A Novel Generic Graph Model for Traffic Grooming in Heterogeneous WDM Mesh Networks

Hongyue Zhu, *Student Member, IEEE*, Hui Zang, *Member, IEEE*, Keyao Zhu, *Student Member, IEEE*, and Biswanath Mukherjee, *Member, IEEE*

***Abstract*—As the operation of our fiber-optic backbone networks migrates from interconnected SONET rings to arbitrary mesh topology, traffic grooming on wavelength-division multiplexing (WDM) mesh networks becomes an extremely important research problem. To address this problem, we propose a new generic graph model for traffic grooming in heterogeneous WDM mesh networks. The novelty of our model is that, by only manipulating the edges of the auxiliary graph created by our model and the weights of these edges, our model can achieve various objectives using different grooming policies, while taking into account various constraints such as transceivers, wavelengths, wavelength-conversion capabilities, and grooming capabilities. Based on the auxiliary graph, we develop an integrated traffic-grooming algorithm (IGABAG) and an integrated grooming procedure (INGPROC) which jointly solve several traffic-grooming subproblems by simply applying the shortest-path computation method. Different grooming policies can be represented by different weight-assignment functions, and the performance of these grooming policies are compared under both nonblocking scenario and blocking scenario. The IGABAG can be applied to both static and dynamic traffic grooming. In static grooming, the traffic-selection scheme is key to achieving good network performance. We propose several traffic-selection schemes based on this model and we evaluate their performance for different network topologies.**

***Index Terms*—Graph model, mesh network, optical network, traffic grooming, wavelength-division multiplexing (WDM).**

I. Introduction

WAVELENGTH-DIVISION multiplexing (WDM) is a key approach to increase the bandwidth of an optical network [1]. As WDM technology continues to mature, there exists a large gap between the capacity of a WDM channel (e.g., OC-48, or OC-192, or OC-768) and the bandwidth requirement of a typical connection request (e.g., STS-1, OC-3, OC-12, etc.). If the entire bandwidth of a wavelength channel is allocated to a low-speed connection, a large portion of the transmission capacity could be wasted. In order to use the network resources efficiently, low-speed traffic streams need to be efficiently multiplexed, or "groomed," onto high-speed lightpaths [2].

The traffic-grooming problem can be formulated as follows [3]. Given a network configuration (including physical topology, where each edge is a physical link, number of transceivers at each node, number of wavelengths on each fiber, and the capacity of each wavelength) and a set of connection requests with different bandwidth granularities, such as OC-12, OC-48, etc., we need to determine how to set up lightpaths to satisfy the connection requests. Because of the subwavelength granularity of the connection requests, one or more connections can be multiplexed on the same lightpath.

The set of connection requests can all be given in advance (static traffic), or given one at a time (dynamic traffic). Traffic grooming with static traffic is a dual optimization problem. In a nonblocking scenario, where the network has enough resources to carry all of the connection requests, the objective is to minimize the network cost, e.g., total number of wavelength-links used in a WDM mesh network, while satisfying all the requests, where a wavelength-link is defined as a wavelength in a fiber-link. In a blocking scenario, where not all connections can be set up due to resource limitations, the objective is to maximize the network throughput. With dynamic traffic, where connections arrive one at a time, the objective is to minimize the network resources used for each request, which implicitly attempts to minimize the blocking probability for future requests.

Traffic grooming is usually divided into four subproblems [3], which are not necessarily independent:

1) determining the virtual topology that consists of lightpaths;
2) routing the lightpaths over the physical topology;
3) performing wavelength assignment to the lightpaths;
4) routing the traffic on the virtual topology.

The virtual-topology design problem [4]–[9] is conjectured to be NP-hard [1]. In addition, routing and wavelength assignment (RWA) [10] is also NP-hard [11]. Therefore, traffic grooming in a mesh network is also a NP-hard problem [3].

To solve the traffic-grooming problem, one approach is to deal with the four subproblems separately. It first determines the virtual topology, then performs routing and wavelength assignment, and finally routes the traffic requests. There are considerable research results on each subproblem already and they can be utilized to solve the traffic-grooming problem. Although this divide-and-conquer method makes traffic grooming easier to handle, it cannot achieve the optimal solution even if we can get the optimal solution for each subproblem, since these four subproblems are not necessarily independent and the solution to one subproblem might affect how optimally another

Manuscript received February 12, 2002; revised June 20, 2002, approved by IEEE TRANSACTIONS ON NETWORKING Editor R. Srikant. This work was supported by Sprint Advanced Technology Laboratories and by the National Science Foundation under Grant ANI-9805285.

H. Zhu was with the Transmission Research Group, Sprint Advanced Technology Laboratories, Burlingame, CA 94010 USA. He is now with the Department of Computer Science, University of California, Davis, CA 95616 USA (e-mail: zhuh@cs.ucdavis.edu).

K. Zhu and B. Mukherjee are with the Department of Computer Science, University of California, Davis, CA 95616 USA (e-mail: zhuk@cs.ucdavis.edu; mukherje@cs.ucdavis.edu).

H. Zang is with Sprint Advanced Technology Laboratories, Burlingame, CA 94010 USA (e-mail: hzang@sprintlabs.com).

Digital Object Identifier 10.1109/TNET.2003.810310

subproblem can be solved. Sometimes, using the optimal solution for one subproblem might not lead to the optimal solution to the whole problem. Moreover, this approach requires all the traffic requests to be known in advance, which cannot be satisfied in dynamic grooming.

Another approach is to solve the four subproblems as a whole. Since it can take into account all the constraints regarding the four subproblems simultaneously, this approach has a potential to achieve better performance. With static traffic, the traffic-grooming problem can be formulated as an integer linear program (ILP) [3], and an optimal solution can be obtained for some relatively small networks. However, an ILP is not scalable and cannot be directly applied to large networks. One way to make the problem tractable is to develop heuristic algorithms and jointly solve the grooming problem for one connection request at a time. To the best of our knowledge, no integrated heuristic algorithm for solving the traffic-grooming problem has been developed for wavelength-routed networks in previous work.

A. Previous Work

Traffic grooming is an important and practical problem for designing WDM networks and it is receiving increasing research attention both in academia and in industry. The work in [12] reviews most of the recent research work on traffic grooming in WDM ring and mesh networks.

Past research efforts on traffic grooming have mainly focused on SONET/WDM ring networks. The major cost of such a network is considered to be dominated by SONET add-drop multiplexers (ADMs). Therefore, minimizing the number of SONET ADMs has been the objective of static traffic grooming in recent research. The general traffic-grooming problem in a SONET/WDM ring network is proven to be NP-complete [13], [14]. An optimal algorithm for a single-hub ring is proposed in [13] and several optimal or near-optimal algorithms for traffic grooming and wavelength assignment to reduce the number of wavelengths and SONET ADMs are proposed in [15]. As a network design problem, the authors in [16] attempt to minimize the network cost, which is dominated by SONET ADMs, in an optical add-drop wavelength-division-multiplexed (OADM) ring network. Six optical WDM ring architectures are provided in [16] and the cost of different architectures, as well as the switching capabilities of different architectures under various traffic assumptions are compared. The maximum terminal-equipment savings using wavelength ADMs are quantified in [17] for WDM rings carrying uniform and distance-dependent traffic. Grooming with arbitrary traffic in bidirectional-line-switched rings (BLSRs) is addressed in [14]. In [18], based on a general formulation of the virtual-topology problem, a framework used to evaluate the performance of heuristics and requiring less computation than evaluating the optimal solution is presented. The authors in [19] formulate the grooming optimization problem as an ILP and compare single-hop grooming and multihop grooming. Instead of single-ring architectures, interconnected WDM rings are studied in [20] and several strategies for traffic grooming in such networks are compared. All the above references except [16] focus on static traffic only. The authors of [21] study the dynamic traffic-grooming problem in SONET/WDM rings and formulate it as a bipartite graph-matching problem.

As our fiber-optic backbone networks migrate from rings to mesh, traffic grooming on WDM mesh networks becomes an extremely important area of research. The work in [22] formulates the static traffic-grooming problem as an ILP and proposes a heuristic to minimize the number of transceivers. In [23], several lower bounds for regular topologies are presented and greedy and iterative greedy schemes are developed. However, in both [22] and [23], the authors relax the physical-topology constraints, assuming all the virtual topologies are implementable on the given physical topology, i.e., they do not consider lightpath routing and wavelength assignment. The authors in [3] propose several node architectures for supporting traffic grooming in WDM mesh networks and formulate the static traffic-grooming problem as an ILP. They present two heuristics and compare the performance with that of the ILP. The works in [24]–[29] consider a dynamic traffic pattern in WDM mesh networks. In [24], the authors propose a connection admission control scheme to ensure fairness in terms of connection blocking. A theoretical capacity correlation model is presented in [25] to compute the blocking probability for WDM networks with constrained grooming capability. In [26], two route-computation algorithms are proposed and compared, and the results indicate that, in order to achieve good performance in a dynamic environment, different grooming policies and route-computation algorithms need to be used under different network states. The work in [27] compares two schemes to dynamically establish reliable low-speed traffic in WDM mesh networks with traffic-grooming capability. In [28], the problem of planning and designing a WDM mesh network with certain forecast traffic demands, to satisfy all the connections as well as minimize the network cost, is studied. In [29], the authors investigate the design of multilayer mesh networks to satisfy each connection's bandwidth and protection requirements while minimizing the overall network cost.

B. Challenges of Traffic Grooming in a Heterogeneous WDM Mesh Network

The WDM backbone network is expected to emerge as a multivendor heterogeneous network. As WDM networks migrate from ring topologies to mesh topologies, it is very important to solve the traffic-grooming problem in a heterogeneous mesh network environment.

In terms of wavelength-conversion capability, heterogeneity means that some of the nodes in a network may have full wavelength-conversion capability (any incoming wavelength can be converted into any outgoing wavelength), some may have no wavelength-conversion capability (traffic must stay on the same wavelength when bypassing these nodes) [30], [31], and some may have partial wavelength-conversion capability (some wavelengths can be converted into some other wavelengths) [32]–[35]. In previous work, however, it was assumed that all the nodes in a network either have wavelength-conversion capability or none has wavelength-conversion capability. In addition, if a node has this capability, it always has full wavelength-conversion capability. This all-or-nothing assumption may not be practical or valid in the future WDM network. It is necessary to address the partial and sparse wavelength-conversion scenarios.

In a WDM mesh network, each node must support two functionalities: wavelength routing, which can be accomplished by an optical crossconnect (OXC), and optical multiplexing/demultiplexing, by which several wavelengths can be multiplexed to or demultiplexed from the same fiber-link. Besides, in order to groom low-speed connections onto a high-speed wavelength channel, a node will need to employ access stations, which can multiplex/demultiplex and switch low-speed connections using various multiplexing techniques, e.g., time-division multiplexing (TDM). A WDM mesh network may consist of systems from multiple vendors, and different vendors may employ different node architectures, which may have different grooming capabilities. Some architectures may have full grooming capabilities, while others may impose some constraints on the grooming capability, such as the number of transceivers used for originating and terminating groomable wavelength channels (also known as grooming ports). In addition, some nodes may have no grooming capability. These partial and sparse grooming-capability scenarios are very practical and should also be considered when solving the traffic-grooming problem.

To solve the traffic-grooming problem, the integrated approach is desirable not only because it has the potential to achieve better performance than the separated approach, but also because it can be used directly for dynamic traffic grooming, where the separated approach cannot be used. For a given connection request, the integrated approach should address the following issues.

1) Should this connection be routed on the current virtual topology, if it is possible to do so? Sometimes, it may be better to set up a new lightpath even though the connection can be carried on the current virtual topology.
2) How to change the virtual topology to accommodate the connection? i.e., between which two nodes should we set up a new lightpath, if any? In some cases, we can set up a lightpath directly from the source of the traffic to the destination. In other cases, it is not necessary or possible to set up this lightpath and we may need to set up one or more lightpaths and route the connection onto these lightpaths and/or some existing lightpaths.

Different decisions on these questions can result in different network performance. These decisions reflect the intentions of the network operator, and they are referred to as *grooming policies* [26]. By using different grooming policies, a network operator can achieve various objectives, such as minimizing the number of wavelength-links, minimizing the number of lightpaths, minimizing the traffic hops on the virtual topology, etc. As the network state changes, the optimization objective may also need to change. Dynamically evolving the grooming policy according to the network state is also a challenge for traffic grooming. Dynamic traffic grooming in a WDM mesh network is addressed in our extension work [36].

C. Our Contribution

In this paper, we propose a novel generic graph model for traffic grooming in a heterogeneous WDM mesh network. In this model, various factors of heterogeneity of the network, such as the number of transceivers at each node, the number of wavelengths on each fiber-link, as well as wavelength-conversion capabilities and grooming capabilities of each node, are represented by different edges in an auxiliary graph constructed by our model. This model can also achieve various objectives using different grooming policies. Moreover, instead of designing a route-computation algorithm for each grooming policy, simple shortest-path route-computation algorithms can be used in this model to achieve various grooming policies by carefully choosing the weight functions for the edges in the auxiliary graph. Three different grooming policies are proposed and their performance is compared under blocking and nonblocking scenarios. Based on the auxiliary graph, we develop an integrated traffic-grooming algorithm which jointly solves the four traffic-grooming subproblems. The integrated traffic-grooming algorithm can be applied to both static and dynamic traffic grooming. In static grooming, proper selection of the traffic requests is key to achieving a good network performance. We present several traffic-selection schemes based on this model and evaluate their performance for different network topologies.

D. Organization of the Paper

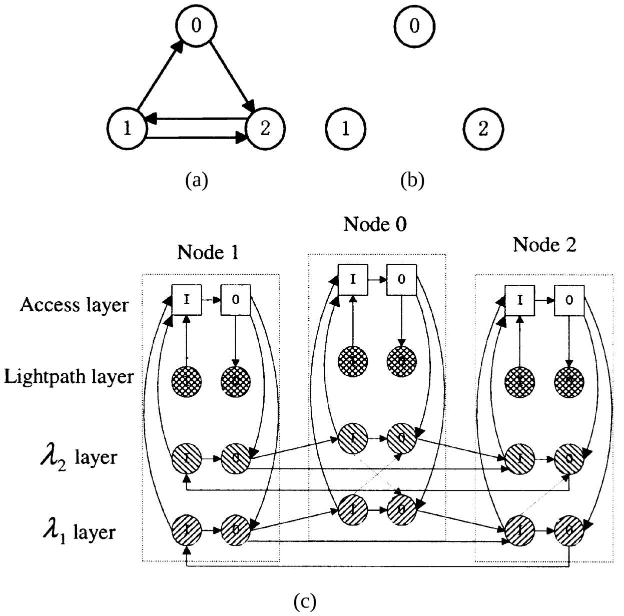
In Section II, we demonstrate how to construct, according to the network state, an auxiliary graph, which is the basis of our graph model. Based on this, an integrated traffic-grooming algorithm and three selection schemes used by the algorithm for static traffic are proposed and an illustrative example is given in Section III. In Section IV, the grooming policy is analyzed and three different grooming policies are proposed. Methods to choose the weight-assignment functions for the auxiliary graph to achieve these policies are also discussed. In Section V, the performance of different grooming policies is shown for blocking and nonblocking scenarios. The performance of the three selection schemes used by the integrated traffic-grooming algorithm for static traffic is also compared under different network scenarios. Section VI concludes the paper.

II. Construction of an Auxiliary Graph

In order to solve the traffic-grooming problem, we first construct an auxiliary graph according to the given network configuration.

An illustrative example is shown in Fig. 1. In order to make the constructed auxiliary graph clear to see, we choose a very simple network topology. Network 1 [Fig. 1(a)] is a three-node network with four unidirectional fiber-links, each of which has two wavelengths. Node 0 has wavelength converters with full wavelength-conversion capability, node 1 has no wavelength converter, and node 2 has wavelength converters with limited wavelength-conversion capability in the sense that only wavelength λ_1 can be converted to λ_2. In the beginning, there is no lightpath in the network, so there is no edge in the virtual topology of Network 1, as shown in Fig. 1(b). An auxiliary graph is constructed as in Fig. 1(c).

In general, a network can be represented by a graph $G_0(V_0, E_0)$, where V_0 and E_0 are its node set and link set, respectively. Assuming that each link has W wavelengths, λ_1 through λ_W, we construct the corresponding auxiliary graph as follows.



(a) (b) (c)

Fig. 1. (a) Physical topology of network 1. (b) Virtual topology of network 1. (c) Auxiliary graph of network 1.

For clarity, we will use the terms *node* and *link* to represent a vertex and an edge, respectively, in the original network $G_0(V_0, E_0)$, and we will use the terms *vertex* and *edge* to represent a vertex and an edge in the auxiliary graph $G(V, E)$, respectively.

Auxiliary graph G is a layered graph with $W + 2$ layers. Layers 1 through W denote the W *wavelength layers*, layer $W+1$ is called the *lightpath layer*, and layer $W + 2$ is called the *access layer*, where a traffic flow starts and terminates. Each node has two ports on each layer, an input port and an output port. Let $N_i^{l,p}$ denote port p on layer l at node i; then $V = \{N_i^{l,p} | p \in \{0,1\},\ 1 \le l \le W+2,\ \forall i \in V_0\}$, where $N_i^{l,0}$ and $N_i^{l,1}$ denote the input port and the output port on layer l at network node i, respectively. Each edge in the auxiliary graph G has a property tuple $PT(c, w)$ associated with it, where c denotes the capacity of this edge and w denotes its weight. The edges are inserted in auxiliary graph G as follows.

- **Wavelength Bypass Edges (WBE).**

  There is an edge from the input port to the output port on each wavelength layer at node i

  $$\langle N_i^{l,0}, N_i^{l,1} \rangle \in E, \qquad \forall i \in V_0,\ 1 \le l \le W. \tag{1}$$

  We call the edge $\langle N_i^{l,0}, N_i^{l,1} \rangle$ *wavelength bypass edge* on layer l at node i and it is denoted as $WBE(i,l)$. The capacity of the edge is ∞.
- **Grooming Edges (GrmE).**

  There is an edge from the input port to the output port on access layer at node i if node i has grooming capability

  $$\left\langle N_i^{W+2,0}, N_i^{W+2,1} \right\rangle \in E, \qquad \forall i \in V_0. \tag{2}$$

  We call the edge $\langle N_i^{W+2,0}, N_i^{W+2,1} \rangle$ *grooming edge* at node i and it is denoted as $GrmE(i)$. The capacity of the edge is ∞.
- **Mux Edges (MuxE).**

  There is an edge from the output port on the access layer to the output port on the lightpath layer at each node

  $$\left\langle N_i^{W+2,1}, N_i^{W+1,1} \right\rangle \in E, \qquad \forall i \in V_0. \tag{3}$$

  We call the edge $\langle N_i^{W+2,1}, N_i^{W+1,1} \rangle$ *mux edge* at node i and it is denoted as MuxE(i). The capacity of the edge is ∞.
- **Demux Edges (DmxE).**

  There is an edge from the input port on the lightpath layer to the input port on the access layer at each node

  $$\left\langle N_i^{W+1,0}, N_i^{W+2,0} \right\rangle \in E, \qquad \forall i \in V_0. \tag{4}$$

  We call the edge $\langle N_i^{W+1,0}, N_i^{W+2,0} \rangle$ *demux edge* at node i and it is denoted as DmxE(i). The capacity of the edge is ∞.
- **Transmitter Edges (TxE).**

  There is an edge from the output port on the access layer to the output port on wavelength layer l if there are transmitters available on wavelength λ_l at node i

  $$\left\langle N_i^{W+2,1},\ N_i^{l,1} \right\rangle \in E, \qquad \forall i \in V_0;\ 1 \le l \le W;\ TX_i^l > 0 \tag{5}$$

  where $TX_i^l (1 \le l \le W)$ denotes the number of transmitters that can operate at wavelength λ_l at node i.

  We call the edge $\langle N_i^{W+2,1}, N_i^{l,1} \rangle$ *transmitter edge* on layer l at node i and it is denoted as $TxE(i,l)$. The capacity of the edge is ∞.
- **Receiver Edges (RxE).**

  There is an edge from the input port on wavelength layer l to the input port on the access layer if there are receivers available on wavelength λ_l at node i

  $$\left\langle N_i^{l,0},\ N_i^{W+2,0} \right\rangle \in E, \qquad \forall i \in V_0;\ 1 \le l \le W;\ RX_i^l > 0 \tag{6}$$

  where $RX_i^l (1 \le l \le W)$ denotes the number of receivers that can operate at wavelength λ_l at node i.

  We call the edge $\langle N_i^{l,0}, N_i^{W+2,0} \rangle$ *receiver edge* on layer l at node i and it is denoted as $RxE(i,l)$. The capacity of the edge is ∞.
- **Converter Edges (CvtE).**

  There is an edge from the input port on wavelength layer l_1 to the output port on wavelength layer l_2 at node i if wavelength λ_{l_1} can be converted to wavelength λ_{l_2} without using an access station at node i

  $$\left\langle N_i^{l_1,0},\ N_i^{l_2,1} \right\rangle \in E, \qquad \forall\, i \in V_0; \quad \text{wavelength } l_1 \text{ is convertible to } l_2 \text{ at node } i. \tag{7}$$

  We call the edge $\langle N_i^{l_1,0}, N_i^{l_2,1} \rangle$ *converter edge* from layer l_1 to layer l_2 at node i and it is denoted as $CvtE(i, l_1, l_2)$. The capacity of the edge is ∞.

- **Wavelength-Link Edges (WLE).**
  There is an edge from the output port on wavelength layer l at node i to the input port on wavelength layer l at node j if there is a physical link from node i to node j and wavelength λ_l on this link is not used.

$$\left\langle N_i^{l,1}, N_j^{l,0} \right\rangle \in E, \qquad (i,j) \in E_0; \quad \text{wavelength } \lambda_l \text{ on link } (i,j) \text{ is not used.} \tag{8}$$

  We call the edge $\langle N_i^{l,1}, N_j^{l,0} \rangle$ *wavelength-link edge* on layer l from node i to node j and it is denoted as WLE(i,j,l). The capacity of this edge is the capacity of the corresponding wavelength on the link from node i to node j.
- **Lightpath Edges (LPE).**
  There is an edge from the output port on the lightpath layer at node i to the input port on the lightpath layer at node j if there is a lightpath from node i to node j

$$\left\langle N_i^{W+1,1},\ N_j^{W+1,0} \right\rangle \in E, \quad \exists \text{ a lightpath from node } i \text{ to node } j. \tag{9}$$

  We call the edge $\langle N_i^{W+1,1},\ N_j^{W+1,0} \rangle$ the *lightpath edge* from node i to node j and it is denoted as LPE (i,j). The capacity of this edge is the residual capacity of the corresponding lightpath from node i to node j.

As a final step in constructing the auxiliary graph, we need to assign weights to each edge, i.e., determine w in the property tuple $PT(c,w)$ of each edge. The weights can reflect the cost of each network element (transceiver, wavelength-link, wavelength converter, etc.), and/or a certain grooming policy. By applying different weight settings, this graph model can be used to achieve different objectives. These weights can either be fixed, or they can be adjusted according to the current network state. We will discuss the weight function in Section IV-B.

Note that, for each edge, we can keep some other useful edge-specific information in the property tuple also. For instance, for each lightpath edge, the routing and wavelength assignment information can be saved in the property tuple.

From the above procedure, it should be clear that the auxiliary graph reflects the current state of the network, which can be heterogeneous, with different nodes having different resources and capabilities.

III. Solving the Traffic-Grooming Problem Based on the Auxiliary Graph

Traffic grooming is usually divided into four subproblems [3]:

- determine the virtual topology of the network, i.e., which nodal transmitter should be directly connected to which nodal receivers;
- route the lightpaths over the physical topology;
- assign wavelengths to the lightpaths (this problem has been shown to be NP-hard in [11] and there are various heuristics to solve it [10]);
- route the traffic on the virtual topology.

In previous work, these four subproblems were usually solved separately [3]. For instance, routing the traffic can only be done after the virtual topology has been selected. This approach has a drawback since these four subproblems are not necessarily independent. It does not take into account the impact of routed traffic on the network state when determining the virtual topology. In addition, it cannot combine the knowledge about the physical-topology layer and virtual-topology layer to determine the route of the traffic. In our study, based on the auxiliary graph, we propose an integrated algorithm which jointly solves the four subproblems. Since it can take advantage of all the updated information about the subproblems, this approach has a potential to achieve very significant improvement in performance.

A. Integrated Grooming Algorithm Based on the Auxiliary Graph (IGABAG)

We first introduce the IGABAG algorithm, which solves the four subproblems for one traffic demand, and we then provide its complexity analysis.

1) Algorithm: The IGABAG algorithm needs initialization before being used. The initialization takes as a parameter the network configuration, which includes network topology, as well as node and link configuration, and according to the network configuration, it constructs the corresponding auxiliary graph G using the method discussed in Section II.

The input of the IGABAG algorithm is a traffic demand, which is represented by $T(s,d,g,m)$, where s and d are the source and destination nodes, respectively, g is the granularity of the traffic demand, for instance, OC-48, and m is the amount of the traffic in units of g. The algorithm works as follows.

```
**Algorithm IGABAG**
Input: a traffic demand $T(s,d,g,m)$.
Step 1: Delete the edges whose capacity
 is less than the bandwidth granularity
 of $T$, since they cannot accommodate $T$.
Step 2: Find the shortest path $p$ from the
 output port on the access layer of the
 source to the input port on the access
 layer of the destination of $T$ on graph
 $G$. If not successful, restore the edges
 previously deleted in Step 1 and return
 −1.
Step 3: If $p$ contains wavelength-link
 edges, one or more lightpaths going
 through the corresponding wavelength-
 links needs to be set up. A lightpath
 starts whenever $p$ travels through a
 transmitter edge, follows the subsequent
 wavelength-link edges, and terminates at
 the first receiver edge.
Step 4: Route $T$ along the pre-existing
 lightpaths in $p$ and/or lightpaths set up
 according to $p$. If the capacity of the
 path, which is defined as the minimum
 capacity of the lightpaths along the
 path, is less than the entire amount of
 $T$, route the maximum amount possible,
 say, $n$ units, of the traffic granularity
 $g$.
```

```
Step 5: Restore the edges previously
 deleted in Step 1.
Step 6: Update graph $G$ as follows:
  6.1: For each lightpath newly set up, a
   lightpath edge from the output port of
   starting node of the lightpath to the
   input port of ending node is added on
   the lightpath layer.
  6.2: The wavelength-link edges denoting
   the wavelength-links used by the light-
   path are removed from the corresponding
   wavelength layers. (Note that, if there
   are multiple fiber-links between the
   nodes, the wavelength-link edges are
   removed only when the corresponding
   wavelengths on all the fiber-links
   are used. So this algorithm can also
   be used in the case where there are
   multiple fiber-links between the same
   node pair.)
  6.3: If there is no transmitter (re-
   ceiver) available at node $i$ on wave-
   length $\lambda_l$, the transmitter edge $TxE(i,l)$
   (receiver edge $RxE(i,l)$) will be re-
   moved from $G$, i.e., this node cannot
   source/sink a lightpath on wavelength
   $\lambda_l$ anymore and can only be bypassed by
   a lightpath.
  6.4: If there is no wavelength converter
   which can convert wavelength $\lambda_{l_1}$ to
   wavelength $\lambda_{l_2}$ available at node $i$, the
   converter edge $CvtE(i,l_1,l_2)$ will be
   removed from $G$.
  6.5: Update the property tuple $PT(c,w)$ of
   the edges. For the lightpaths carrying
   the traffic $T$, the capacities of the
   lightpath edges denoting the lightpaths
   carrying the traffic $T$ are decreased by
   the amount of the traffic routed.
   Updating the weights of the edges in
   the graph will change the grooming
   policies. We will discuss the grooming
   policies in Section IV.
Step 7: If the entire traffic is accom-
   modated, return 0. Otherwise, return
   $m-n$, which is the amount of the
   uncarried traffic in units of $g$.
```

It can be observed that the IGABAG algorithm routes a given traffic request under the current network state and updates the network state after routing, making the auxiliary graph always reflect the current network state.

2) Complexity Analysis: Suppose there are N nodes in the network and each link has W wavelengths. In the corresponding auxiliary graph, there are $2 \times N \times (W+2)$ vertices. Since the running time of shortest-path computation using Dijkstra algorithm is $O(V^2)$, where V is the number of the vertices in the graph, the running time of IGABAG algorithm is $O(N^2W^2)$. If each node in the network has full wavelength-conversion capability, all the wavelength layers can be collapsed into one wavelength layer since all the wavelengths are equivalent. In this special case, the running time of IGABAG algorithm is $O(N^2)$.

B. Integrated Grooming Procedure (INGPROC) and Traffic-Selection Schemes

The IGABAG algorithm is used to accommodate one connection request, but in traffic-grooming problems, we need to route a set of requests. Based on the IGABAG algorithm, we propose the following INtegrated Grooming PROCedure (INGPROC) to solve the traffic-grooming problem. The input of INGPROC includes network configuration and a set of traffic requests.

```
Procedure INGPROC
Input: network configuration and a set of
   traffic requests.
Step 1: Initialize IGABAG with network
   configuration.
Step 2: Select a traffic demand $T(s,d,g,m)$
   from the traffic-request set.
Step 3: Apply IGABAG to $T$ and let $k$
   denote the return value.
Step 4: If $k > 0$, insert $T(s,d,g,k)$ into the
   request set.
Step 5: Go to Step 2 unless all the
   traffic has been routed, or no traffic
   can be routed with the remaining
   network resources.
```

Note that the INGPROC procedure can be applied to both static and dynamic grooming. For dynamic grooming, INGPROC just chooses the current traffic request in Step 2. For static grooming, where all the traffic demands are known in advance, the order in which the requests are routed plays an important role in achieving good performance. We propose the following traffic-request-selection schemes for static traffic grooming.

- *Least Cost First (LCF).* LCF chooses the most cost-effective traffic request under the current network state and routes it. The cost of a traffic request is the weight of the shortest path for routing the traffic on the corresponding auxiliary graph divided by the amount of the traffic, which is computed as the granularity multiplied by the units of the traffic. Note that, after routing a connection, LCF need to re-compute the cost of the unrouted connections under the updated network state. If there are N nodes in the network, W wavelengths on each link, and D traffic demands, the running time of INGPROC using LCF is $O(D^2N^2W^2)$, assuming no wavelength-conversion capability, or $O(D^2N^2)$, assuming full wavelength-conversion capability.
- *Maximum Utilization First (MUF).* MUF selects the connection with the highest utilization, which is defined as the total amount of the request divided by the number of hops from the source to the destination on the physical topology. The running time of INGPROC using MUF is

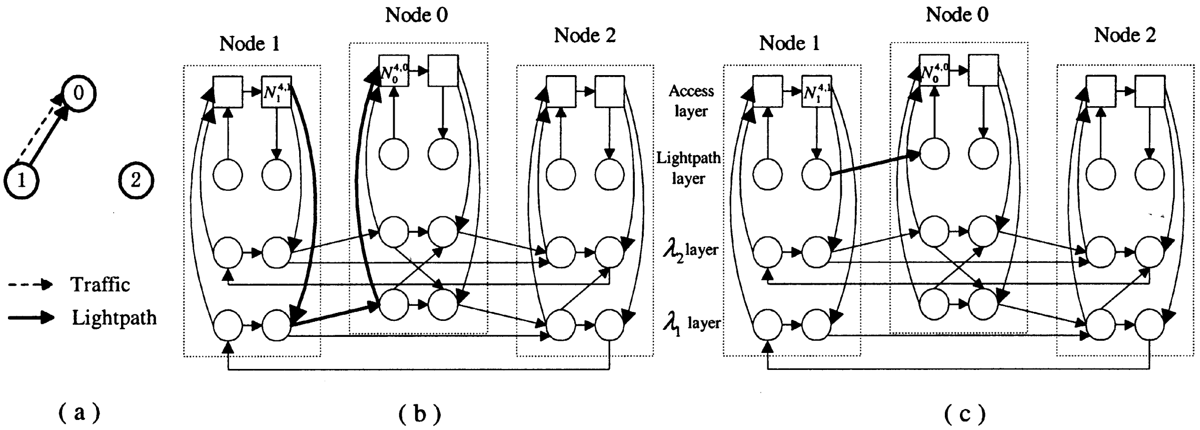


Fig. 2. (a) Virtual topology of network 1. (b) Corresponding auxiliary graph before routing the first traffic request T_1. (c) Corresponding auxiliary graph after routing the first traffic request T_1.

$O(D\log D + DN^2W^2)$, assuming no wavelength-conversion capability, or $O(D\log D + DN^2)$, assuming full wavelength-conversion capability.[1]

- *Maximum Amount First (MAF).* MAF selects the connection with the largest amount of demand and routes it. The running time of INGPROC using MAF is $O(D\log D + DN^2W^2)$, assuming no wavelength-conversion capability, or $O(D\log D + DN^2)$, assuming full wavelength-conversion capability.

We will compare the performance of these traffic-selection schemes in Section V.

In dynamic grooming, connections arrive one at a time, hold for a certain time period, and terminate. When a connection terminates, the resource used for this connection must be released. How to use our graph model to solve dynamic-grooming problem is shown in our extension work [36].

C. Illustrative Example

To illustrate how the graph model and the IGABAG algorithm work, we give an example based on the network in Fig. 1. Suppose the capacity of each wavelength is OC-48 and each node has grooming capability and two tunable transceivers.

The first connection request T_1 is *T*(1, 0, OC-12, 2). To satisfy this request, we need to find in the auxiliary graph a path from $N_1^{4,1}$ to $N_0^{4,0}$. It is easy to see that there exists a path along the edges TxE(1,1), WLE(1,0,1), and RxE(0,1), shown as bold lines in Fig. 2(b). Since this path contains a wavelength-link edge WLE(1,0,1), which denotes a wavelength-link, we need to set up a lightpath L_1 using λ_1 on the fiber-link from node 1 to node 0. After setting up L_1, we need to add a lightpath edge LPE(1,0) into the graph, which means that there is a lightpath from node 1 to node 0. Meanwhile, the wavelength-link edge WLE(1,0,1) must be removed from the graph since this wavelength-link cannot be used to set up another lightpath later on. This connection T_1 then can be routed onto lightpath L_1 and the residual capacity of L_1 is 2 × OC-12. So the capacity of edge LPE(1,0) is 24, which means that the capacity is equivalent to 24 OC-1's. The current virtual topology and the updated auxiliary graph are shown in Fig. 2(a) and (c), respectively.

Suppose the second connection request T_2 is *T*(2, 0, OC-12, 1). Following the same procedure as above, we need to determine a path from $N_2^{4,1}$ to $N_0^{4,0}$. There exist several paths in the auxiliary graph.

- **Case 1 (Single-hop grooming).** One path is along the edges TxE(2,2), WLE(2,1,2), WBE(1,2), WLE(1,0,2), and RxE(0,2), shown as bold lines in Fig. 3(b). This path contains edges WLE(2,1,2) and WLE(1,0,2), which denote wavelength λ_2 on the fiber-links from node 2 to node 1 and from node 1 to node 0, respectively. If this path is chosen, a lightpath L_2 consisting of these two wavelength-links needs to be set up. As a result, a lightpath edge LPE(2,0) is added into the graph and the two wavelength-link edges WLE(2,1,2) and WLE(1,0,2) are removed from the graph. Since both receivers at node 0 are used, we remove all the receiver edges, i.e., edges RxE(0,1) and RxE(0,2), which means that node 0 cannot sink lightpaths any more. After the traffic is routed onto lightpath L_2, the capacity of lightpath edge LPE(2,0) is 36 units. In this case, we set up one lightpath using two wavelength-links. Since the connection traverses a single lightpath, we call this approach *single-hop grooming*. Fig. 3(a) and (c) shows the current virtual topology and the updated auxiliary graph, respectively.
- **Case 2 (Multihop grooming).** Another path is along the edges *TxE*(2,1), WLE(2,1,1), RxE(1,1), GrmE(1), MuxE(1), LPE(1,0), and DmxE(0), shown as bold lines in Fig. 4(b). This path contains edges WLE(2,1,1) and LPE(1,0), which denote wavelength λ_1 on the fiber-link from node 2 to node 1 and the lightpath from node 1 to node 0, respectively. If choosing this path, we need to set up a lightpath L_3 from node 2 to node 1 using wavelength λ_1 on the fiber-link from node 2 to node 1, and a lightpath edge LPE(2,1) is added and wavelength-link edge WLE(2,1,1) removed. Then, we route T_2 onto the newly setup lightpath L_3 and the pre-existing lightpath L_1. The capacities of lightpath edge LPE(2,1) and LPE(1,0) are

[1] Here we use comparison sorts, such as heapsort and merge sort, to determine the order of the connections, whose running time is $O(D\log D)$. A linear-time sorting algorithm, such as counting sort, radix sort, and bucket sort, can also be applied to determine the order of the connections. Then, the running time of INGPROC is $O(DN^2W^2)$ and $O(DN^2)$, assuming no wavelength-conversion capability and full wavelength-conversion capability, respectively.

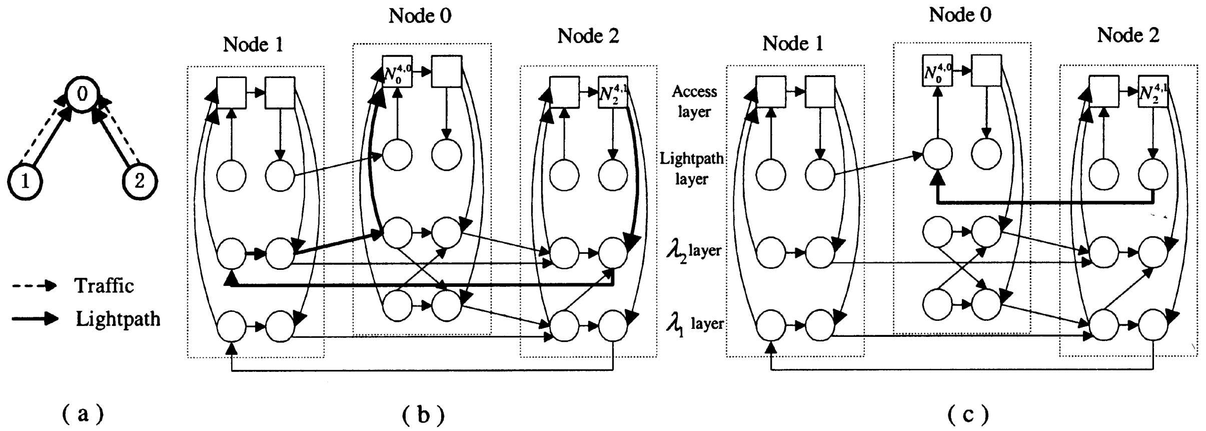


Fig. 3. (a) Virtual topology of network 1. (b) Corresponding auxiliary graph before routing the second traffic request T_2. (c) Corresponding auxiliary graph after routing the second traffic request T_2 using single-hop grooming.

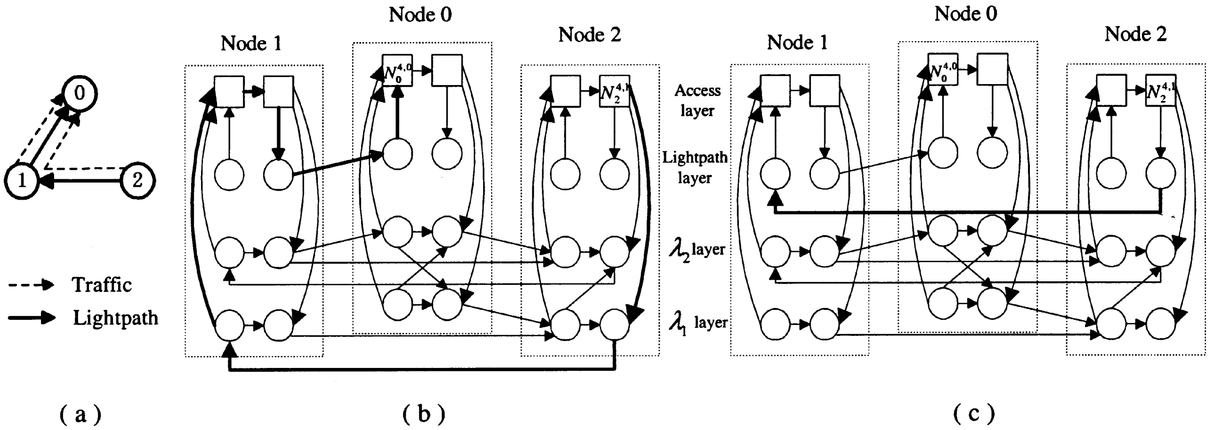


Fig. 4. (a) Virtual topology of network 1. (b) Corresponding auxiliary graph before routing the second traffic request T_2. (c) Corresponding auxiliary graph after routing the second traffic request T_2 using multihop grooming.

36 and 12, respectively. In this case, we have to route the connection onto two lightpaths, but only one more wavelength-link is required for satisfying this traffic. Since the connection traverses multiple lightpaths, we call this approach *multihop grooming*. However, this kind of multihop grooming will add burden on the electrical devices, which are the bottleneck and major cost in a WDM network, at the intermediate node(s) (node 1 in this case). Fig. 4(a) and (c) shows the current virtual topology and the updated auxiliary graph, respectively.

Which path should be chosen depends on the grooming policy. Since the IGABAG algorithm chooses the shortest path, the grooming policy should be reflected in the weight-assignment function. We will discuss the grooming policies in Section IV-A.

Suppose the third traffic demand T_3 is T(1, 0, OC-48, 1). If we use single-hop grooming for the second connection, we cannot find a path from $N_1^{4,1}$ to $N_0^{4,0}$ after removing all the lightpath edges since they cannot accommodate this traffic request and it will be blocked. However, if we use multihop grooming for the second connection, we can still find a path in the graph since there is a wavelength λ_2 available which can be used to set up a lightpath from node 1 to node 0 to carry the T_3 traffic.

From the above example, it can be seen that the IGABAG algorithm can deal with the grooming problem in the blocking manner under the wavelength and transceiver constraints using different grooming policies. It is a straightforward matter to verify that our algorithm can be used in nonblocking scenarios if enough network resources are given for the traffic demands.

One of the advantages of the graph model is that, if a path is obtained from the source to the destination in the auxiliary graph, all the four subproblems of traffic grooming are solved simultaneously. Therefore, it can avoid the limitations introduced by having to solve the four subproblems separately.

IV. Grooming Policies and Weight Assignment

A grooming policy determines how to carry the traffic in a certain situation. It reflects the intentions of the network operator. In this section, we first analyze all the possible operations when routing a traffic request. The different ordering of the possible operations forms different grooming policies. Then, we discuss how to assign weights to edges in the auxiliary graph to achieve different grooming policies.

A. Grooming Policies

When solving the traffic-grooming problem, given a traffic demand $T(s, d, g, m)$, we need to determine how to route the traffic under the current network state.

TABLE I
COMPARISON OF FOUR OPERATIONS

	Add new lightpath(s)	Single-hop or multihop grooming
Operation 1	No	Single-hop
Operation 2	No	Multihop
Operation 3	Yes	Single-hop
Operation 4	Yes	Multihop

In general, for a traffic demand $T(s, d, g, m)$ in a network, there are four possible operations that can be used to carry the traffic without altering the existing lightpaths. Note that we do not consider reconfiguring existing lightpaths, such as splitting or rerouting a lightpath, since then the traffic on the network would be interrupted.

- *Operation 1*: Route the traffic onto an existing lightpath directly connecting the source s and the destination d.
- *Operation 2*: Route the traffic through multiple existing lightpaths.
- *Operation 3*: Set up a new lightpath directly between the source s and the destination d and route the traffic onto this lightpath. Using this operation, we set up only one lightpath if the amount of the traffic is less than the capacity of the lightpath.
- *Operation 4*: Set up one or more lightpaths that do not directly connect the source s and the destination d, and route the traffic onto these lightpaths and/or some existing lightpaths. Using this operation, we need to set up at least one lightpath. However, since some existing lightpaths may be utilized, the number of wavelength-links used to set up the new lightpaths is probably less than that of wavelength-links needed to set up a lightpath directly connecting the source s and the destination d.

The characteristics of these four operations are summarized in Table I.

Each operation must satisfy certain prerequisites before it can be applied. For instance, if there is no lightpath between the source and the destination of the traffic that can accommodate the traffic, then Operation 1 cannot be used. In some situations, all the operations are applicable, while in other situations, only some of them can be used. If none of them can be applied, the traffic must be blocked without reconfiguring the existing lightpaths.

In a situation where multiple operations can be applied, how to choose the operations is a matter of the grooming policy. By combining the various operations in different orders, we can achieve different grooming policies.

Here we present three different grooming policies. In each of them, for a given traffic demand $T(s, d, g, m)$, if there is a lightpath from s to d which can carry the traffic request, we always choose it since this is the best solution for the connection request, i.e., we always use Operation 1 when it is applicable.

- *Minimizing the Number of Traffic Hops (MinTH).*
  If Operation 1 fails, we always try to set up a lightpath from s to d and route the traffic onto this lightpath. Only when such a lightpath cannot be set up, we use multihop grooming. This policy is achieved by Operation 1 followed by Operation 3. After that, we will use Operations 2 and 4 and choose the one with fewer hops on the virtual topology.
- *Minimizing the Number of Lightpaths (MinLP).*
  This policy tries to set up the minimum number of *new* lightpaths to carry the traffic. If Operation 1 fails, we try to route the traffic using multiple existing lightpaths (Operation 2). If Operation 2 also fails, then we try to set up one or more lightpaths to accommodate the traffic using Operation 3 or Operation 4.
- *Minimizing the Number of Wavelength-Links (MinWL).*
  This policy tries to consume the minimum number of *extra* wavelength-links to carry the traffic. The difference between MinLP and MinWL is that, after Operations 1 and 2 fail, MinWL compares the number of wavelength-links used by Operations 3 and 4, and it chooses the one requiring fewer wavelength-links.

B. Weight Assignment

Policies MinTH, MinLP, and MinWL can be easily achieved by applying different weight-assignment functions to the graph model and using the IGABAG algorithm.

Since each grooming policy is achieved by combining the four operations in different ways, it is helpful to analyze the weight function of each operation first. Since IGABAG chooses the shortest path found in the auxiliary graph, the order in which the four operations are combined is determined by the relationship of the weight of each operation.

In the following discussion, the weight of each kind of edge is nonnegative and represented by the name of this kind of edge, e.g., the weight of a wavelength-link edge is represented by WLE. In addition, we assume that the same kind of edges has the same weight and there is no wavelength converter at each node, i.e., there is no converter edge in the auxiliary graph. It is straightforward to extend the discussion to more general cases.

- Operation 1 uses a single existing lightpath to route the traffic. Since each lightpath edge always has a mux and demux edge connected with it in the path, the weight of the path found in the auxiliary graph is
$$\text{MuxE} + \text{LPE} + \text{DmxE}. \tag{10}$$
- Operation 2 uses n ($n \geq 2$) existing lightpaths to carry the traffic. Since each lightpath edge always has a mux and demux edge connected with it and there is a grooming edge between two lightpaths, the weight of the path found in the auxiliary graph is
$$n \times (\text{MuxE} + \text{LPE} + \text{DmxE}) + (n-1) \times \text{GrmE}. \tag{11}$$
- Operation 3 sets up a lightpath between the source and the destination of the traffic and routes the traffic onto it. According to the IGABAG algorithm, the lightpath follows the path found in the graph, which consists of a transmitter edge, m ($m \geq 1$) wavelength-link edges, $m-1$ wavelength bypass edges, and a receiver edge. Therefore, the weight of the path found in the auxiliary graph is
$$\text{TxE} + m \times \text{WLE} + (m-1) \times \text{WBE} + \text{RxE}. \tag{12}$$

- Operation 4 sets up k ($k \geq 1$) lightpaths and routes the traffic onto them and k' ($k' \geq 0$) existing lightpaths. Supposing that each newly set up lightpath uses l_i ($l_i \geq 1, 1 \leq i \leq k$) wavelength-links, the weight of the path found in the auxiliary graph is

$$\begin{aligned}\sum_{i=1}^{k} (\mathrm{TxE} + l_i \times \mathrm{WLE} + (l_i - 1) \times \mathrm{WBE} + \mathrm{RxE}) \\ + k' \times (\mathrm{MuxE} + \mathrm{LPE} + \mathrm{DmxE}) \\ + (k + k' - 1) \times \mathrm{GrmE}. \qquad (13)\end{aligned}$$

To make Operation 1 the first choice among the four operations, we need to ensure that the value of (10) is always the least among the four expressions.

Based on the analysis of the weight of each operation, we can easily manipulate the weights of the edges to satisfy the different grooming policies, as follows.

- MinTH policy tries to carry the current traffic request using the minimum number of lightpath hops on the virtual topology. It can be observed that, for each traffic, there is a grooming edge following each hop on the virtual topology except the last one. At the same time, if a grooming edge is encountered in the path, the traffic must experience a hop on the virtual topology. Therefore, minimizing the traffic hops is equivalent to minimizing the number of grooming edges in the path found by IGABAG. Hence, we need to assign a large weight to the grooming edges such that the weight of a path containing n ($n \geq 1$) grooming edges is always greater than that of any path containing $n - 1$ grooming edges. We call this kind of edges *dominant edges* in the graph.
- MinLP policy tries to set up the minimum number of new lightpaths for the current traffic request. For each newly set up lightpath, there must be a transmitter edge and a receiver edge in the path according to which the lightpath is set up. In addition, if there are n ($n \geq 0$) transmitter edges and receiver edges in the path, n lightpaths must be set up according to IGABAG. Therefore, minimizing the number of lightpaths is equivalent to minimizing the number of transmitter edges and receiver edges. So we should make transmitter edges and receiver edges dominant edges in the graph.
- MinWL policy tries to use as few unused wavelength-links as possible to accommodate the current traffic request. It is straightforward to see that we can achieve this policy by making wavelength-link edges dominant edges.

In grooming policies MinTH, MinLP, and MinWL, some operations are always applied before others, and these policies can be easily achieved by our graph model. However, this does not mean that the graph model can only apply the four operations in fixed orders. The order in which the four operations are performed by our model really depends on the weight assignment. If we appropriately assign weights to the edges in the auxiliary graph, the four operations can be applied in different orders at different times, which gives the network operator the maximum flexibility. For example, if we assign the weight to each edge according to the cost of the corresponding component, say, the weight of a TxE/RxE is the cost of a transmitter/receiver, the weight of a CvtE is the cost of a converter, the weight of a WLE is the cost of the corresponding wavelength-links, etc., the graph model will choose the most cost-effective operation to route a connection, under the current network state. This intrinsic flexibility that the operations can be applied in different orders is one of the major advantages of our model.

Note that our model can also be used for virtual-topology design. Suppose we are given the physical fiber topology, the maximal configuration of the nodes, and the traffic demands to be supported. Now, if we assign the weight to each edge according to the cost of the corresponding component, after routing all the traffic using the graph model, we can determine the virtual topology and the configuration of each node, such as the number of transceivers and converters at each node. Hence, our model can be used for network design while minimizing its total cost.

In dynamic grooming, the network state varies as connection requests come and go. To achieve good performance, the grooming policy should be evolved according to the current network state. For instance, if transceivers are becoming scarce resource, we should make full use of existing lightpaths to accommodate the new traffic and avoid setting up new lightpaths. The graph model can easily satisfy this requirement by adjusting the weights of edges according the current network state, i.e., the weight of an edge can be made a function of the network state. This capability of easily changing grooming policies makes the graph model very suitable for dynamic traffic grooming.

V. Numerical Examples

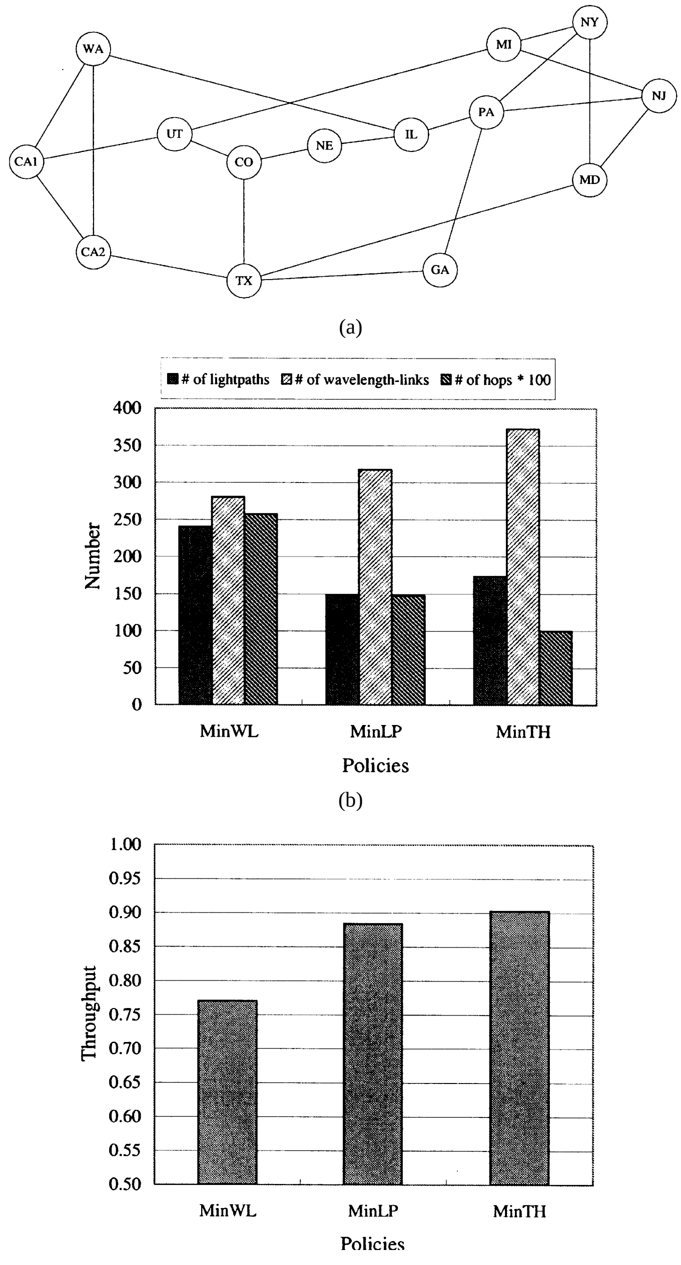
In this section, we first compare the performance of the three grooming policies MinWL, MinLP, and MinTH under the blocking and the nonblocking scenarios. Then, we compare the performance of the INGPROC procedure when using different traffic-selection schemes LCF, MAF, and MUF with the optimal solution obtained via an ILP in a relatively small network. Finally, the performance of the three traffic-selection schemes is investigated for a larger practical-sized network to demonstrate how the network throughput changes under different configurations.

A. Comparison of Grooming Policies

We compare the performance of these three grooming policies via simulation. The topology we used is the NSF network topology, which has 14 nodes and 21 links, as shown in Fig. 5(a). Each link is bidirectional and carries 32 wavelengths, and the capacity of each wavelength is OC-192. All nodes have grooming capability and there are 32 tunable transceivers and no wavelength converter at each node.

The traffic is randomly generated and uniformly distributed among all node pairs. For each node pair, there may exist several types of connections simultaneously, for instance, OC-3, OC-12, OC-48, and OC-192. The distributions of each type of connections are independent. For this example, the traffic is generated as follows.

- For each node pair (i, j), the probability that there is an OC-3 type of connection between them is 0.3. If there is an OC-3 connection between i and j, the amount m of the



(a)

(b)

(c)

Fig. 5. Comparison of different grooming policies. (a) NSF network. (b) Comparison of different grooming policies using a nonblocking model. (c) Comparison of different grooming policies using a blocking model.

traffic $T(i,j,OC-3,m)$ is uniformly distributed between 1 and 32.

- For each node pair (i,j), the probability that there is an OC-12 type of connection between them is 0.3. If there is an OC-12 connection between i and j, the amount m of the traffic $T(i,j,\text{OC-}12,m)$ is uniformly distributed between 1 and 16.
- For each node pair (i,j), the probability that there is an OC-48 type of connection between them is 0.3. If there is an OC-48 connection between i and j, the amount m of the traffic $T(i,j,\text{OC-}48,m)$ is uniformly distributed between 1 and 8.
- For each node pair (i,j), the probability that there is an OC-192 type of connection between them is 0.05. If there is an OC-192 connection between i and j, the amount m of the traffic $T(i,j,\text{OC-}192,m)$ is uniformly distributed between 1 and 2.

TABLE II
AVERAGE TRAFFIC GENERATED FOR THE NSF NETWORK

	# of connections	Amount in terms of OC-1
OC-3	54	2689
OC-12	55	5641
OC-48	54	11309
OC-192	8	2131

TABLE III
WEIGHTS OF EDGES ASSIGNED IN THE EXPERIMENTS FOR THE THREE GROOMING POLICIES

	MinTH	MinLP	MinWL
WLE	10	10	1000
GrmE	1000	20	0
TxE	20	200	20
RxE	20	200	20
LPE	1	1	1
MuxE	0	0	0
DmxE	0	0	0
WBE	0	0	0

In our simulation experiments, ten different traffic matrices randomly generated according to the above distribution are used, and the average traffic distribution is shown in Table II. On average, the total number of connection requests in a traffic matrix is 171 and the total traffic amount is equivalent to 21770 OC-1's. The network resources are enough to carry all the traffic demands, so this is a nonblocking model and the objective is to minimize the resources used to carry all of the traffic.

In Step 2 of INGPROC, we use the traffic-selection heuristic LCF to choose the traffic demand, which is discussed in Section III-B. In the experiments, we assign weights to the edges such that all the requirements for the grooming policies are satisfied. The weights of different edges assigned in the experiments for the three grooming policies are shown in Table III. Note that the grooming policies will be achieved as long as the relationship of the weights of the different edges satisfies the grooming-policy requirements, no matter what value a specific edge is assigned.

The results based on ten simulation experiments are shown in Fig. 5(b). It can be observed that, to carry all the traffic demands, the MinWL policy consumes the fewest wavelength-links, the MinLP policy sets up the minimum number of lightpaths, and the MinTH policy achieves the minimum number of average traffic hops on the virtual topology. This demonstrates that the weight-assignment functions of the three policies can really accomplish the corresponding grooming policies. In addition, the MinWL policy sets up the most number of lightpaths and the traffic experiences the largest number of hops on the virtual topology. This is because this policy prefers to use short lightpaths to carry connections. Since each lightpath will occupy one transmitter and one receiver at the source node and the destination node, respectively, the MinLP policy uses the fewest transceivers. The MinTH policy consumes the largest number of wavelength-links since it always tries to set up a lightpath from the source to the destination when the connection cannot be routed using a single existing lightpath.

We also study the performance of the three policies under a blocking scenario. The same NSF network topology and the same ten traffic matrices are used. However, each link now has only eight wavelengths, and each node has twelve tunable transceivers. Since the network resources are reduced and not all the requests can be satisfied, the objective in the blocking scenario is to maximize the carried traffic, i.e., the throughput of the network.

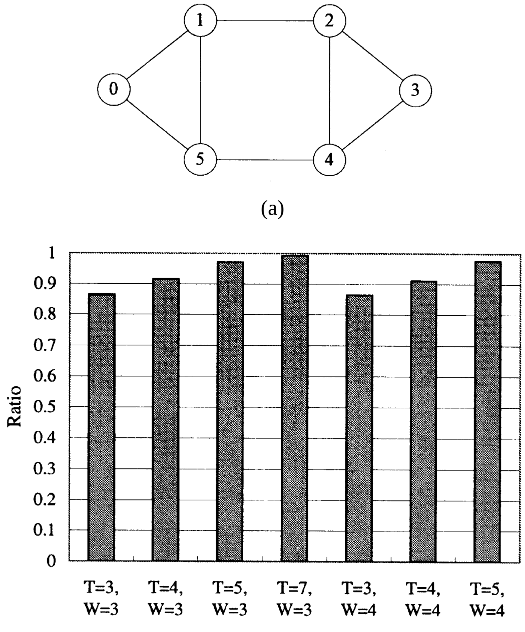
The results in Fig. 5(c) demonstrate that MinTH achieves the highest throughput among the three policies. This is because, in a blocking scenario, the network resources are limited. To improve the network throughput, we should use our limited resources efficiently, and single-hop grooming is usually more efficient to use lightpath capacity to carry the traffic than multihop grooming. From the perspective of a traffic request, fewer hops mean that less resources (lightpaths) are used to accommodate the traffic. From the perspective of a lightpath from node s to node d, its efficiency is higher when using the same amount of capacity to carry the traffic whose source and destination are also s and d than to carry other traffic.

B. Comparison of Traffic-Selection Schemes in a Relatively Small Network

We compare the performance of our heuristics with the optimal solution obtained through an ILP [3]. Since the ILP can be solved only for small networks, for this comparison, we use a six-node network with bidirectional links shown in Fig. 6(a), and the traffic matrices are as follows. There are three types of connections: OC-1, OC-3, and OC-12, and the amount of each connection type between each node pair is uniformly distributed between 0–16, 0–8, and 0–2, respectively. For our example, the total traffic amount becomes equivalent to 988 OC-1's. The capacity of each wavelength is OC-48. Each node has grooming capability with a limited number of transceivers and no wavelength converter. Since the network resources may not be enough to accommodate all the requests, our objective is to maximize the network throughput, i.e., the amount of successfully carried traffic. We use the MinTH policy in this experiment.

The results are shown in Table IV, where T denotes the number of transceivers at each node and W denotes the number of wavelengths per link. The numbers in the table are the percentage and the amount of the traffic routed using different traffic-selection schemes under different network configurations. We observe that the performance of LCF is better than those of the other two traffic-selection schemes in most cases and close or equal to the optimal solution.

To further compare the performance of our heuristics and that of the ILP, we tried twelve different traffic matrices with the same distribution as above. Fig. 6(b) shows the average ratio of the amount of routed traffic by heuristic LCF to the amount of routed traffic by the ILP, under different network configurations. It can be observed that our heuristic can achieve a near-optimal solution, while using much less running time than the ILP. (The running time of LCF is less than one second on machine A[2] while it takes several minutes to more than one hour for the ILP to get the solutions on the same machine.)



(a)

(b)

Fig. 6. Comparison of traffic-selection schemes in a relatively small network. (a) Network 2: a six-node network. (b) Average ratio of the amount of carried traffic by LCF to the amount of carried traffic by ILP.

TABLE IV
PERFORMANCE COMPARISON OF ILP AND DIFFERENT HEURISTICS FOR ROUTING STATIC TRAFFIC DEMANDS

	ILP	LCF	MUF	MAF
T=3, W=3	76.7% (758)	67.9% (671)	65.9% (651)	65.0% (642)
T=4, W=3	95.7% (946)	90.9% (898)	85.3% (843)	86.6% (856)
T=5, W=3	96.9% (957)	96.5% (953)	89.5% (884)	94.0% (929)
T=7, W=3	96.9% (957)	96.8% (956)	96.9% (957)	96.9% (957)
T=3, W=4	76.7% (758)	67.9% (671)	65.9% (651)	65.0% (642)
T=4, W=4	96.4% (952)	95.5% (944)	85.3% (843)	85.8% (848)
T=5, W=4	100% (988)	100% (988)	100% (988)	100% (988)

C. Comparison of Traffic-Selection Schemes in a Larger Representative Network

We also examine the heuristics on a larger representative network (Network 3) shown in Fig. 7(a). This network has 19 nodes and 31 links. All the nodes have grooming capability and no wavelength converter. Each link is bidirectional, and each wavelength has a capacity of OC-192. Our experiment results are based on ten different traffic matrices, which are randomly generated using the same method used for the NSF network. The average total number of connection requests is 330 and the average total traffic amount is equivalent to 42692 OC-1's. The distribution of the average generated traffic is shown in Table V.

The blocking model and MinTH policy are used in this experiment. We vary the number of transceivers at each node and the

[2] Machine A is a Windows PC with a 500-MHz Pentium III processor and 2-GB memory.

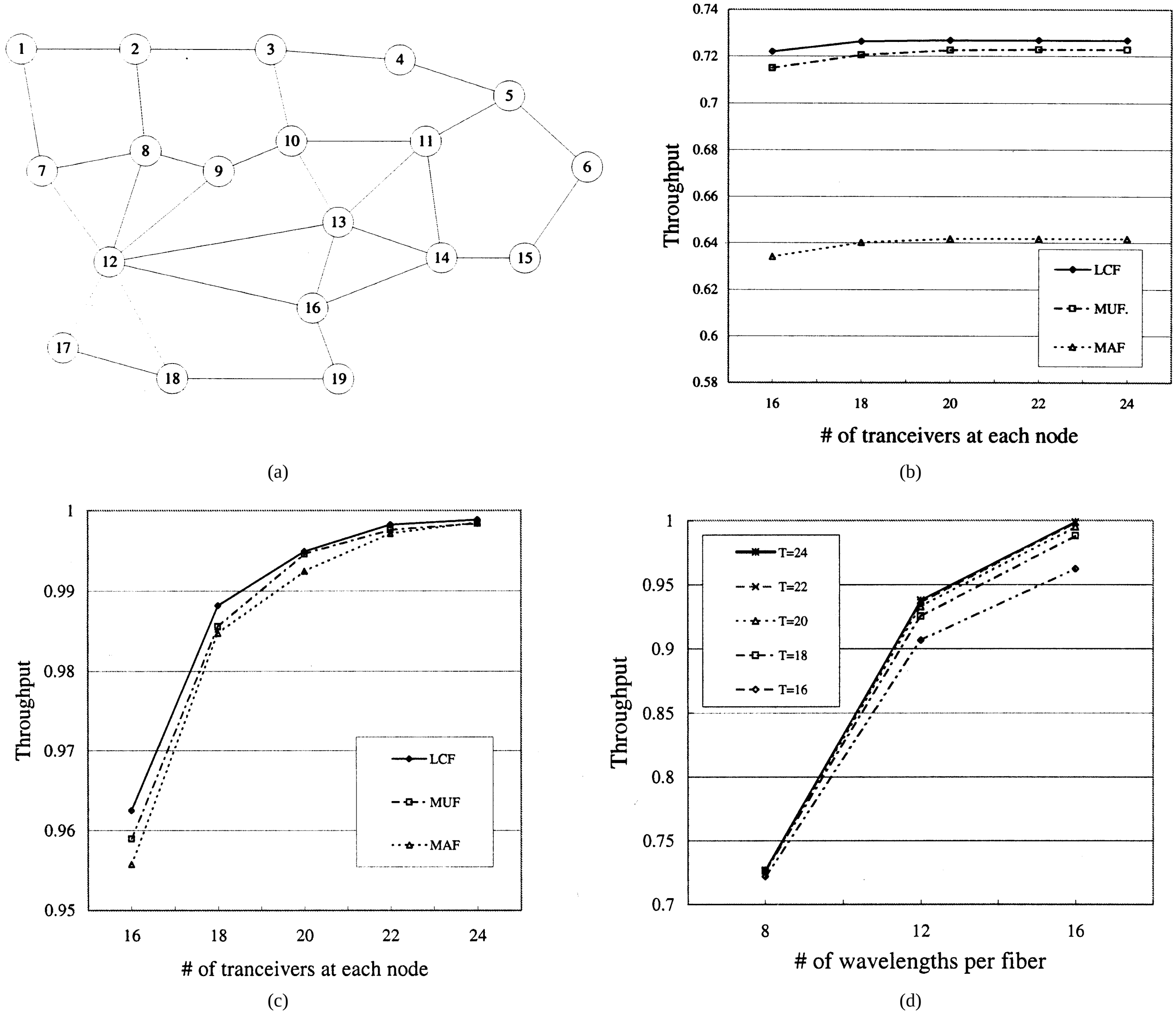


Fig. 7. Comparison of traffic-selection schemes in a larger representative network. (a) Network 3: a 19-node network. (b) Network throughput using different heuristics when each link has eight wavelengths. (c) Network throughput using different heuristics when each link has 16 wavelengths. (d) Network throughput using heuristic LCF under different network configurations.

TABLE V
TRAFFIC GENERATED FOR NETWORK 3

	# of connections	Amount in terms of OC-1
OC-3	106	5175
OC-12	105	10272
OC-48	103	22368
OC-192	17	4877

number of wavelengths on each link to obtain the performance of the three traffic-selection schemes under different network configurations. Fig. 7(b) and (c) shows, when using the heuristics LCF, MUF, and MAF, how the network throughput changes as the number of transceivers at each node varies from 16 to 24, with the assumption that each fiber-link has eight and 16 wavelengths, respectively. It can be observed that MUF performs better than MAF, and LCF performs best since LCF chooses the connection according to the current network state, while the other heuristics do not take this into account.

On the other hand, the time complexity of INGPROC when using selection scheme LCF is larger than when using MAF and MUF, as shown in Section III-B. In this experiment, the running time of LCF is about 1–5 min on machine B[3] while the running time of MUF and MAF are both within several seconds on the same machine. This is because, after routing each connection, LCF needs to recompute the cost of the remaining connection requests under the updated network state, and computing the cost of each connection needs to determine the route of the connection on the auxiliary graph, whose time complexity is $O(N^2W^2)$, while MUF and MAF do not need this step. Note that the ILP cannot practically solve the problem of this size. It reported not enough memory available when we ran the ILP for this example on machines A and B. Even if each link has only four wavelengths and there are only 60 connection requests, the ILP will have 40800

[3] Machine B is a Linux PC with a 1.7-GHz Pentium IV processor and 2-GB memory.

constraints and 1420597 variables and it could not obtain the solution within three days.

To further demonstrate the scalability of the IGABAG algorithm and the three traffic-selection heuristics, we conducted, on machine B, an experiment on a large nationwide network with 277 nodes. We assume that each node has 20 tunable transceivers and no wavelength converter, and there are 20 wavelengths on each link. The running time of the IGABAG algorithm is less than 0.3 s. To route 3496 connections, MUF and MAF take about 17 min, while LCF is estimated to run about 20 days. It can be observed from these results that the IGABAG algorithm scales well with network size and can even satisfy the complexity requirement for on-line provisioning in dynamic traffic grooming. For static traffic, MUF and MAF can handle a large amount of connection requests in a reasonable time period, while LCF has a scalability problem as the number of connection requests increases.

Finally, Fig. 7(d) shows that, when the LCF heuristic is used, the network throughput increases as the number of wavelengths on each link increases. When each link has only eight wavelengths, the throughput improves little as the number of transceivers is increased from 18 to 24. This is because the resource bottleneck is the number of wavelengths on each fiber-link rather than the number of transceivers at each node. For each node configuration, the network throughput increases as the number of wavelengths on each fiber increases.

VI. Conclusion

In this paper, we developed a novel generic graph model for traffic grooming in a heterogeneous WDM optical mesh network. This model takes into account various constraints, such as transceivers, wavelengths, wavelength-conversion capabilities, and grooming capabilities, which means that it can be applied very generally to a heterogeneous WDM mesh network environment. With static grooming (where all connections to be set up are known *a priori*), it can achieve various objectives by using different grooming policies under blocking and nonblocking scenarios. Moreover, our ability to easily adjust the grooming policy according to the current network state makes this model very suitable for dynamic traffic grooming (where connections need to be set up one at a time). At the same time, the versatility of the model is accomplished by using a uniform mechanism for manipulating the edges in the auxiliary graph, namely, by adjusting their weights and performing shortest-path computation, which is a very simple and novel feature of our proposed model.

Based on the auxiliary graph, we develop an integrated grooming algorithm, IGABAG, which jointly solves the four traffic-grooming subproblems for one traffic demand, and a grooming procedure, INGPROC, which can accommodate both static and dynamic traffic grooming using the IGABAG algorithm. Among the three proposed grooming policies MinTH, MinLP, and MinWL, our study shows that MinWL consumes the minimum wavelength-links and MinLP uses the minimum transceivers under nonblocking scenarios, while the traffic travels using the minimum number of hops on the virtual topology in nonblocking scenarios when MinTH is used, and MinTH achieves the maximum throughput under blocking scenarios. For static traffic grooming, the LCF heuristic outperforms MUF and MAF when combined with the INGPROC procedure, while MUF and MAF scale better than LCF as the number of connection requests increases.

References

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

[2] E. Modiano and P. Lin, “Traffic grooming in WDM networks,” *IEEE Commun. Mag.*, vol. 39, pp. 124–129, July 2001.

[3] K. Zhu and B. Mukherjee, “Traffic grooming in an optical WDM mesh network,” *IEEE J. Select. Areas Commun.*, vol. 20, pp. 122–133, Jan. 2002.

[4] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath communications: An approach to high bandwidth optical WANs,” *IEEE Trans. Commun.*, vol. 40, pp. 1171–1182, July 1992.

[5] B. Mukherjee, S. Ramamurthy, D. Banerjee, and A. Mukherjee, “Some principles for designing a wide-area optical network,” *IEEE/ACM Trans. Networking*, vol. 4, pp. 684–696, Oct. 1996.

[6] R. Ramaswami and K. Sivarajan, “Design of logical topologies for wavelength-routed optical networks,” *IEEE J. Select. Areas Commun.*, vol. 14, pp. 840–851, June 1996.

[7] A. Ganz and X. Wang, “Efficient algorithm for virtual topology design in multihop lightwave networks,” *IEEE/ACM Trans. Networking*, vol. 2, pp. 217–225, June 1994.

[8] R. Krishnaswamy and K. Sivarajan, “Design of logical topologies: A linear formulation for wavelength routed optical networks with no wavelength changers,” in *Proc. IEEE INFOCOM*, vol. 2, Mar. 1998, pp. 919–927.

[9] O. Gerstel, P. Lin, and G. H. Sasaki, “Combined WDM and SONET network design,” in *Proc. IEEE INFOCOM*, vol. 2, Mar. 1999, pp. 734–743.

[10] H. Zang, J. P. Jue, and B. Mukherjee, “A review of routing and wavelength assignment approaches for wavelength-routed optical WDM networks,” *SPIE Opt. Networks Mag.*, vol. 1, pp. 47–60, Jan. 2000.

[11] I. Chlamtac, A. Ganz, and G. Karmi, “Lightnets: Topologies for high speed optical networks,” *J. Lightwave Technol.*, vol. 11, pp. 951–961, May/June 1993.

[12] K. Zhu and B. Mukherjee, “A review of traffic grooming in WDM optical networks: Architectures and challenges,” *SPIE Opt. Networks Mag.*, to be published.

[13] A. L. Chiu and E. H. Modiano, “Traffic grooming in algorithms for reducing electronic multiplexing costs in WDM ring networks,” *J. Lightwave Technol.*, vol. 18, pp. 2–12, Jan. 2000.

[14] P. J. Wan, G. Calinescu, L. Liu, and O. Frieder, “Grooming of arbitrary traffic in SONET/WDM BLSRs,” *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1995–2003, Oct. 2000.

[15] X. Zhang and C. Qiao, “An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings,” *IEEE/ACM Trans. Networking*, vol. 8, pp. 608–617, Oct. 2000.

[16] O. Gerstel, R. Ramaswami, and G. H. Sasaki, “Cost-effective traffic grooming in WDM rings,” *IEEE/ACM Trans. Networking*, vol. 8, pp. 618–630, Oct. 2000.

[17] J. Simmons, E. Goldstein, and A. Saleh, “Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic,” *J. Lightwave Technol.*, vol. 17, pp. 48–57, Jan. 1999.

[18] R. Dutta and G. N. Rouskas, “On optimal traffic grooming in WDM rings,” *IEEE J. Select. Areas Commun.*, vol. 20, pp. 110–121, Jan. 2002.

[19] J. Wang, W. Cho, V. R. Vemuri, and B. Mukherjee, “Improved approaches for cost-effective traffic grooming in WDM ring networks: ILP formulations and single-hop and multihop connections,” *J. Lightwave Technol.*, vol. 19, pp. 1645–1653, Nov. 2001.

[20] J. Wang and B. Mukherjee, “Interconnected WDM ring networks: Strategies for interconnection and traffic grooming,” SPIE Opt. Networks Mag., to be published.

[21] R. Berry and E. Modiano, “Reducing electronic multiplexing costs in SONET/WDM rings with dynamically changing traffic,” *IEEE J. Select. Areas Commun.*, vol. 18, pp. 1961–1971, Oct. 2000.

[22] V. R. Konda and T. Y. Chow, “Algorithm for traffic grooming in optical networks to minimize the number of transceivers,” in *Proc. IEEE Workshop High Performance Switching and Routing*, May 2001, pp. 218–221.

[23] M. Brunato and R. Battiti, "A multistart randomized greedy algorithm for traffic grooming on mesh logical topologies," presented at the ONDM Working Conf., Feb. 2002.

[24] S. Thaigaran and A. Somani, "Capacity fairness of WDM networks with grooming capabilities," *SPIE Opt. Networks Mag.*, vol. 2, pp. 24–31, May/June 2001.

[25] R. Srinivasan and A. K. Somani, "A generalized framework for analyzing time-space switched optical networks," *IEEE J. Select. Areas Commun.*, vol. 20, pp. 202–215, Jan. 2002.

[26] K. Zhu and B. Mukherjee, "On-line approaches for provisioning connections of different bandwidth granularities in WDM mesh networks," in *Proc. OFC*, Mar. 2002, pp. 549–551.

[27] S. Thiagarajan and A. Somani, "Traffic grooming for survivable mesh networks," in *Proc. OPTICOMM*, Aug. 2001, pp. 54–65.

[28] L. A. Cox and J. Sanchez, "Cost savings from optimized packing and grooming of optical circuits: Mesh versus ring comparisons," *SPIE Opt. Networks Mag.*, vol. 2, pp. 72–90, May/June 2001.

[29] A. Lardies, R. Gupta, and R. A. Patterson, "Traffic grooming in a multi-layer network," *SPIE Opt. Networks Mag.*, vol. 2, pp. 91–99, May/June 2001.

[30] S. Subramaniam, M. Azizoglu, and A. K. Somani, "All-optical networks with sparse wavelength conversion," *IEEE/ACM Trans. Networking*, vol. 4, pp. 544–557, Aug. 1996.

[31] J. Iness and B. Mukherjee, "Sparse wavelength conversion in wavelength-routed WDM networks," *Photon. Netw. Commun. J.*, vol. 1, pp. 183–205, Nov. 1999.

[32] J. Yates, J. Lacey, D. Everitt, and M. Summerfield, "Limited-range wavelength translation in all-optical networks," in *Proc. IEEE INFOCOM*, vol. 3, Mar. 1996, pp. 954–961.

[33] R. Ramaswami and G. Sasaki, "Multiwavelength optical networks with limited wavelength conversion," *IEEE/ACM Trans. Networking*, vol. 6, pp. 744–754, Dec. 1998.

[34] K. R. Venugopal, M. Shivakumar, and P. S. Kumar, "A heuristic for placement of limited range wavelength converters in all-optical networks," in *Proc. IEEE INFOCOM*, vol. 2, Mar. 1999, pp. 908–915.

[35] T. Tripathi and K. N. Sivarajan, "Computing approximate blocking probabilities in wavelength routed all-optical networks with limited-range wavelength conversion," *IEEE J. Select. Areas Commun.*, vol. 18, pp. 2123–2129, Oct. 2000.

[36] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "Dynamic traffic grooming in WDM mesh networks using a novel graph model," *SPIE Opt. Networks Mag.*, to be published.

Hongyue Zhu (S'02) received the B.S. and M.S. degrees in computer science from Beijing University of Posts and Telecommunications, Beijing, China, in 1995 and 1998, respectively. He is currently working toward the Ph.D. degree in computer science at the University of California, Davis.

He is currently a Research Assistant in the Networks Laboratory, Department of Computer Science, University of California, Davis. He was with Sprint Advanced Technology Laboratories, Burlingame, CA, as a summer intern in 2001 and 2002. His research interests include traffic grooming in WDM networks, WDM network design and analysis.

Hui Zang (S'97–M'02) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1997, and the M.S. and Ph.D. degrees in computer science from the University of California, Davis, in 1998 and 2001, respectively.

In 2000, she joined Sprint Advanced Technology Laboratories, Burlingame, CA, where she is a Principal Applied R&D Engineer. She was one of the Guest Editors of *IEEE Network* special issue on "Traffic Engineering in Optical Networks." She is the author of the book *WDM Mesh Networks—Management and Survivability*, (Norwell, MA: Kluwer, 2002). She has published about 30 conference papers and journal articles and currently has three U.S. patents pending in the field of optical networking. Her research interests include WDM optical networks, IP over WDM networks and quality of service.

Dr. Zang serves or has served as a Technical Committee member of IPoW'00, IPoW'01, APOC'02, DRCN'03, OptiComm'03, and Globecom'03 Optical Networking and Systems Symposium. She also helped organize OptiComm'02 as a Panel Co-chair.

Keyao Zhu (S'98) received the B.S. degree from Beijing University, Beijing, China, in 1998 and the M.S. degree from University of California, Davis, in 2000. He is currently working toward the Ph.D. degree in computer science at the University of California, Davis.

He is currently a Research Assistant in the Networks Laboratory, Department of Computer Science, University of California, Davis. His research interests include wavelength-routed WDM network design and analysis, WDM network protection, restoration, and traffic grooming.

Biswanath Mukherjee (S'82–M'87) received the B.Tech. (Hons) degree from the Indian Institute of Technology, Kharagpur, India, in 1980 and the Ph.D. degree from the University of Washington, Seattle, in 1987.

In 1987, he joined the University of California, Davis, where he has been a Professor of computer science since 1995. He served as Chairman of the Department of Computer Science from 1997 to 2000. He is a Member of the Board of Directors of IPLocks Inc., a Silicon Valley startup company. He has consulted for and served on the Technical Advisory Board of a number of startup companies in optical networking. He has served on the editorial boards of the *ACM/Baltzer Wireless Information Networks (WINET)*, the *Journal of High-Speed Networks*, *Photonic Network Communications*, and *Optical Network Magazine*. He is the author of the textbook *Optical Communication Networks* (New York: McGraw-Hill, 1997), a book which received the Association of American Publishers, Inc. 1997 Honorable Mention in Computer Science. His research interests include lightwave networks, network security, and wireless networks.

Dr. Mukherjee is a corecipient of paper awards presented at the 1991 and 1994 National Computer Security Conferences. He has served on the editorial boards of the IEEE/ACM TRANSACTIONS ON NETWORKING and *IEEE Network*. He has also served as Editor-at-Large for optical networking and communications for the IEEE Communications Society, and as the Technical Program Chair of the IEEE INFOCOM 1996 conference. While at the University of Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship.