Optimization of Wavelength Assignment for QoS Multicast in WDM Networks

Xiao-Hua Jia, Ding-Zhu Du, Xiao-Dong Hu, Man-Kei Lee, and Jun Gu, *Senior Member, IEEE*

***Abstract*—This paper discusses quality-of-service (QoS) multicast in wavelength-division multiplexing (WDM) networks. Given a set of QoS multicast requests, we are to find a set of cost suboptimal QoS routing trees and assign wavelengths to them. The objective is to minimize the number of wavelengths in the system. This is a challenging issue. It involves not only optimal QoS multicast routing, but also the optimal wavelength assignment. Existing methods consider channel setup in WDM networks in two separate steps: routing and wavelength assignment, which has limited power in minimizing the number of wavelengths. In this paper, we propose a new optimization method, which integrates routing and wavelength assignment in optimization of wavelengths. Two optimization algorithms are also proposed in minimizing the number of wavelengths. One algorithm minimizes the number of wavelengths through reducing the maximal link load in the system; while the other does it by trying to free out the least used wavelengths. Simulation results demonstrate that the proposed algorithms can produce suboptimal QoS routing trees and substantially save the number of wavelengths.**

***Index Terms*—Light-tree, QoS multicast, routing, wavelength assignment, WDM network.**

I. Introduction

ADVANCES in electro-optic technologies have made optical communication a promising network choice to meet the increasing demands for high channel bandwidth and low communication latency of network applications. *wavelength-division multiplexing* (WDM) [1] is basically frequency-division multiplexing in the optical frequency domain, where on a single optical fiber there are multiple communication channels at different wavelengths. There are two types of architectures of WDM optical networks: *single-hop* systems and *multihop* systems [2]. In single-hop systems, a communication channel should use the same wavelength throughout the route of the channel; while in multihop systems, a channel can consist of multiple lightpaths (a *lightpath,* for point to point communication, is a path that occupies the same wavelength throughout the path) and wavelength conversion is allowed at the joint nodes of two lightpaths in the channel. In this paper, we consider single-hop systems, since all-optical wavelength conversion is still an immature and expensive technology.

Multicast is a point to multipoint communication, by which a source node sends messages to multiple destination nodes. Multicast uses a tree structure as the route to transmit messages to multiple destinations. A light-tree concept was proposed in [3] to support multicast in optical networks. A *light-tree*, as a point to multipoint extension of a lightpath, is a tree in the physical topology and occupies the same wavelength in all fiber links in the tree. Each fork node of the tree is a multicast-capable optical switch, where a power splitter [4] is used to split an input optical signal into multiple signals which are then forwarded to output ports without electrical conversions. Hence, light-trees facilitate single-hop point to multipoint communications (i.e., multicast).

End-to-end delay is an important quality-of-service (QoS) parameter in data communications. QoS multicast requires that the delay of messages from the source to any destination be within a bound. There are many network applications nowadays which need the support of QoS multicast, such as multimedia conferencing systems, video on demand systems, real-time control systems, and so on. In this paper, we consider the routing and wavelength assignment problem for a set of QoS multicast requests in a system. The problem is formalized as follows: given a set of QoS multicast requests in a WDM network system, compute a set of QoS routing trees and assign wavelengths to them. The objective is to minimize the number of distinct wavelengths to be used under the following constraints on each routing tree:

- the delay from the source to any destination along the tree does not exceed a given bound;
- the total cost of the tree is suboptimal.

A new optimization method for wavelength assignment is proposed, which integrates routing and wavelength assignment by using rerouting and reassigning techniques. Simulation results demonstrate that the proposed method significantly reduces the number of (distinct) wavelengths used in the system, and the generated routing trees guarantee the QoS constraint and have less cost.

The traffic model discussed in this paper is static. The result of this research can be used as a guideline for configuration of a WDM network, such as the configuration of lightpaths or light-trees. The traffic type discussed in this paper (i.e., QoS multicast traffic) is a general one. It allows us to describe and consider all different types of traffic, such as unicast traffic, multicast traffic, QoS traffic, and non-QoS traffic, in a uniform way, because they are special cases of QoS multicast traffic. An "optimal" solution to our proposed problem is applicable

Paper approved by D. K. Hunter, the Editor for Optical Communications of the IEEE Communications Society. Manuscript received December 8, 1999; revised June 29, 2000 and July 18, 2000. This work is supported in part by the City University of Hong Kong under Grant 7000677, Grant 7000927, and Grant 7001035, by the CERG of Hong Kong under Grant 9040442, and by the National 973 Information Technology and High-Performance Software Program of China under Grant G1998030402.

X.-H. Jia and M.-K. Lee are with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong.

D.-Z. Du is with the Department of Computer Science, University of Minnesota, Minneapolis, MN 55455 USA.

X.-D. Hu is with Institute of Applied Mathematics, Chinese Academy of Sciences, Beijing 100080, China.

J. Gu is with the Department of Computer Science Hong Kong University of Science and Technology, Kowloon, Hong Kong.

Publisher Item Identifier S 0090-6778(01)01291-0.

to other single-hop or light-tree based systems. Another significant aspect of this work is the discovery of the effectiveness of wavelength optimization methods in multicast domain. We have found that the optimization via reassigning the least used wavelength works more effectively than the optimization via traffic load-balancing.

The rest of the paper is organized as follows. Section II discusses the network model, the problem definition, and related work. A set of (re)routing and wavelength (re)assignment algorithms are proposed in Section III. Simulation results and performance analysis are discussed in Section IV. Finally, a summary of results is made in Section V.

II. Preliminaries

A. System Models and Problem Specification

The WDM network is modeled by an undirected and connected graph $G(V, E, c, d)$, where V is the vertex-set of G representing the set of nodes in the network with $|V| = n$, and E is the edge-set of G with $|E| = m$ corresponding to fiber links between nodes in the network. Each link carries two oppositely-directed fibers for data transmissions in the two directions of the link. Each edge $e \in E$ is associated with two weight functions $c(e)$ and $d(e)$, where $c(e)$ represents the communication cost of e and $d(e)$ the delay of e. $c(e)$ can be interpreted as the monetary cost of using link e. $d(e)$ represents the transmission time of a packet between the two endpoints of link e, which includes switching and propagation delays. Both functions c and d are additive over the links of a path $P(u, v)$ between two nodes u and v, i.e.,

$$c(P(u,v)) \equiv \sum_{e \in P(u,v)} c(e) \quad \text{and}$$
$$d(P(u,v)) \equiv \sum_{e \in P(u,v)} d(e).$$

We consider k bidirectional QoS multicast requests in the system, denoted by $\{r_i(s_i, D_i, \Delta_i) | i = 1, 2, \ldots, k\}$. Each request, $r_i(s_i, D_i, \Delta_i)$, is for setting up a QoS multicast channel from source node s_i to a group of destination nodes D_i ($s_i \notin D_i$) and the data transmission delay from s_i to any node in D_i should be within bound Δ_i. The actual value of Δ_i depends on the application. We are to find the QoS routing trees (one for each of the QoS multicast requests) and assign wavelengths to them.

Let $T_i(s_i, D_i, \Delta_i)$ be the routing tree for request $r_i(s_i, D_i, \Delta_i)$. When multicasting a message from source s_i to D_i along tree $T_i(s_i, D_i, \Delta)$, the light signal is split at s_i and forwarded to the output ports leading to its children, which then transmit the signal to their children until all nodes in the tree (thus, all nodes in D_i) receive it. Based on the splitting mechanism, optical multicasting has some improved characteristics over electronic multicast, since "splitting light" is conceptually easier than copying a packet in an electronic buffer. We assume an optical signal can be split into an arbitrary number of optical signals at a switch. Thus, there is no restriction on node degree in a routing tree.

The QoS requirement of routing tree $T_i(s_i, D_i, \Delta_i)$ is that the delay from s_i to any nodes in D_i should not exceed Δ_i. Let $P_{T_i}(s_i, u)$ denote the path in $T_i(s_i, D_i, \Delta_i)$ from s_i to $u \in D_i$. This requirement can be formally represented as

$$\begin{aligned} d(P_{T_i}(s_i,u)) &= \sum_{e \in P_{T_i}(s_i,u)} d(e) \\ &= d(P_{T_i}(s_i,u)) \leq \Delta_i \qquad \forall\, u \in D_i. \end{aligned} \tag{1}$$

In order to make Δ_i valid for a request r_i, it must satisfy:

$$\Delta_i \geq d(P_G(s_i, u)), \qquad \forall\, u \in D_i \tag{2}$$

where $P_G(s_i, u)$ is the shortest path from s_i to u in G.

Since every node in a routing tree is multicast-capable, a message flows through each branch of the tree once and only once to reach all the destinations. Therefore, the cost of multicasting a message, called the cost of the tree, is the sum of the cost of all links in the tree. It can be formally defined as

$$c(T_i(s_i, D_i, \Delta_i)) \equiv \sum_{e \in T_i(s_i, D_i, \Delta_i)} c(e).$$

One objective of multicast routing is to construct a routing tree which has the minimal cost. This problem is regarded as the (minimum) Steiner tree problem [5], which was proved to be NP-hard [6]. We will use a heuristic algorithm to find suboptimal routing trees that meet the QoS requirement (1). A routing tree is called *suboptimal* if its cost is not bigger than a small constant times that of the optimal routing tree (regardless of the QoS requirement).

Another important objective of routing and wavelength assignment in WDM networks is to minimize the number of (distinct) wavelengths used in the system, because the number of wavelengths that could be used in a system is limited. In a single-hop WDM system, two channels must use different wavelengths if their routes share a common link, which is the wavelength conflict rule.

The problem we consider in this paper is, for a given set of QoS multicast requests, to construct a set of suboptimal QoS routing trees, and assign wavelengths to them so that the number of wavelengths used is minimized.

B. Related Work

In the past, QoS multicast in the electronic domain has been extensively studied. Many QoS multicast routing methods have been proposed [7]–[10]. A suite of centralized multicast routing algorithms were proposed in [7] for trading off between cost and delay functions. Some distributed algorithms were proposed in [8] and [10] that generate delay-bounded suboptimal routing trees. The method proposed in [8] can be integrated with the operation of connection configurations. A broadcast version of the problem was discussed in [9], where an efficient algorithm was proposed to construct a spanning tree in which the distance from the root to any node is at most a constant times the shortest-path distance in the graph and the total cost of the tree is at most the same constant times the cost of minimum spanning tree. The

```
Input A request $r_i(s_i, D_i, \Delta_i)$, and a graph $G(V, E, c, d)$
Output A sub-optimal QoS routing tree $t_A$ if $t_A \neq \emptyset$ (otherwise $t_A = \emptyset$)
Step 0 Initialization
        $t_A := \emptyset$.
Step 1 Construct a low cost routing tree
        produce a tree $t_A$ by employing a heuristic for the Steiner tree problem under function $c$.
Step 2 Modify $t_A$ into a QoS routing tree
        while $t_A$ does not meet delay requirement (1) for some $u \in D_i$ do begin
            add the shortest path from $s_i$ to $u$ under distance function $d$ into $t_A$,
            remove redundant edges in $t_A$ to keep it a tree structure.
            if $t_A$ still does not meet delay requirement (1) then
              return $t_A = \emptyset$.
        end-while
        return $t_A$.
```

Fig. 1. Algorithm A: QoS routing.

```
Input A set of routing trees $\{\, t(r_i) \mid i = 1, 2, \cdots, k \,\}$
Output A set of routing trees with assigned wavelengths $\{\, t(r_i) \mid i = 1, 2, \cdots, k \,\}$
Step 1 Create an auxiliary graph $G_a(V_a, E_a)$
        $V_a := \{\, t(r_i) \mid i = 1, 2, \cdots, k \,\}$,
        $E_a := \{\, (t(r_i), t(r_j)) \mid t(r_i) \cap t(r_j) \neq \emptyset, i \neq j \,\}$.
Step 2 Assign wavelengths to vertices in $V_a$ sequentially
        $V' := V_a$.
        while $V' \neq \emptyset$ do begin
            choose $t(r_j) \in V'$ which has the least degree,
            $V_j := \{\, t(r_i) \in V' \mid (t(r_j), t(r_i)) \notin E_a \}$, // $V_j$: set of vertices not adjacent to $t(r_j)$
            find a maximal set $V_j^{max} \subseteq V_j$ with no edge between any pair of vertices in $V_j^{max}$,
            assign all vertices in $V_j^{max}$ a wavelength, // vertices (trees) in $V_j^{max}$ share no links
            $V' := V' \setminus V_j^{max}$.
        end-while
        return $\{\, t(r_i) \mid i = 1, 2, \cdots, k \,\}$ with assigned wavelengths.
```

Fig. 2. Algorithm B: wavelength assignment.

cost and distance functions considered in this work are assumed to be identical.

Recently, multicast routing in optical networks has been researched. It was studied in [4] concerning the design of multicast-capable optical switches. In [3], the concept of light-trees was introduced for multicast in WDM networks. The objectives of setting up the light-trees in [3] are to minimize the network-wide hop distance and to minimize the total number of transceivers used in the network. In [11], an asymptotic upper bound on the number of wavelengths needed to support multicast capability was obtained by using some properties of expander graphs.

Our problem is to route a set of QoS multicast channels and assign wavelengths to them, aiming at minimizing the number of wavelengths used. Two optimization problems are related with our problem: *load balancing problem* (LBP) and *wavelength assignment problem* (WAP). LBP aims at routing a set of communication channels so that the maximal link load in a system is minimized. The load on a link is defined as the number of channels over the link. Because the number of wavelengths required is at least equal to the maximal link load in a system, load balance routing can reduce the number of wavelengths to be used. WAP is to assign wavelengths to a set of routes so that the number of wavelengths required can be minimized.

Most of the existing work on LBP/WAP is for unicast communications. It was proved in [12]–[14] that LBP/WAP is NP-complete even in simple networks such as rings and trees. The existing methods for channel setup in WDM networks, such as in [2], [14], usually take the following two separate steps: 1) generate optimal routes which minimize the maximal link load (i.e., load balance routing) and 2) assign wavelengths to them by using minimal number of wavelengths (i.e., WAP). The load balance routing is usually formulated into an integer linear program (ILP) [15], [16], which is NP-hard [6]. The ILP is then transformed into a linear program (LP) by relaxing the integral constraint, which can be solved efficiently. In the end, an (integral) solution of ILP is obtained from the solution of LP through rounding techniques [17]. WAP is usually transformed into a vertex coloring problem, which is again NP-complete [5]. Heuristic methods can be employed to find approximation solutions.

It would be very complicated to apply the techniques for unicast to the case of multicast. In this paper we propose a new approach that integrates routing with wavelength assignment by using rerouting and wavelength reassignment. It is different from the existing methods, which consider routing and wavelength assignment separately.

III. Rerouting Algorithms

A. Overview of the Proposed Algorithms

In this section, we will first introduce algorithm A for QoS routing (Fig. 1) and algorithm B for wavelength assignment (Fig. 2). Then we propose two optimization algorithms, algorithm C (Fig. 3) and D (Fig. 4), aiming at minimizing the number of wavelengths over the results produced by algorithms A and B. Since the number of wavelengths required is greater than or equal to the maximal link load, algorithm C reroutes some of the routing trees to reduce the maximal link load by

```
Input A set of routing trees $\{ t(r_i) \mid i = 1, 2, \cdots, k \}$
Output A set of re-routed trees $\{ t(r_i) \mid i = 1, 2, \cdots, k \}$ with balanced load
Step 1 Calculate the load on each link
        $L(e) := \{ t(r_i) \mid e \in t(r_i) \}$, for $e \in E$,
        $l_{max} := \max\{ |L(e)| \mid e \in E \}$.
Step 2 Re-route the trees to balance the load
        $E_{max} := \{ e \in E \mid l(e) = l_{max} \}$.
        while $E_{max} \neq \emptyset$ do begin
            choose an edge $e_{max} \in E_{max}$.
            $R := \{ r_i \mid e_{max} \in t(r_i) \}$.
            // re-routing any tree in $R$ successfully will reduce load on $e_{max}$ by 1
            while $R \neq \emptyset$ do begin
                choose an $r_i \in R$,
                $j := l_{max} - 1$.
                while $j \geq 1$ do begin // re-route $r_i$ on the most lightly loaded links
                    $E_j := E \setminus \{ e \mid |L(e)| \geq j \}$, // avoid using links whose load $\geq j$
                    run algorithm A for $r_i$ on $G(V, E_j, c, d)$.
                    if $t_A \neq \emptyset$ then // re-routing $r_i$ on $E_j$ succeeds
                        $t(r_i) := t_A$,
                        $j := j - 1$. // try to re-route $r_i$ on $E_{j-1}$
                    else exit while-$j$. // cannot re-route $r_i$ on lighter links
                end-while-$j$
                if $j = l_{max} - 1$ then // re-routing $r_i$ fails
                    $R := R \setminus \{r_i\}$, // try to re-route another tree in $R$
                else $E_{max} := E_{max} \setminus \{e_{max}\}$, // load on $e_{max}$ is reduced by 1
                exit while-$R$. // work on another edge in $E_{max}$
            end-while-$R$
            if $R = \emptyset$ then // none of trees in $R$ can be re-routed
                stop. // the maximal link load cannot be reduced any further
        end-while-$E_{max}$
        if $E_{max} = \emptyset$ then // the maximal link load was reduced by 1
            go to Step 1. // reduce the maximal link load further
```

Fig. 3. Algorithm C: optimization through load balancing.

```
Input A set of routing trees $\{ t(r_i) \mid i = 1, 2, \cdots, k \}$ with assigned wavelengths
Output A set of (re)routed trees $\{ t(r_i) \mid i = 1, 2, \cdots, k \}$ with re-assigned wavelengths
Step 1 Calculate the distribution of currently used wavelengths
        $T(w) := \{ t(r_i) \mid t(r_i)$ is assigned wavelength $w \}$, for $w \in W$,
        sort $|T(w)|$ in ascending order, i.e., $|T(w_1)| \leq |T(w_2)| \leq \cdots \leq |T(w_{|W|})|$.
Step 2 Re-route the trees that are assigned the least used wavelength
        while $T(w_1) \neq \emptyset$ do begin // re-route all trees in $T(w_1)$ to save $w_1$
            choose $t(r_i) \in T(w_1)$,
            $j := |W|$.
            while $j > 1$ do begin // re-route $t(r_i)$ to use wavelength $w_j$
                $E_j := E \setminus \{T(w_j)\}$, // remove $T(w_j)$ from network graph $G$
                run algorithm A for $r_i$ on $G(V, E_j, c, d)$.
                if $t_A \neq \emptyset$ then // re-routing $r_i$ succeeds and $r_i$ can use $w_j$
                    $t(r_i) := t_A$,
                    assign $r_i$ wavelength $w_j$, // because $t(r_i)$ disjoint with $T(w_j)$
                    $T(w_1) := T(w_1) \setminus \{t(r_i)\}$;
                    exit while-$j$. // re-route another tree in $T(w_1)$
                else $j := j - 1$. // try to use another wavelength $w_{j-1}$
            end-while-$j$
            if $j = 1$ then // $r_i$ cannot use any other wavelength but $w_1$
                stop.
        end-while-$T(w_1)$
        if $T(w_1) = \emptyset$ then // wavelength $w_1$ can be saved
            $W := W \setminus \{w_1\}$,
            go to Step 1. // try to save another wavelength in $W$
```

Fig. 4. Algorithm D: optimization through wavelength reassignment.

avoiding use of the links whose load is the maximum. In order to reduce the number of wavelengths in wavelength assignment, algorithm D reroutes the trees whose wavelengths are the least used, which tries to free out the least used wavelengths. The two optimization algorithms C and D are used together with the two basic algorithms A and B.

B. Algorithm A for QoS Routing

For each QoS multicast request, $r_i(s_i, D_i, \Delta_i)$, algorithms A constructs a suboptimal QoS routing tree (see Fig. 1). It first generates a low cost routing tree by applying a heuristic for the Steiner tree problem, and then modifies this tree into the one which meets the QoS requirement by checking delay requirement (1) for all destinations. If the delay requirement is not met for a destination, the shortest path in terms of delay function d from the source to the destination will replace the corresponding tree path linking the destination.

In Step 1), we use an MST-based heuristic [7] to generate a routing tree for request $r(s_i, D_i, \Delta_i)$. Instead of producing an MST of $G(V, E, c, d)$ which spans all nodes in V, this algorithm, 1) generates an edge-weighted complete graph G' on $\{s_i\} \cup D_i$ such that the weight of edge (u, v) is the cost of the shortest path between u and v in terms of function c in graph G; 2) produces an MST of graph G'; and 3) obtains tree t_A in G by substituting each edge of the MST in G' with corresponding path in G. Refer to [7] for more details.

In Step 2), we apply a depth-first search [18], [9] on t_A as follows: traverse an edge-by-edge walk from s_i through the nodes of t_A. Each edge is traversed twice, once in each direction. If node u in D_i is visited the first time and the delay requirement is not met, the shortest path from s_i to u on G in terms of function d is added into t_A. If the delay requirement still cannot be met for node u after the modification, then algorithm A returns $t_A = \emptyset$.

Let t_M be the routing tree on $G(V, E, c, d)$ produced in Step 1) of Algorithm A. It was proved in [7] that $c(t_M) \le 2 \cdot c(t_{\text{opt}})$, where t_{opt} is the routing tree of minimum cost. Furthermore, for the final routing tree produced at Step 2), we can prove that its cost is at most a constant times of $c(t_M)$ (thus a constant times of $c(t_{\text{opt}})$).

Theorem 1: Suppose that the delay function d is proportional to the cost function c. For each QoS multicast request $r_i(s_i, D_i, \Delta_i)$, algorithm A can output QoS routing tree t_A which satisfies

$$c(t_A) \le \beta \cdot c(t_M), \qquad \text{for } \beta \ge 1 + \frac{2}{\alpha - 1} \tag{3}$$

and $\alpha \equiv \min\{(\Delta_i / d(P_G(s_i, u)))|u \in D_i\} > 1$.

Proof: The theorem can be proved by applying the same argument in [9] to the complete graph G' on $\{s_i\} \cup D_i$. A detailed proof is included in the Appendix. □

The inequality (3) says that for each node $u \in D_i$, the delay from s_i to u along t_A is at most α times the delay of the shortest path from s_i and u in graph G, that is

$$d(t_A) \le \alpha \cdot d(P_G(s_i, u)) \qquad \forall\, u \in D_i \tag{4}$$

and the cost of t_A is at most β times the cost of t_M. The relationship between ratios α and β indicates that reducing the network cost is at the expense of increasing the delay, and likewise for reducing the delay. There is always a tradeoff between cost and delay.

Theorem 2: Algorithm A can output a QoS routing tree in time $O(|D_i||V|^2)$.

Proof: Since delay requirement (1) is enforced in the execution of algorithm A, a QoS routing tree can be produced (if there exists one). In Step 1), it takes time $O((|D_i| + 1|)|V|^2)$ to generate a shortest path in G between each pair of nodes in $\{s_i\} \cup D_i$, and $O((|D_i| + 1)^2)$ to construct an MST in complete graph G'. In Step 2), the depth first search can be done in $O(|V|)$. Therefore, the time-complexity of algorithm A is $O(|D_i||V^2)$. □

C. Algorithm B for Wavelength Assignment

To assign wavelengths to a set of routing trees without causing wavelength conflict, we introduce an auxiliary graph G_a, where each vertex in G_a represents a routing tree and there is an edge between two vertices in G_a if and only if the two routing trees share a common link in G. Assigning wavelengths to the trees is reduced to the problem of coloring all vertices in G_a such that no two adjacent vertices receive the same color (because two adjacent vertices in G_a indicate that their corresponding routing trees share a common link in G). Minimizing the use of colors in this coloring problem is NP-complete [6]. We use a simple heuristic based on the sequential coloring approach in [19]. This coloring method 1) chooses a vertex which has the least degree, 2) finds a maximal set of vertices that are not adjacent to the selected vertex and there is no edge between any pair of vertices in the set, 3) assigns a wavelength to the vertices in this set and removes them from the graph, and 4) repeats this process until all vertices are colored and removed.

Theorem 3: Algorithm B can assign wavelengths to a given set of routing trees $\{t(r_i)|i = 1, 2, \ldots, k\}$ in time $O(k^2|V|^2)$, where $K = \max\{|D_i||i = 1, 2, \ldots, k\}$.

Proof: At Step 1), $G_a(V_a, E_a)$ can be produced in time $O(k^2|V|^2)$, because $t(r_i)$ has $(|V| - 1)$ edges for $i = 1, \ldots, k$. At Step 2), every given routing tree is assigned a wavelength, and two trees are assigned the same wavelength if and only if they do not share any edge. In addition, choosing a vertex of the least degree and finding a maximal set both can be done by checking adjacency among vertices in V_a. Thus the wavelength assignment can be finished in $O(k^2|V|^2)$. □

D. Algorithm C: Optimization through Load Balancing

Given a set of routing trees, algorithm C minimizes the number of wavelengths by reducing the maximal link load in the system. It will 1) calculate the load on each link and 2) choose a tree which contains the links having the maximum load, and then reroute it by running algorithm A on the subgraph of $G(V, E, c, d)$ after removing the links having the maximum load. The rerouting operation is repeated until the maximal link load cannot be reduced any further.

At Step 2), in the while-loop of j we search for the smallest j such that the tree of $r_i \in R$ can be rerouted on the most lightly loaded edges. This can be considered as a greedy strategy for LBP.

Theorem 4: Algorithm C can finish rerouting a given set of routing trees in time $O(((K|V|l_{\max})^2 + k|E|)l_{\max})$, where $K = \max\{|D_i||i = 1, 2, \ldots, k\}$.

Proof: Step 2) is to find out all links having the maximal link load, and then for each such link, try to reroute one tree occupying it. It has three layers of loops. The outside loop is on each link having the maximal load. The second layer loop tries to successfully reroute one tree occupying the concerned link. The inner loop reroutes the tree to the most lightly loaded links. In this loop, we set a load threshold (i.e., value of j) for links to be removed from G before rerouting the chosen tree, say r_i. The initial value of j is set to $l_{\max}-1$, so that links whose load is greater than or equal to $l_{\max}-1$ will be removed. This is to ensure that after rerouting r_i, the maximal link load in the network will still be less than $l_{\max}$ (thus the maximal load is reduced). Inside this loop, each time the load threshold (i.e., j's value) is decremented by 1. That is, r_i will be rerouted to more lightly loaded links upon each interaction of the loop. When algorithm A returns $t_A=\emptyset$, r_i cannot be rerouted at the level with this value of j. Now, there are two cases, $j=l_{\max}-1$ or $j<l_{\max}-1$.

1) $j=l_{\max}-1$. In this case, it is impossible to reroute r_i without using links with load $l_{\max}$ or $l_{\max}-1$. It exits the while-loop of j and tries to reroute another tree in R, because successfully rerouting any tree in R would reduce the load on $e_{\max}$ by 1. When R becomes empty, that means all trees in R have been tried and none of them can be rerouted. In this case, the load on $e_{\max}$ cannot be reduced. The algorithm stops.
2) $j<l_{\max}-1$. In this case, r_i is rerouted successfully on $G(V,E_{j+1},c,d)$ whose link loads are less than or equal to j. This means that the load on link $e_{\max}$ can be reduced by 1 (the one was occupied by r_i). The next link in $E_{\max}$ will be chosen for load reduction. The maximal link load in the system, $l_{\max}$, is reduced by 1 when $E_{\max}$ becomes empty.

At Step 1), it takes time $O(k|E|)$ to calculate the load on each link. At Step 2), the while-loop of $E_{\max}$ would run at most $(|V|-1)$ times, because each routing tree has at most $(|V|-1)$ edges. The while-loop of R would run at most $l_{\max}$ times (for each tree in R). Each while-loop of j would be executed at most $(l_{\max}-1)$ times (for $j=l_{\max}-1,\ldots,2,1$) and each time algorithm A (whose time complexity is $O(|D_i||V|^2)$) will be called. Note that once the maximal load is reduced by 1, the algorithm will go back to Step 1). This process can be repeated at most $(l_{\max}-1)$ times, because the maximal load can be decreased (at most) from $l_{\max}$ to 1. Thus, the algorithm can terminate in time $O((K|V|^3 l_{\max}^2+k|E|)l_{\max})$. □

E. Algorithm D: Optimization through Wavelength Reassignment

For a set of routing trees assigned with wavelengths, algorithm D reduces the number of wavelengths by reassigning some of the trees in such a way that some of the wavelengths they are currently using can be freed. It will, 1) for each wavelength, calculate the set of routing trees it is assigned to and 2) reroute the trees which are assigned with the least used wavelength, so that they can be assigned with other wavelengths. The rerouting operation is repeated until the number of wavelengths used cannot be reduced any further. Let W be the set of currently used wavelengths.

Theorem 5: Algorithm D can reassign wavelengths to a given set of routing trees in time $O(K|V|^2|W|^3/k)$, where $K=\max\{|D_i||i=1,2,\ldots,k\}$.

Proof: In Step 2), when exiting the inner while-loop-j, there are two possible cases, $j=1$ or $j>1$.

1) $j=1$. In this case, there exists a tree r_i which cannot be rerouted and assigned to any other wavelength. That is, the number of wavelengths cannot be reduced. The algorithm stops.
2) $j>1$. In this case, tree r_i can be rerouted and reassigned with another wavelength w_j. The next tree in $T(w_1)$ will be chosen for rerouting. When $T(w_1)$ becomes empty, this means that w_1 can be freed.

At Step 1), it takes time $O(k)$ to determine the distribution of currently used wavelengths. At Step 2), the inner while-loop runs at most $(|W|-1)$ times (for $j=|W|,\ldots,2$) and each run of algorithm A needs time $O(|D_i||V|^2)$. The outer while-loop would run at most $|T(w_1)|$ times (for each tree in $T(w_1)$), where $|T(w_1)|\le|W|/k$. When w_1 is saved and removed from W, the algorithm will go back to Step 1). This process is repeated at most $(|W|-1)$ times. Therefore, the algorithm can terminate in time $O(K|V|^2|W|^3/k)$. □

IV. Simulations

In the above section, we have proposed two algorithms, algorithms C and D, for optimizing wavelength assignment. In order to evaluate the performance of the two algorithms, we simulate four different combinations of algorithms A, B, C, and D: nonoptimization AB, load balancing optimization ACB, wavelength assignment optimization ABD, and combined optimization $ACBD$.

A. Description of Simulation Model

Network topologies $G(V,E,c,d)$ are generated by using the approach introduced in [20] to emulate wide-area sparse networks deliberately. One-hundred nodes are distributed randomly over a rectangular coordinate grid. Each node is placed at a location with integer coordinates. A link between two nodes u and v is added by using the probability function $P(u,v)=\lambda\exp(-p(u,v)/\gamma\delta)$, where $p(u,v)$ is the distance between u and v, δ is the maximum distance between any two nodes, and $0<\lambda,\gamma\le 1$. Larger values of λ produce graphs with higher link densities, while small values of γ increases the density of short links relative longer ones. In our simulations, γ and λ both are set to 0.9. As a result, the nodes in generated graphs have average degrees of five. Both cost and delay functions c and d on link (u,v) in the generated graphs are the distance between nodes u and v on the rectangular coordinated grid.

The QoS multicast requests are generated randomly. For request $r_i(s_i,D_i,\Delta_i)$, nodes s_i and D_i are randomly picked up from the nodes in generated network graph. The delay bound Δ_i is set as: $\Delta_i\equiv\alpha\cdot\max\{d(P_G(s_i,u))|u\in D_i\}$. It is common in real-time communications to set the delay bound to be a constant times the worst case delay.

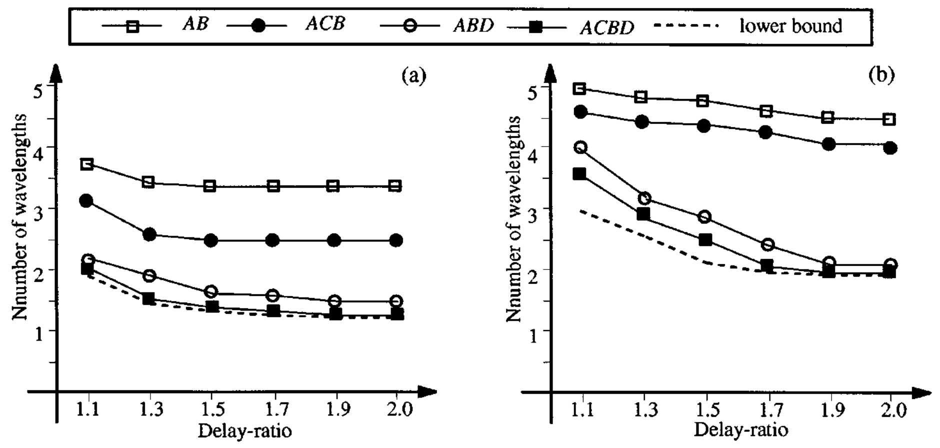

Fig. 5. Number of wavelengths versus delay-ratio with five channels: (a) five and (b) ten destinations.

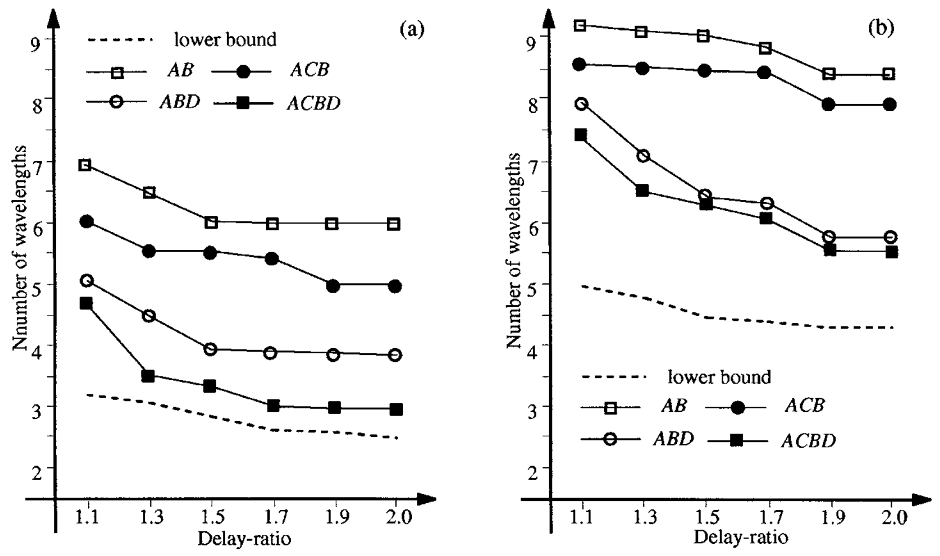

Fig. 6. Number of wavelengths versus delay-ratio with ten channels: (a) five and (b) ten destinations.

The objective of our simulation work is to find the effectiveness of our proposed optimization algorithms. That is, to determine the saving in wavelengths under various network environments. Throughout the simulations, we introduce a lower bound of number of wavelengths as a performance benchmark. The lower bound is defined as the maximal link load in the system, which is obtained by running algorithm AC(without considering wavelength assignment). At each simulation point, the simulation runs 50 times. Each time a different set of QoS multicast requests is generated, and algorithms AB, ACB, ABD, and $ACBD$ are applied, respectively. The number of wavelengths presented in the figures below are the mean values of 50 simulation runs.

B. Analysis of Simulation Results

In the simulations, we simulate the number of wavelengths against three parameters: delay ratio α, number of multicast destinations, and the number of multicast requests (i.e., the number of channels to be established).

Figs. 5–7 show the number of wavelengths versus delay ratio α. α varies from 1.1 to 2. The numbers of channels are set at 5, 10, and 20, respectively. Figs. 5–7(a) display the cases where the number of destinations is 5, and Figs. 5–7(b) are the cases where the number of destinations is 10. From Figs. 5–7 the following observations can be made.

1) Algorithm D reduces the number of wavelengths more effectively than algorithm C. Comparing the performance of AB with that of ACB, we can see that it only reduces the number of wavelengths slightly by using load balance rerouting algorithm C alone. Furthermore, by looking at the curves of ABD and $ACBD$, we can find that $ACBD$ does not improve the performance significantly over ABD. The reason is that algorithm D tries to free out the least used wavelengths during rerouting, which saves the number of wavelengths directly; while algorithm C aims at reducing the maximal link load during rerouting. Reducing the maximal link load in the system does not directly reduce the number of wavelengths (the wavelength assignment phase may negate some of the benefits gained at the load balancing phase).
2) The number of wavelengths monotonically decreases as the delay ratio increases. There are two reasons for this phenomenon: a) with a larger delay bound, algorithms C and D can have more chance to succeed in rerouting

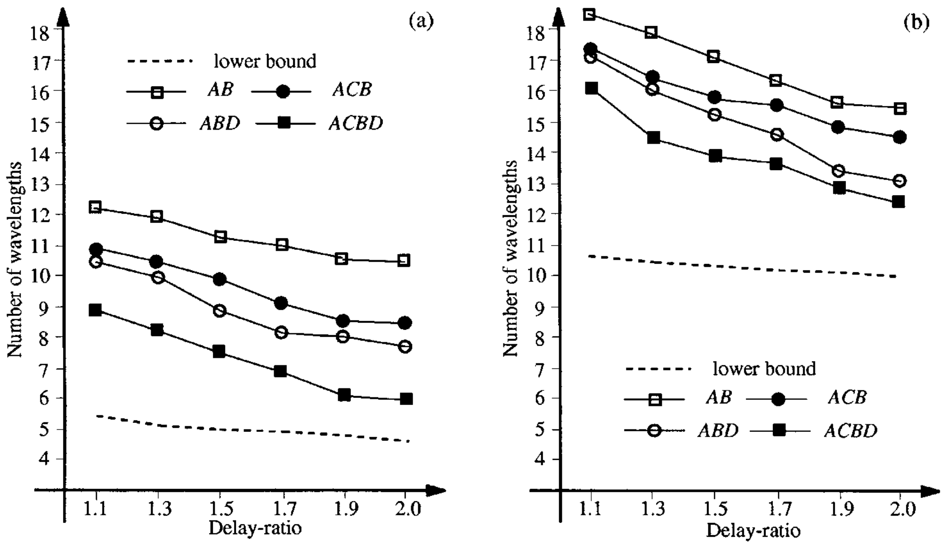

Fig. 7. Number of wavelengths versus delay-ratio with 20 channels: (a) five and (b) ten destinations.

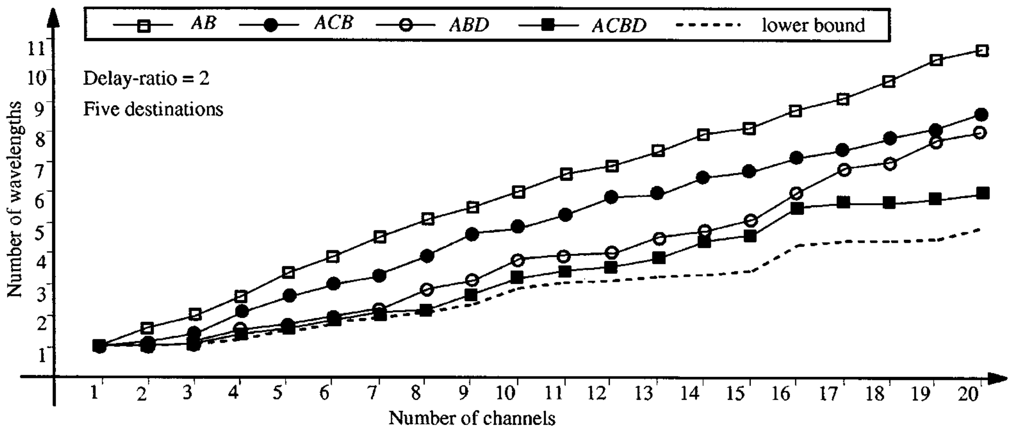

Fig. 8. Number of wavelengths versus number of channels with five destinations.

QoS routing trees, which results in more wavelength savings. b) according to Theorem 1, QoS routing trees of larger delay ratios have smaller cost. Usually, routing trees having smaller cost have less number of links. They, thus, have less chance to share links with each other. Therefore, algorithm D is able to assign the same wavelength to more trees without causing wavelength conflict when α is larger.

3) The optimization algorithms work more effectively in sparse networks. Comparing Figs. 5–7(a) with Figs. 5–7(b), we can see when the number of multicast destinations is smaller, the saving of wavelengths is more significant and the performance is closer to our defined low bound (the curves with dot line). A similar trend can be observed with the decrease of the number of channels in the system.

This is because when the system has multicasts with fewer destinations (or has fewer requests), the link loads are lighter (i.e., fewer trees share links with each other). The rerouting operations in our optimization algorithms have more chance to succeed. It can also be anticipated that our rerouting algorithms would work more effectively in sparse networks, where there is more room for rerouting.

The second group of simulations concerns the number of wavelengths versus the number of channels. In this group of simulations, the delay-ratio α is fixed at 2.0. The number of channels varies from 1 to 20. Fig. 8 shows the case where the number of destinations is set to 5, and in Fig. 9, it is set to 10. From Figs. 8 and 9, we see a continuous change of wavelengths as the increase of channels. The curves in Figs. 8 and 9 are consistent with the above observation 3). Our optimization algorithms work better in the cases where routing trees are smaller (i.e., fewer multicast destinations) or there are fewer routing trees (i.e., fewer channels).

V. Conclusion

In this paper, we have proposed two rerouting algorithms for optimizing wavelength assignment of QoS multicasts. From the simulation results, we can see that the optimization algorithms can significantly reduce the number of wavelengths over the

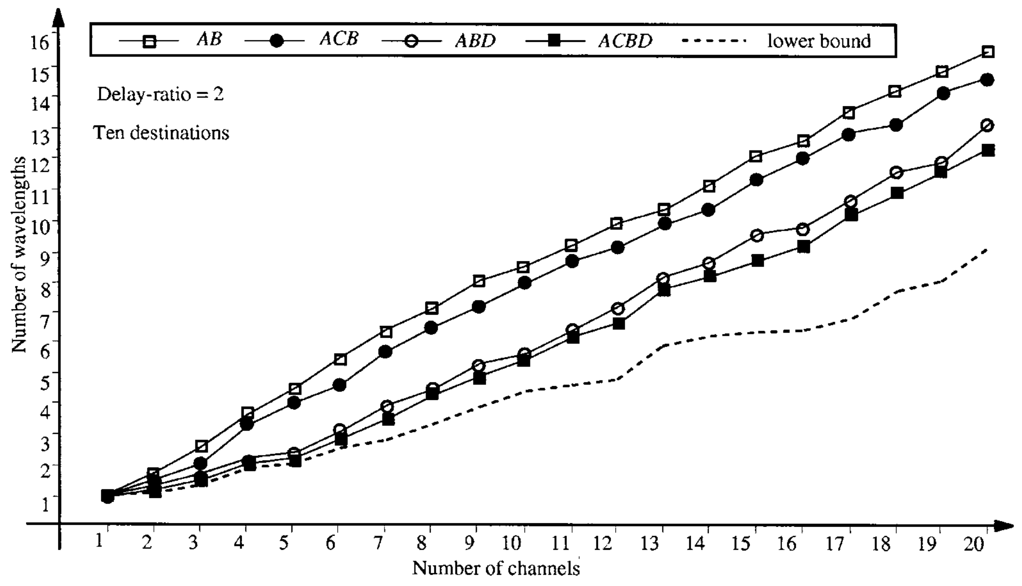

Fig. 9. Number of wavelengths versus number of channels with ten destinations.

cases where no optimization is done (i.e., algorithm AB). Moreover, we have found algorithm D (rerouting aiming at wavelength reassignment) demonstrates a considerably better performance than algorithm C (rerouting aiming at load balancing). This means that rerouting to free out the least used wavelengths is more effective than rerouting for load balancing.

We assume a bidirectional communication model in this paper. However, our optimization algorithms C and D are independent of this assumption. The proposed approach and the two optimization algorithms can be extended to the unidirectional model, where data (signal) is transmitted only in one direction from the source to the destination. The wavelength conflict rule in the unidirectional model becomes that two channels must use different wavelengths if their routes share a common link in the same direction.

Appendix

Proof of Theorem 1: Let $u_1, u_2, \ldots, u_k$ be the vertices that caused the shortest paths to be added during the depth first search in the order that they were encountered, and let $u_0 = s_i$. When the shortest path $P_G(s_i, u_j)$ from s_i to u_j ($j \geq 1$) was added into the current tree t_A, the total cost of the added edges is at most $c(P_G(s_i, u_j))$. In addition, the edges on the path to u_j consisting of the shortest path (in terms of function d) to u_{j-1} followed by the path in t_M from u_{j-1} to u_j are modified in order, so that

$$d(P_{t_A}(s_i, u_j)) \leq d(P_G(s_i, u_{j-1})) + d(P_{t_M}(u_{j-1}, u_j)).$$

The shortest path to u_j on G (in terms of function d) was added because

$$d(P_{t_A}(s_i, u_j)) > \Delta_i \geq \alpha \cdot d(P_G(s_i, u_j)).$$

Combining these two inequalities, we get

$$\alpha \cdot d(P_G(s_i, u_j)) < d(P_G(s_i, u_{j-1})) + d(P_{t_M}(u_{j-1}, u_j)).$$

To sum over j, we obtain

$$\alpha \sum_{j=1}^{k} d(P_G(s_i, u_j)) < \sum_{j=1}^{k} (d(P_G(s_i, u_{j-1})) + d(P_{t_M}(u_{j-1}, u_j)))$$

which implies

$$(\alpha - 1) \sum_{j=1}^{k} d(P_G(s_i, u_j)) < \sum_{j=1}^{k} d(P_{t_M}(u_{j-1}, u_j)).$$

Due to the assumption that the delay function d is proportional to the cost function c, we demonstrate that the total c-cost of the added paths is at most

$$(\alpha - 1) \sum_{j=1}^{k} c(P_G(s_i, u_j)) < \sum_{j=1}^{k} c(P_{t_M}(u_{j-1}, u_j)).$$

Notice that the depth first search traverses each edge exactly twice, and hence the sum on the right-hand side of the above inequality is at most twice the c-cost of t_M. This yields, the total cost of the added paths is less than $2 \cdot c(t_M)/(\alpha - 1)$. Therefore, the c-cost of the final routing tree t_A is less than

$$c(t_A) \leq \left(1 + \frac{2}{\alpha - 1}\right) c(t_M) \leq \beta \cdot c(t_M).$$

In addition, for $u \in D_i \backslash \{u_1, \ldots, u_k\}$, requirement (1) is met. For $u \in \{u_1, \ldots, u_k\}$, requirement (1) is met after the shortest path from s to u is added into t_A. Therefore, t_A satisfies requirement (1) and is a QoS routing tree. □

Acknowledgment

The authors would like to thank the anonymous referees for their valuable and constructive comments, which helped greatly in improving the quality and presentation of this paper.

References

[1] P. E. Green, *Fiber-Optic Networks*. Cambridge, MA: Prentice-Hall, 1992.

[2] D. Banerjee and B. Mukherjee, "A practical approach for routing and wavelength assignment in large wavelength-routed optical networks," *IEEE J. Select. Areas Commun.*, vol. 14, pp. 903–908, June 1996.

[3] L. H. Sahasrabuddhe and B. Mukherjee, "Light-trees: Optical multicasting for improved performance in wavelength-routed networks," *IEEE Commun. Mag.*, vol. 37, pp. 67–73, Feb. 1999.

[4] B. Mukherjee, *Optical Communication Networks*. New York: McGraw-Hill, 1997.

[5] F. K. Hwang and D. S. Richards, "Steiner tree problems," *Networks*, vol. 22, no. 1, pp. 55–89, 1992.

[6] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.

[7] K. Bharath-Kumar and J. M. Jaffe, "Routing to multiple destinations in computer networks," *IEEE Trans. Commun.*, vol. COM-31, pp. 343–351, Mar. 1983.

[8] X. Jia, "A distributed algorithm of delay-bounded multicast routing for multimedia applications in wide area networks," *IEEE/ACM Trans. Networking*, vol. 6, pp. 828–837, Dec. 1998.

[9] S. Khuller, B. Raghavachari, and N. Young, "Balancing minimum spanning trees and shortest-path trees," *Algorithmica*, vol. 14, no. 4, pp. 305–321, 1995.

[10] V. P. Kompella, J. C. Pasquale, and G. C. Polyaos, "Optimal multicast routing with quality of service constraints," *J. Network and Systems Management*, vol. 4, no. 2, pp. 107–131, 1996.

[11] R. K. Pankaj, "Wavelength requirements for multicasting in all-optical networks," *IEEE/ACM Trans. Networking*, vol. 7, pp. 414–424, June 1999.

[12] T. Erlebach and K. Jansen, "Scheduling of virtual connectins in fast networks," in *Proc. 4th Workshop on Parallel Systems and Algorithms (PASA)*, Germany, Apr. 10–12, 1996, pp. 13–32.

[13] P. Raghavan and E. Upfal, "Efficient routing in all-optical networks," in *Proc. 26th Annual ACM Symp. Theory of Computing (STOC)*, Montreal, PQ, Canada, May 23–25, 1994, pp. 134–143.

[14] G. Wilfong and P. Winkler, "Ring routing and wavelength translation," in *Proc. 9th Annual ACM-SIAM Symp. Discrete Algorithms (SODA)*, San Francisco, CA, 1998, pp. 333–341.

[15] P. Klein, S. Plotkin, C. Stein, and É. Tardos, "Faster approximation algorithms for the unit capacity concurrent flow problem with applications to routing and finding sparse cuts," *SIAM J. Computing*, vol. 23, no. 3, pp. 466–487, 1994.

[16] A. Srinivasan, "A survey of the role of multicommodity flow and randomization in network design and routing," *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, vol. 43, pp. 271–302, 1999.

[17] P. Raghavan and C. D. Thompson, "Randomized rouding: A technique for probably good algorithms and algorithmic proofs," *Combinatorica*, vol. 7, no. 4, pp. 365–374, 1987.

[18] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introuction to Algorithms*. Cambridge, MA: MIT Press, 1992.

[19] D. W. Matula, G. Marble, and J. D. Isaacson, "Graph coloring algorithms," in *Graph Theory and Computing*, R. C. Read, Ed. New York: Academic, vol. 1972, pp. 109–122.

[20] B. M. Waxman, "Routing of multipoint connections," *IEEE Select. Areas Commun.*, vol. 6, pp. 1617–1622, Dec. 1988.

Xiao-Hua Jia received the Ph.D. degree in information science from the University of Tokyo, Tokyo, Japan, in 1991.

He is currently an Associate Professor in the Department of Computer Science, City University of Hong Kong. His research interests include operating systems, distributed systems, network systems, computer communications, and real-time communications.

Ding-Zhu Du received the M.S. degree from Institute of Applied Mathematics, Chinese Academy of Sciences, in 1982, and the Ph.D. degree from the University of California at Santa Barbara in 1985.

He visited MSRI at Berkeley in 1985–1986, MIT in 1986–1987, and Princeton University in 1990-1991. Currently, he is a Professor with the Department of Computer Science, University of Minnesota, and also a Research Professor at the Institute of Applied Mathematics, Chinese Academy of Sciences. He has published more than 120 papers and 28 books. His research interests include combinatorial optimization, communication networks, and theory of computation.

Dr. Du is currently the Editor-in-Chief of the *Journal of Combinatorial Optimization* and is a member of editorial boards for eight journals.

Xiao-Dong Hu received the B.S. degree in applied mathematics from Qinghua University, Beijing, China, in 1985, and the Ph.D. degree in operations research and cybernetics from Institute of Applied Mathematics, Chinese Academy of Sciences, in 1989.

He was a Postdoctoral Fellow at Rutgers Center for Operations Research, Rutgers University, in 1990–1991, a Visiting Associate Professor at the Graduate School of Information Science, Japan Advanced Institute of Science and Technology, in 1993–1994, and a Research Fellow with the Computer Science Department, City University of Hong Kong, in 1998–2000. Currently, he is a Research Professor at the Institute of Applied Mathematics, Chinese Academy of Sciences. His research interests include combinatorial optimization and computer and communication networks.

Man-Kei Lee received the B.S. degree in computer science from the City University of Hong Kong in 1998. Currently, he is working toward the M.Phil. degree at the Computer Science Department, City University of Hong Kong.

His research interests include mobile computing, client server model, and reliable communications.

Jun Gu (S'86–M'90–SM'90) received the B.S. degree in electrical engineering (with honors) from the University of Science and Technology of China in 1982 and the Ph.D. degree in computer science (with honors) from the University of Utah, Salt Lake City, in 1989.

He was a Professor of Electrical and Computer Engineering at the University of Calgary, Calgary, AB, Canada, from 1994 to 1997. Currently, he is a Professor of Computer Science at Hong Kong University of Science and Technology. His research interests include VLSI circuit and system design, high-performance software systems, parallel and distributed processing, and combinatorial optimization.

Dr. Gu is on the editorial boards of five journals including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING and IEEE TRANSACTIONS ON VLSI SYSTEMS.