; {*Mutual-exclusion, where $mutex = 1$ initially.*}
   For each $1 \leq i' \leq r$ if $t(e) \in V(G_I^{i'})$ then
   Add $\delta[j](t(e))$ to $d_{\lambda_j}((t(e))$ in $G_I^{i'}$ if e uses wavelength λ_j;
   $signal(mutex)$.

The correctness of the algorithm can be seen clearly from the greedy approach. We use an ordered data structure in node indices to store the nodes used in G_I^i, that is, use an array B_i of size $|V_M|$ initialized to 0 for G_I^i and add "1" to $B_i[j]$ "1" if an edge pointing to node j occurs in G_I^i. The time complexity of constructing G_I^i and updating edge weights in Steps 2 and 3 is $O(r^2 g_i^2)$, because adding each edge to MT_i requires to examine all r B_i's and update the weight of the edges pointing to nodes in B_i used in other groups when necessary, and each of this may need to wait for other groups updates in case of concurrent updating which brings in another r factor. Step 1 constructing shortest paths can be done in $O(g_i^2 k(kn + m + n\log(kn))) = O(g_i^2 k(kn + m + n\log(kn)))$ using Dijkstra's algorithm [9] (single source all destinations). Since $|MT_i| \leq |V_M|$, we have the following theorem [38].

Theorem 5: On-line multiple multicast for r groups in a WDM network of n nodes and m links with k available wavelengths can be completed in $O(g^2 k(r^2/k + kn+m+n\log(kn)))$ time, where g is the maximal group size.

6. On-Line Multiple Multicast in Unreliable Networks

In this section we consider the on-line communication problem in an unreliable WDM network in which both optical channel (wavelength) and switch gate (conversion) faults may occur. The basic idea to achieve fault-tolerance in multicast is to augment every edge in E_M with multiple alternative paths such that any MT constructed in G_M is always connected via at least one of these paths for any possible F. For multiple multicast, same as for the off-line case [35] since edge weight is shared, the above updating must also be mutually exclusive. Our algorithm is given as follows [38].

Algorithm FOLMMC
{*Establish physical paths for multicast tree MT_i for on-line multicast $\mathcal{M}_i$ found by Algorithm OLMMC in an unreliable WDM network with up to f faulty edges.*}
for $i = 1$ **to** $|E(MT)|$ **do**
  if $e_i \in F$ **then**
    $wait(mutex1)$; {*Mutual-exclusion, where $mutex = 1$ initially.*}
    Deduct $\delta[k](e_i)$ from $d_{\lambda_k}(e_i)$ if e_i uses wavelength λ_k;
    {*Reduce its queue length by 1 to reflect release of channel e_i for its future use.*}
    $signal(mutex1)$;
    $j := 1$; $alt := FALSE$;
    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 $MT^{(v)}$.*}
    $MT = MT \parallel \mathcal{P}(e)[j]$;
    $wait(mutex2)$;
    {*$mutex2 = 1$ initially.*}
    Add $\delta[k](t(e'))$ to $d_{\lambda_k}(e')$ for each $e' \in \mathcal{P}(e)[j]$ using wavelength λ_k;
    $signal(mutex2)$;
    {*Update MT and the edge weight for each edge on the new path.*}
  $wait(mutex3)$; {*$mutex3 = 1$ initially.*}
  For each $e \in \mathcal{R}(MT_i) - MT_i$ mark $w(e)$ with weight ∞;
  $signal(mutex3)$.

MT_i can be constructed by Algorithm OLMMC in time t_{MT}. Because each mutual exclusion for updating causes r^2 factor delay due to the reasons explained in Algorithm OLMMC, and inside the **for**-loop the computation takes $O(f^2 + |MT_i|) = O(f^2 + |V_M|)$ time, Algorithm **FOLMMC** requires $O(|E_{MT}|r^2(f^2 + |V_M|))$ time. With $|V_M| = 2kn$ and $|E_M| = k^2 n + km$, we have the following theorem [38]:

Theorem 6: On-line multiple multicast of r in an

unreliable WDM network with f faculty channels can be completed in $O(kr^2(f^2+kn)(kn+m))$ 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.

From the above discussion it is clear that multicast in an unreliable WDM network wit h f faulty edges requires $O(r^2(f^2+kn)/g^2t_{MC})$ time.

7. Group Membership Updating

On-line communication allows dynamic membership changes in the designated communication groups during the course of communication. In this section we present efficient algorithms for updating communication groups to accommodate dynamic group membership changes such as insertion and deletion of requests and destinations, group splitting and merging during the course of on-line multicast and multiple multicast. Our algorithms work for both reliable and unreliable WDM networks on the cost model (1).

7.1 Group Membership Updating for On-Line Multicast

We consider the problem of updating the group membership for on-line multicast where destination nodes can be dynamically inserted to or deleted from the multicast tree.

Assume that MT is the current multicast tree rooted at s and spans to all nodes in D. We use $p(v)$ to denote the precedent (parent) node of v in MT. When a node d is inserted to MT, we first compute $dist(u,d)$ and $dist(d,u)$ for every node $u \in MT$ in G_M, then we update MST with the path that has the minimal $dist(u,d)+dist(d,v)-dist(p(v),v)$. If $dist(d,v) \geq dist(p(v),v)$, we include path $u \rightarrow d$ into MT. Otherwise we include path $u \rightarrow d \rightarrow v$ and delete path $p(v) \rightarrow v$. Our algorithm is presented as follows:

Algorithm NodeInsert
{*Insert a new destination d to the multicast group.*}
1. For every node $u \in MT$ compute $dist(u,d)$ of path $P[u \rightarrow d]$ and $dist(d,u)$ of path $P[d \rightarrow u]$;
2. Compute $\min_{u \neq v \in MT}\{dist(u,d)+dist(d,v)-dist(p(v),v)\}$ and let the found nodes be u^* and v^*;
3. If $dist(d,v^*) \geq dist(p(v^*),v^*)$ then $MT = MT \cup P[u^* \rightarrow d]$
else $MT = (MT - P[p(v^*) \rightarrow v^*]) \cup P[u^* \rightarrow d] \cup P[d \rightarrow v^*]$.

In the case of deleting a destination from the multicast group, we compute the shortest cycle connecting the two parts of MT, $MT'(s)$ and MT'', that are disconnected due to removal of node d. That is, for all $u, u' \in MT'(s)$ and $v, v' \in MT''$ we compute $dist(u,v)$ and $dist(v',u')$ and take the minimal $dist(u,v)+dist(v',u')-dist(p(u'),u')$ to update MT. Below is our algorithm:

Algorithm NodeDelete
{*Delete a destination node d from the multicast group.*}
1. For every node $u \in MT'(s)$ and $v \in MT''$ compute $dist(u,v)$ and $dist(v,u)$;
2. Compute $\min_{(u,u' \in MT'(s)) \wedge (v,v' \in MT'')}\{dist(u,v)+dist(v',u')-dist(p(u'),u')$ and let these nodes be $u^*=u, u'^*=u', v^*=v, v'^*=v'$;
3. If $dist(v'^*,u'^*) \geq dist(p(u'^*),u'^*)$ then $MT = MT's \cup P[u^* \rightarrow v^*] \cup MT''$
else $MT = (MT'(s) - P[p(u'^*) \rightarrow u'^*]) \cup P[u^* \rightarrow v^*] \cup P[v'^* \rightarrow u'^*] \cup MT''$.

A direct implementation of the above algorithms requires clearly $O(|V_M||MT|^2)$ time, because there are $\Theta(|MT|^2)$ pairs of nodes and for each we need to compute two or three distances of shortest paths of length at most $|V_M|$. A more careful implementation suggests to precompute all-pairs shortest paths in G_M, which takes $O(|V_M|^3)$ time, and store them in a table for later retrieval. With this scheme, a distance can be obtained in $O(1)$ time by a table look-up, and therefore the total time for the above algorithms becomes $O(|MT|^2)$. Since $|MT| \leq |V_M| = (2k+1)n$, we have the following theorem.

We can also see that the resulting MT after updating has the same approximation ratio as the original MT. This is because the updating in both cases of insertion and deletion adds a minimum possible weight to incorporate the changes.

Theorem 7: [39] For on-line multicast in a WDM network of n nodes and k available wavelengths, dynamically inserting and deleting a destination requires $O(k^2n^2)$ time, preserving the same approximation ratio as the multicast tree before updating, provided that a precomputation of all-pairs shortest paths in G_M is given.

7.2 Group Membership Updating for On-Line Multiple Multicast

We now consider group membership maintenance for on-line multiple multicast in WDM networks.

When multiple multicast is carried out in on-line fashion, we are concerned with how to maintain MF with respect to the following dynamical changes:

(a) destinations may dynamically join and leave multicast groups,
(b) a group may be split into two (or more), with one (or more) of its destinations being a new source(s), and

(c) two (or more) groups may be merged together, where the source of a designated group becomes the common source of all the groups whereas sources of other groups become destinations.

For (a), we employ algorithms NodeInsert and NodeDelete of the previous section to dynamically maintain the multicast tree MT_i for each multicast group $\mathcal{M}_i$, where adding an edge to MT_i also updates the queuing delay at the corresponding edge accordingly for all i. Concurrent updates on queuing delay to the same edge are coordinated with a suitable synchronization mechanism. Time complexity for this case is at most r times of that required for on-line maintaining a single multicast tree due to the waiting time for edge weight updating, that is, $O(rk^2n^2)$ if some precomputation is done.

For (b), we reconstruct a multicast tree for each new group after splitting. Construction of different multicast trees are carried out concurrently without knowing each other using an on-line multicast tree construction algorithm described in Sect. 5.2. Concurrent updates to queuing delay are handled in the same way as in (a). The time complexity for this case is thus $O(g^2k(r^2/k + kn + m + n\log(kn))$ by the on-line multiple multicast time complexity [38], as each step of updating edge weight requires $O(r)$ time waiting for total r groups of multicast.

For (c), we need to merge two (or more) multicast trees MT_i and MT_j. This can be done by finding out the shortest path joining them into a single multicast tree rooted at the root of MT_i by the following algorithm.

Let $MT_i = \{s_i\} \cup D_i$ and $MT_j = \{s_j\} \cup D_j$, and s_i the designated root for $MT_i \cup MT_j$.

Algorithm GroupMerge

{*Merge multicast groups MT_i and MT_j into one group, with root s_i of MT_i being the common root.*}

1. For every node $u \in D_i$ and every node $v \in MT_j$ compute u^* and v^* such that

$$dist(u^*, v^*) + dist(v^*, s_j) - dist(s_j, v^*) = \min_{u \in D_i, v \in MT_j} \{dist(u, v) + dist(v, s_j) - dist(s_j, v)\};$$

Keep the corresponding path of $dist(x, y)$ in $P[x \to y]$;

2. If $v^* \neq s_j$ then merge $MT_i \cup MT_j \cup P[u^* \to v^*] \cup P[v^* \to s_j] - P[s_j \to v^*]$,
{*$dist(v^*, s_j) < dist(s_j, v^*)$.*}
else merge $MT_i \cup MT_j \cup P[u^* \to v^*]$.

end.

The time complexity for GroupMerge is $O(r|MT_i|\,|MT_j|) = O(rk^2n^2)$, where r is the factor for waiting time for each node updating.

Summarizing the above cases, we have the following theorem [39]:

Theorem 8: In on-line multiple multicast of r groups in a WDM network of n nodes and k available wavelengths with maximal group size g, a single group membership change and merging require $O(rk^2n^2)$ time, and a group splitting requires $O(g^2k(r^2/k + kn + m + n\log(kn))$ time.

With the time complexity of maintaining a single multicast tree $t_{MC} = O(k^2n^2)$ by Theorem 7, we know that a single group membership change and merge in multiple multicast of r groups require $O(rt_{MC})$ time, and a group splitting requires $O(\frac{g^2}{k}t_{MC})$ time.

8. QoS Extension

We now consider the problem of multicasting with quality of service (QoS) extension. In general, QoS can be any resource or timing constraints to be observed in the routing process. End-to-end delay has been widely regarded as an important criterion of QoS. We consider here the problem of constructing minimal cost multicast tree with bounded source-to-destination delay. Clearly, this problem is harder than the unconstrained multicast problem which is already NP-hard and therefore only approximate or heuristic solutions are feasible at present.

There have been numerous approaches proposed to solve this problem in different environments. Most proposed approaches to delay-bounded multicasting in WDM networks consist of two phases. In the first phase an approximate Steiner tree with minimal cost is constructed, which is used to produce a delay-bounded multicast tree in the second phase. The task in the second phase can be accomplished by identifying those paths whose delay exceeds the bound (violations) and replacing them with new ones that make the multicast tree observe the delay-bound [28], [45].

The solutions presented in the previous sections can be extended to incorporate with the QoS requirement of end-to-end bounded-delay by specifying a delay constraint and ensuring that the multicast tree under construction satisfies the delay constraint for each source-destination pair. To achieve the above, several heuristics may be employed. One method is that when a new edge is inserted into the multicast tree examine every source-destination pair in the tree whose path goes through the new edge to ensure that its delay does not exceed the bound. Another method is to first construct a multicast tree without considering the delay constraint, and then find all paths whose delay exceeds the bound and replace them with alternative ones with a bounded-delay.

There have been different models to calculate the end-to-end delay in WDM networks [28], [45]. All algorithms for multicasting with QoS guarantee under the delay constraint shall use the cost model (1) to calculate the cost and a suitable delay model of to calculate

the delay for each source-destination pair in the multicast tree. The bounded-delay constraint is then applied for each source-destination pair of concern when a new path is added to the multicast tree.

9. Concluding Remarks

We have given an overview on some recent results on multicasting in multi-hop optical WDM networks with limited wavelength conversion [35], [38], [39]. The contents covered in this paper include off-line and on-line routing in both reliable and unreliable networks for multicast and multiple multicast on a general cost model. For on-line routing, efficient algorithms for updating group membership to accommodate dynamic membership changes during the course of routing have also been presented. All the algorithms run efficiently in time polynomial to the network size and the number of wavelengths. Discussions on possible extension of these algorithm to provide QoS under the delay constraint have also been made.

References

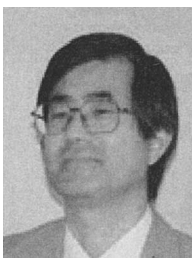
[1] A. Aggarwal, A. Bar-Noy, D. Coppersmith, R. Ramaswami, B. Schieber, and M. Sudan, "Efficient routing in optical networks," J. ACM, vol.46, pp.973–1001, 1996.
[2] Y. Aumann and Y. Rabani, "Improved bounds for all optical routing," Proc. 6th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'95), pp.567–576, 1995.
[3] R.A. Barry and P.A. Humblet, "On the number of wavelengths and switches in all-optical networks," IEEE Trans. Commun. (Part I), vol.42, no.2/3/4, 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. 2nd Workshop on Optics and Computer Science (WOCS), in CD-Rom, 1997.
[5] K.-M. Chan and T.-S. Yum, "Analysis of least congested path routing in WDM lightwave networks," Globecomm, pp.962–969, 1994.
[6] K.W. Cheung, "Scalable, fault tolerant 1-hop wavelength routing," Globecom'91, pp.1240–1244, 1991.
[7] I. Chlamtac, A. Faragó, and T. Zhang, "Lightpath (wavelength) routing in large WDM networks," IEEE J. Sel. Areas Commun., vol.14, no.5, 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. Commun., vol.40, no.7, pp.1171–1182, 1992.
[9] E.W. Dijkstra, "A note on two problems in connexion with graphs," Numerische Mathematik, vol.1, pp.269–271, 1959.
[10] V. Eramo and M. Listanti, "Comparison of unicast/multicast optical packet switching architectures using wavelength conversion," Optical Networks Magazine, vol.3, no.2, pp.18–26, March/April 2002.
[11] T. Erlebach and K. Jansen, "Scheduling of virtual connections in fast networks," Proc. 4th Workshop on Parallel Systems and Algorithms (PASA'96), pp.13–32, 1996.
[12] J.C. Gargano, P. Hell, and S. Perenees, "Colouring all directed paths in a symmetric tree with applications to WDM routing," Lecture Notes on Computer Science 1099 (Proceedings of ICALP'97), pp.505–515, 1997.
[13] L. Gargano, "Limited wavelength conversion in all-optical networks," Proc. 25th International Colloquium on Automata, Languages and Programming, pp.544–555, 1998.
[14] P.E. Green, Fiber-Optic Communication Networks, Prentice-Hall, 1992.
[15] K. Kaklamanis, G. Persiano, T. Erlebach, and K. Jansen, "Constrained bipartite edge coloring with applications to wavelength routing," Lecture Notes on Computer Science 1099 (Proc. of ICALP'97), pp.460–470, 1997.
[16] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for Steiner trees," Acta Informatica, vol.15, pp.141–145, 1981.
[17] K. Bharath-Kumar and J.M. Jaffe, "Routing to multiple destinations in computer networks," IEEE Trans. Commun., vol.COM-31, no.3, pp.343–351, 1983.
[18] E. Kumar and E. Schwabe, "Improved access to optical bandwidth in trees," Proc. 8th Annual ACM-SIAM Symp. on Discrete Algorithms (SODA'97), pp.437–44, 1997.
[19] H.-M. Lee and G.J. Chang, "Set-to-set broadcasting in communication networks," Discrete Applied Mathematics, vol.40, pp.411–421, 1992.
[20] K. Li, Y. Pan, and S.Q. Zheng, eds., Parallel Computing Using Optical Interconnections, Kluwer Academic Publishers, 1998.
[21] W. Liang, G. Havas, and X. Shen, "Improved lightpath routing in large WDM networks," To appear in Proc. 18th Intern. Conf. on Distributed Computing Systems, Amsterdam, The Netherlands, IEEE Computer Society Press, pp.516–523, May 1998.
[22] W. Liang and H. Shen, "Multicast broadcasting in large WDM networks," Proc. 12th Intern. Conf. on Parallel Processing Symp. (IPPS/SPDP), Orlando, Florida, USA, IEEE Computer Society Press, 1998.
[23] R. Malli, X. Zhang, and C. Qiao, "Benefit of multicasting in all-optical WDM networks," Conf. on All-optical Networks, SPIE, vol.3531, pp.209–220, 1998.
[24] G. De Marco, L. Gargano, and U. Vaccaro, Concurrent multicast in weighted networks, manuscript.
[25] A.D. McAulay, Optical Computer Architectures, John Wiley, 1991.
[26] M. Mihail, K. Kaklamanis, and S. Rao, "Efficient access to optical bandwidth," Proc. FOCS'95, pp.548–557, 1995.
[27] B. Mukherjee, "IEEE Commun. Mag., vol.37, no.1/2, Jan/Feb 1999.
[28] Y. Ofek and B. Yener, "Reliable concurrent multicast from bursty sources," Proc. IEEE INFOCOM'96, pp.1433–1441, 1996.
[29] P. Raghavan and E. Upfal, "Efficient routing in all-optical networks," Proc. STOC'94, pp.133–143, 1994.
[30] R. Ramaswami, "Multi-wavelength lightwave networks for computer communication," IEEE Commun. Mag., vol.31, no.2, pp.78–88, 1993.
[31] G.N. Rouskas and M.H. Ammar, "Analysis and optimization of transmission schedules for single-hop WDM networks," Infocom'93, pp.1342–149, 1993.
[32] G.N. Rouskas and M.H. Ammar, "Multi-destination communication over tunable-receiver single-hop WDM networks," TR-96-12, Department of Computer Science, North Carolina State University, 1996.
[33] L.H. Sahasrabuddhe and B. Mukherjee, "Light-trees: Optical multicasting for improved performance in wavelength-routed networks," IEEE Commun. Mag., vol.37, no.2, pp.67–73. 1999.
[34] H. Shen, "Efficient multiple multicasting in hypercubes," J. System Architectures, vol.43, no.9, pp.655–662, 1997.
[35] H. Shen, F. Chin, and Y. Pan, "Efficient fault-tolerant routing in multihop optical networks," IEEE Trans. Parallel and Distributed Systems, vol.10, no.10, pp.1012–1025, 1999.

[36] H. Shen and W. Liang, "Efficient multiple multicast in WDM networks," Proc. 1998 Intern. Conf. on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA, 1998.
[37] H. Shen, Y. Pan, and S. Horiguchi, "Routing in multihop optical WDM networks with limited wavelength conversion," in Optical Switching/Networking and Computing for Multimedia Systems, ed. M. Guizani and A. Battou, pp.217–248, Marcel-Dekker, New York, 2002.
[38] H. Shen, J. Sum, G.H. Young, and S. Horiguchi, "Efficient algorithms for on-line communication in WDM networks," Proc. 2000 Intern. Conf. on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA, 2000.
[39] H. Shen, J. Sum, G.H. Young, and S. Horiguchi, "Efficient dynamic group membership updating for on-line communication in optical WDM networks," Proc. 2000 Intern. Conf. on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA, 2000.
[40] Silbertcharze, Operating Systems Concepts, Addison Wesley, 1998.
[41] R.J. Vitter and D.H.C. Du, "Distributed computing with high-speed optical networks," IEEE Computer, vol.26, no.1, pp.8–18, 1993.
[42] S.S. Wagner and H. Kobrinski, "WDM applications in broadband telecommunication networks," IEEE Commun. Mag., vol.27, no.3, pp.22–30, 1989.
[43] Z. Zhang and A.S. Acampora, "A heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength re-use," IEEE/ACM Trans. Networking, vol.3, no.3, pp.281–288, 1995.
[44] J. Zheng and H.T. Mouftah, "Distributed lightpath control based on destination routing in wavelength-routed WDM networks," Optical Networks Magazine, vol.3, no.4, July/Aug. 2002.
[45] T.F. Znati, T. Alrabiah, and R. Melhem, "Low-cost, bounded-delay multicast routing for QoS-based networks, Computer Networks, vol.38, no.4, pp.423–445, 2002.

Hong Shen received his B.Eng. degree from Beijing University of Science and Technology, M.Eng. degree from University of Science and Technology of China, Ph.Lic. and Ph.D. degrees from Abo Akademi University, Finland, all in Computer Science. He is currently a full professor in the Graduate School of Information Science, Japan Advanced Institute of Science and Technology. Previously he was a professor in Griffith University, Australia. Dr. Shen has published over 140 technical papers on algorithms, parallel and distributed computing, interconnection networks, parallel databases and data mining, multimedia systems and networking. He has served as an editor of Parallel and Distributed Computing Practice, associate editor of International Journal of Parallel and Distributed Systems and Networks, editorial-board member of Parallel Algorithms and Applications, International Journal of Computer Mathematics and Journal of Supercomputing, and chaired various international conferences. Dr. Shen is a recipient of 1991 National Education Commission Science and Technology Progress Award and 1992 Sinica Academia Natural Sciences Award.

Yi Pan received his B.Eng. and M.Eng. degrees in computer engineering from Tsinghua University, China, in 1982 and 1984, respectively, and his Ph.D. degree in computer science from the University of Pittsburgh, USA, in 1991. Currently, he is an associate professor in the Department of Computer Science at Georgia State University. His research interests include parallel and distributed computing, optical networks and wireless networks. His pioneer work on computing using reconfigurable optical buses has inspired extensive subsequent work by many researchers. He is a co-inventor of three U.S. patents (pending) and several provisional patents. He has published more than 120 research papers including over 50 journal papers (more than 20 of which have been published in various IEEE journals) and received many awards from agencies such as NSF, AFOSR, JSPS, IISF and Mellon Foundation. His recent research has been supported by NSF, AFOSR, AFRL, JSPS, and the states of Georgia and Ohio. Dr. Pan is currently serving as an editor-in-chief or editorial board member for 7 journals including 3 IEEE Transactions. He has also served as a program or general chair for several international conferences and workshops. Dr. Pan has delivered over 40 invited talks, including keynote speeches and colloquium talks, at conferences and universities worldwide. Dr. Pan is an IEEE Distinguished Speaker (2000–2002), a Yamacraw Distinguished Speaker (2002), a Shell Oil Colloquium Speaker (2002), and a senior member of IEEE. He is listed in Men of Achievement, Who's Who in Midwest and Who's Who in America.

John Sum received his BEng in Electronic Engineering from the Hong Kong Polytechnic University in 1992. He received his MPhil and PhD degrees from Department of Computer Science and Engineering of the Chinese University of Hong Kong in 1995 and 1998 respectively. He has published many journal and conference papers. In 1992, John received the Hong Kong Institution of Engineers Student Prize in recognition of his academic merit, extra-curricular activities, leasdership qualities and personality during his undergraduate study. He has also received two times the Certificate of Merits from the Department of Computer Science and Engineering of CUHK in appreciation for his excellent teaching assistantship, in 1996 and 1997 respectively. Dr.Sum has been selected for inclusion in the Year 2000 Marquis Who's Who in the World. He has a broad research interest including Web Commerce, Information Economy and some others. Since 1999, John has also been affiliated with the Laboratory of Internet Software Technologies, Institute of Software, Chinese Academy of Science, working on the analysis and design of an Internet platform for procuring over active supply chain. Currently, he is serving as a advisory board member in various MNC providing technological consultant service.



Susumu Horiguchi graduated from Department of Communication Engineering, Tohoku University in 1976, and received the MS. and Dr. degrees both from the same university in 1978 and 1981, respectively. He was a faculty of Department of Information Science at Tohoku University from 1981 to 1992. He was a visiting scientist of IBM Thomas J. Watson Research Center from 1986 to 1987 and a visiting professor of The Center for Advanced Studies at the University of Southwestern Louisiana and Department of Computer Science, Texas A&M University summer in 1994 and 1997. Since 1992, he has been a full Professor of the Graduate School of Information Science at JAIST(Japan Advanced Institute of Science and Technology). He has been conducting his research group as the chair of Multi-Media Integral System Laboratory at JAIST. He has been involved in organizing many international workshops, symposia and conference sponsored by IEEE, IASTED, IEICE and IPS. His research interest has been mainly concerned with optical switch interconnection, interconnection networks, GRIDs computing, parallel computer architecture, and VLSI/WSI architecture. Dr.Horiguchi is a senior member of IEEE Computer Society, and members of IPS and IASTED.