# On Hierarchical Traffic Grooming in WDM Networks

Bensong Chen, George N. Rouskas, *Senior Member, IEEE*, and Rudra Dutta, *Member, IEEE*

***Abstract*—The traffic grooming problem is of high practical importance in emerging wide-area wavelength division multiplexing (WDM) optical networks, yet it is intractable for any but trivial network topologies. In this work, we present an effective and efficient hierarchical traffic grooming framework for WDM networks of general topology, with the objective of minimizing the total number of electronic ports. At the first level of hierarchy, we decompose the network into clusters and designate one node in each cluster as the hub for grooming traffic. At the second level, the hubs form another cluster for grooming intercluster traffic. We view each (first- or second-level) cluster as a *virtual star*, and we present an efficient near-optimal algorithm for determining the logical topology of lightpaths to carry the traffic within each cluster. Routing and wavelength assignment is then performed directly on the underlying physical topology. We demonstrate the effectiveness of our approach by applying it to two networks of realistic size, a 32-node, 53-link topology and a 47-node, 96-link network. Comparisons to lower bounds indicate that hierarchical grooming is efficient in its use of the network resources of interest, namely, electronic ports and wavelengths. In addition to scaling to large network sizes, our hierarchical approach also facilitates the control and management of multigranular networks.**

***Index Terms*—Hierarchical traffic grooming, K-center, optical networks, wavelength division multiplexing (WDM).**

## I. INTRODUCTION

TRAFFIC grooming is the field of study that is concerned with the development of algorithms and protocols for the design, operation, and control of networks with multigranular bandwidth demands. The objective of traffic grooming techniques is to ensure that subwavelength traffic components are transported over the network in an efficient and cost-effective manner. Interest in such techniques has grown steadily in the research community in recent years, reflecting the practical issues arising from the ever-increasing capacity of wavelength channels and the cost associated with terminating optical signals at intermediate nodes. For a comprehensive survey and classification of traffic grooming research, the reader is referred to [10].

Traffic grooming research has, in general, followed one of two directions. In *dynamic* grooming [30], it is assumed that the node grooming capabilities (in terms of available electronic ports, level of wavelength conversion, and switching capacity) are fixed and known, and the goal is to develop online algorithms for grooming and routing of connection requests that arrive in real time. Typical solution approaches transform the grooming problem into a shortest path problem on a new layered graph modeling both the underlying physical topology and the capabilities of individual nodes.

In *static* grooming, the starting point is the set of (forecast) long-term traffic demands, and the objective is to provision the network nodes to carry all the demands while minimizing the overall network cost. The cost metric frequently considered in the literature is the total number of electronic ports required to originate and terminate the lightpaths created to carry the traffic components. Early research in this area focused on ring topologies [9], [12], [28], mainly due to the practical importance of upgrading the existing SONET infrastructure to support multiple wavelengths. As backbone networks migrate from ring to mesh topologies, traffic grooming in general topology networks is becoming the subject of an increasing number of studies [17], [18], [20]–[22]. Most studies provide an integer linear programming (ILP) formulation as the basis for reasoning about and tackling the problem. Unfortunately, solving the ILP directly does not scale to instances with more than a handful of nodes, and consequently it cannot be applied to networks of practical size covering a national or international geographical area. Consequently, either the ILP is tackled using standard relaxation techniques, or the problem is decomposed into subproblems which are solved using heuristics.

The traffic grooming algorithms presented in [17] and [31] are representative of existing traffic grooming approaches. The heuristics in [31] are developed independently of the ILP formulation, and work as follows. First, the traffic demands are sorted in some order (e.g., decreasing order of traffic amount or decreasing order of resource utilization). Then, each traffic demand is considered in this order, and an attempt is made to establish a direct lightpath to carry this demand, subject to wavelength and transceiver constraints. If a lightpath can be established, a path for it is obtained using dynamic routing, and a wavelength is assigned using the first-fit policy. Finally, traffic demands for which a lightpath could not be set up are routed over the established logical topology using a similar dynamic routing algorithm. The work in [17], on the other hand, tackles the traffic grooming problem by working directly with the ILP formulation. As a first step, the original ILP is decomposed into two simpler ILPs. The first ILP addresses only the traffic grooming and routing subproblem, and is solved first. The second ILP addresses the wavelength assignment problem only, and uses the solution to the first ILP. The two ILPs are solved sequentially, and because of the smaller size of the simpler ILPs, this approach is considerably faster than solving the original ILP directly. In general, this decomposition method is not optimal, however, it will yield an optimal solution if the network satisfies a certain condition derived in [17].

As we can see, existing heuristic approaches regard the network as a flat entity for the purposes of lightpath routing, wavelength assignment, and traffic grooming. It is well-known, however, that in existing networks, resources are typically managed

Manuscript received August 23, 2005; revised September 21, 2006. First published March 12, 2008; current version published October 15, 2008. Approved by IEEE/ACM TRANSACTIONS ON NETWORKING Editor F. Neri. This work was supported by the National Science Foundation under Grant ANI-0322107.

B. Chen is with Google, Inc., Mountain View, CA 94043 USA (e-mail: bensong@google.com).

G. N. Rouskas and R. Dutta are with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8206 USA (e-mail: rouskas@ncsu.edu; rdutta@ncsu.edu).

Digital Object Identifier 10.1109/TNET.2007.906655

and controlled in a hierarchical manner. The levels of the hierarchy either reflect the underlying organizational structure of the network or are designed in order to ensure scalability of the control and management functions. Based on this observation, in this work we develop a hierarchical framework for static traffic grooming in mesh networks with the objective of minimizing the total number of electronic ports in the network. (Since a lightpath requires exactly two electronic ports, one at the source and one at the destination, this objective is equivalent to minimizing the number of lightpaths in the logical topology.) While we also decompose the traffic grooming problem into several subproblems, the novelty of our works lies in adopting a hierarchical decomposition which scales well to networks of realistic size, and is more compatible with the manner in which networks are controlled and managed.

Our approach emulates the hub-and-spoke model used by the airline industry to "groom" passenger traffic onto connecting flights. At the first level of the hierarchy, the network is partitioned into clusters, and one node in each cluster (referred to as the *hub*) is responsible for grooming intracluster traffic as well as intercluster traffic originating or terminating locally. At the second level of the hierarchy, the first-level hubs form another cluster for grooming and routing intercluster traffic. The logical topology within a (first- or second-level) cluster is formed by viewing it as a *virtual star*, and applying a customized algorithm for stars which we develop. Finally, a routing and wavelength assignment (RWA) algorithm is used on the underlying topology to route and color the lightpaths.

Our approach has the following desirable characteristics.

- It is hierarchical, facilitating control, management, and security functions.
- It decouples the grooming of traffic components into lightpaths from the routing and wavelength assignment for these lightpaths: grooming is performed on a logical hierarchy of clusters by abstracting each cluster as a virtual star, and applying efficient and near-optimal algorithms; while RWA is performed directly on the underlying physical topology, ensuring efficient use of network resources.
- It provisions only a few nodes (the hubs) for grooming traffic they do not originate or terminate.
- It handles efficiently small traffic demands: at the first level of hierarchy, nodes pack their traffic on lightpaths to the local hub; at the second level, demands among remote clusters are packed onto lightpaths between the corresponding hubs.
- It routes large traffic components on direct lightpaths, eliminating the cost of terminating and switching them at intermediate nodes.

Hierarchical clustering techniques are common in network design, but so far they have been considered in the context of traffic grooming only tangentially. A case for hierarchical approaches in the design of SONET rings was first made in [12], and more recently in [11]. In mesh networks, the blocking island paradigm for tackling a restricted version of traffic grooming was advocated in [7]; this paradigm allows for the abstraction of network resources, and can be applied recursively on the network graph. Our approach is more comprehensive than those in [7], [11], and [12], and is quite general, in the sense that it can be extended to a wide range of variants of the grooming problem. Finally, the concept of "supernodes" in [25] implements a hierarchical approach similar to ours, but is specific to ring networks only.

The rest of the paper is organized as follows. In Section II, we define the traffic grooming problem and present a high-level view of our approach. In Section III, we present an algorithm for traffic grooming in networks with a star physical topology. In Section IV, we present a hierarchical grooming algorithm for mesh networks that utilizes the star grooming algorithm of Section III. We present numerical results in Section V, and we conclude the paper in Section VI.

## II. PROBLEM DEFINITION AND METHODOLOGY

We consider a network of $N$ nodes interconnected by fiber links such that the resulting topology is of general form. Without loss of generality, we assume that each link consists of one fiber per direction, and each fiber can carry $W$ wavelengths simultaneously. We let $C$ be a positive integer denoting the capacity of each wavelength channel, expressed in units of a basic transmission rate (such as OC-3). The capacity $C$ has also been variously called the *grooming factor*, or *granularity*. We assume the existence of a traffic demand matrix $T = [t^{(sd)}]$, where integer $t^{(sd)}$ denotes the amount of (forecast) long-term traffic to be carried from node $s$ to node $d$; consequently, any changes in the demand matrix take place over long time scales, and, for the purposes of this work, the matrix $T$ is assumed fixed. Finally, we allow the traffic demands to be greater than the capacity of a wavelength, i.e., it is possible that $t^{(sd)} > C$ for some $s$, $d$.

Given the forecast traffic demands $\{t^{(sd)}\}$, our objective is to dimension the network to carry the traffic matrix in its entirety by using the minimum number of electronic ports at the network nodes. Similar to other traffic grooming studies, we do not consider the optical port cost in this paper; we note, however, that by minimizing the number of lightpaths (equivalently, electronic ports), our approach does reduce the number of optical ports required to establish these lightpaths. A formulation of this traffic grooming problem as an integer linear problem (ILP) is omitted, but is available in [10]. The problem involves the following conceptual subproblems (SPs): 1) *logical topology SP:* find a set $R$ of lightpaths that forms a logical topology, i.e., a topology in which the lightpaths form the edges between the $N$ nodes; 2) *lightpath routing and wavelength assignment (RWA) SP:* solve the RWA problem on $R$, i.e., assign a wavelength and path over the physical topology to each lightpath in $R$; and 3) *traffic routing SP:* route each traffic component $t^{(sd)}$ through the lightpaths in $R$. This is only a conceptual decomposition that helps in understanding and reasoning about the problem; in an optimal approach, the subproblems would be considered together in the solution. The first and third subproblems together constitute the grooming aspect of the problem.

The above traffic grooming problem defined on a general topology is NP-hard, even when the RWA subproblem is taken out of the picture [8]. Next, we outline our hierarchical approach to traffic grooming in general topologies.

### A. Hierarchical Approach to Traffic Grooming

Our approach borrows ideas from the hub-and-spoke paradigm that is widely used within the airline industry. Specifically, we assume that the network is partitioned into clusters (or islands) of nodes, where each cluster consists of nodes in a contiguous region of the network. The clusters may correspond to independent administrative entities (e.g., autonomous systems), or may be created solely for the purpose of simplifying resource management and control functions (e.g., as in partitioning a single OSPF administrative domain into multiple areas).

For the purposes of traffic grooming, we view each cluster as a *virtual star*, and we designate one node as the *hub* of the cluster. We refer to each cluster as a *virtual star* because, even though the physical topology of the cluster may take any form (and in fact may be quite different than a *physical star* topology), the hub is the only node responsible for grooming intra- and inter-cluster traffic. Consequently, hub nodes are expected to be provisioned with more resources (e.g., larger number of electronic ports and higher switching capacity for grooming traffic) than nonhub nodes. Returning to the airline analogy, a hub node is similar in function to airports that serve as major hubs; these airports are typically larger than nonhub airports, in terms of both the number of gates ("electronic ports") and physical space (for "switching" passengers between gates).

The main idea behind our hierarchical grooming strategy is to solve the first and third subproblems of the traffic grooming problem (i.e., construct the logical topology and determine the routing of traffic components on it) in two steps. In the first step, we apply the StarTopology algorithm we describe in the next section to each cluster; the result of this step is a set of lightpaths within each cluster to route local (intracluster) traffic, as well as intercluster traffic, to and from the local hub. In the second step, we view all the hub nodes as forming a second-level virtual star, and we apply the StarTopology algorithm once more to determine the lightpaths and corresponding routing for intercluster traffic. Finally, given the above collection of inter- and intra-cluster lightpaths, we solve the RWA problem on the underlying physical topology of the network. We provide a detailed description of this hierarchical grooming algorithm in Section IV.

To illustrate our approach, let us consider the 32-node network in Fig. 1. The bottom part of the figure shows a partition of the network into eight clusters, $B_1, \ldots, B_8$, each cluster consisting of four nodes. These clusters represent the first level of the hierarchy. Within each cluster, one node is the hub; for instance, node 2 is the hub for cluster $B_1$. The top part of the figure shows the second-level cluster, consisting of the hub nodes of the eight first-level clusters; one of these nodes, say, node 13, is selected as the hub node for the second-level cluster. We emphasize that, while we view each cluster as a virtual star, the actual physical topology of the cluster is determined by the physical topology of the part of the original network where the cluster nodes lie; for example, the four nodes of cluster $B_8$ form a ring. Since the RWA algorithm is performed on the underlying physical topology *after* the logical topology has been determined, the lightpaths will follow the most efficient paths in the network, despite the fact that the StarTopology algorithm was developed for physical stars (see the next section). Consider, for example, cluster $B_8$ with node 28 as its hub. Suppose that the logical topology obtained by running the StarTopology algorithm on the corresponding *virtual star* with node 28 as the hub, includes the "one-hop" lightpath (32,28) and the "two-hop" lightpath (31,32). After running the RWA algorithm, the "one-hop" lightpath may be routed over the path 32-30-28 (since node 32 is not directly connected to the hub node 28 of the virtual star), while the "two-hop" lightpath may in fact be routed over the direct link 31-30, completely bypassing the hub node 28 (unlike a physical star where a two-hop lightpath is optically switched at the hub). Similar observations apply to all clusters at both levels of the hierarchy.

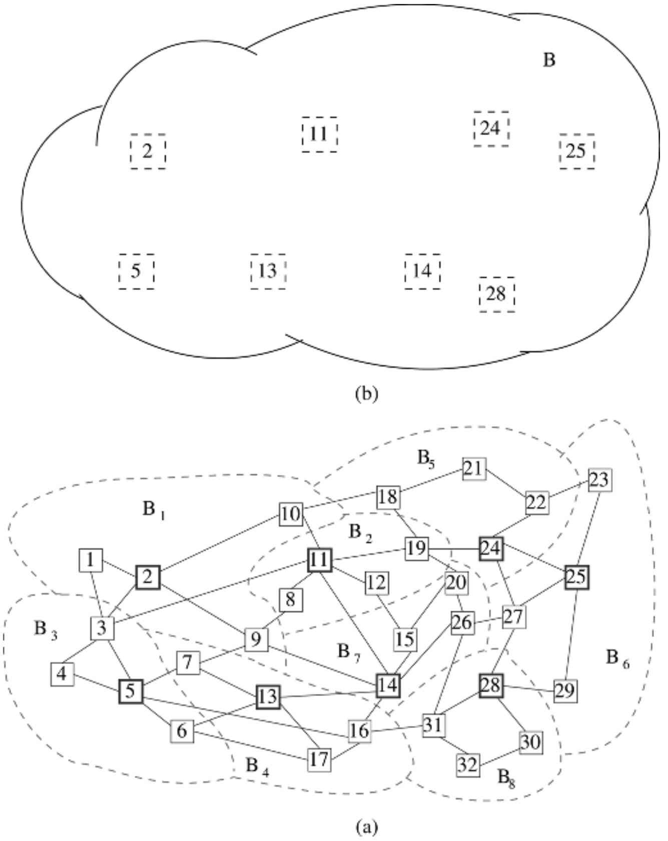


Fig. 1. A 32-node WDM network, its partition into eight first-level clusters $B_1, \ldots, B_8$, and second-level cluster $B$ consisting of the eight first-level hubs. (a) First-level clusters (b) Second-level cluster consisting of first-level hubs, and hub node 13.

## III. TRAFFIC GROOMING IN PHYSICAL STAR NETWORKS

Consider a network of $N$ nodes, a central *hub* node and $N-1$ nodes, each connected to the hub over a bidirectional fiber link that can carry $W$ wavelengths in each direction. Let $C$ be the wavelength capacity and $T = [t^{(sd)}]$ the traffic demand matrix. Nonhub nodes are not allowed to groom or switch nonlocal traffic, either optically or electronically, and all traffic switching and grooming is performed at the hub node. In a physical star, the routing of traffic components (i.e., a solution to the third grooming subproblem) is implicit in the logical topology: traffic is routed over the corresponding two-hop lightpath bypassing the hub, if such a lightpath exists; otherwise it is packed onto single-hop lightpaths to the hub and then to its destination. Regarding the second grooming subproblem, the routing of lightpaths is also implicit in the logical topology, and wavelength assignment can be performed in polynomial time [29]. However, the first grooming subproblem (i.e., determining the set

**Logical Topology Algorithm for Star Networks**

**Input:** A star network with $N$ nodes, capacity $C$ of each wavelength, and traffic matrix $T = [t^{(sd)}]$.

**Output:** The set of lightpaths $R$ in the logical topology such that $|R|$ is minimized, and the number $W$ of wavelengths needed

```
procedure StarTopology
begin
      // Set up direct lightpaths
  1.  Reduction: create direct lightpaths for demands $\geq C$
  2.  $T_r = [t_r^{(sd)}] \leftarrow$ residual traffic matrix, $t_r^{(sd)} < C \ \forall \ s, d$
  3.  Create single-hop lightpaths to carry the residual traffic by electronically switching (grooming) it at the hub
  4.  $U_0 \leftarrow$ number of lightpaths in current logical topology
  5.  Sort all the residual traffic demands $t_r^{(sd)}$ between non-hub nodes $s$ and $d$ in non-increasing order, and label them as $t_1, t_2, \ldots, t_k, k = (N-1)^2$
  6.  $i \leftarrow 1$; // iteration index
  7.  while $t_i > 0$ do
  8.      Create a new two-hop lightpath to route $t_i$ directly from source to destination, if doing so does not violate any wavelength constraints
  9.      $U_i \leftarrow$ number of lightpaths in new logical topology
  10.     $i \leftarrow i + 1$
  11. end while // $i = m$ at this point
  12. Find the smallest of $U_0, U_1, \ldots, U_m$, and return the corresponding logical topology $R$ and the maximum number of wavelengths on any link as the solution
end
```

Fig. 2. Logical topology algorithm for star networks.

of single-hop and two-hop lightpaths) so as to minimize the number of electronic ports in the network remains NP-hard [4].

We have developed the greedy StarTopology algorithm for solving the first grooming subproblem (logical topology design) in physical stars. In the following, we explain the operation of the StarTopology algorithm, a pseudocode description of which is provided in Fig. 2. We first *reduce* the original traffic matrix $T$ by assigning direct lightpaths to all traffic demands $t^{(sd)}$ that can fill a wavelength. After reduction, the residual traffic demands to be groomed are less than the wavelength capacity $C$, for each source-destination pair. Then, we obtain an initial solution by first carrying all such demands on single-hop lightpaths to the hub, electronically grooming them there, and then carrying them on single-hop paths to their respective destinations. In this manner, traffic is packed as tightly as possible onto lightpaths that traverse only one physical link.

The initial all-electronic solution is generally not optimal with respect to minimizing the number of lightpaths, because all lightpaths are very short (single-hop). Intuitively, it would be possible to reroute traffic demands between nonhub nodes onto direct lightpaths that bypass the hub node, to create longer (two-hop) lightpaths; doing so is desirable if the creation of a two-hop lightpath leads to the elimination of two single-hop lightpaths, decreasing the total number of lightpaths. However, if such direct lightpaths carry only a small amount of traffic compared with the wavelength capacity $C$, this approach may not lead to a better solution. Although finding the optimal set of nonhub demands for which to set up direct lightpaths is NP-hard (since the star grooming problem is NP-hard [4]), intuition suggests that a greedy approach of assigning lightpaths to the largest traffic demands will work well in practice.

Steps 5–11 of the algorithm perform the greedy assignment of lightpaths. At each iteration, we check whether creating a direct two-hop lightpath for the largest traffic component currently routed over two single-hop lightpaths would violate the wavelength constraint $W$. If so, we do nothing; otherwise, we create the new two-hop lightpath and remove any single-hop lightpaths for which this was the only traffic component they carried. We continue in this manner, recording the total number of lightpaths after every iteration, until no additional two-hop lightpaths can be created. Among all the logical topologies created at the end of each iteration, the algorithm returns the one with the smallest number of lightpaths as the solution. It is straightforward to see that the time complexity of the star grooming algorithm is $O(N^2)$.

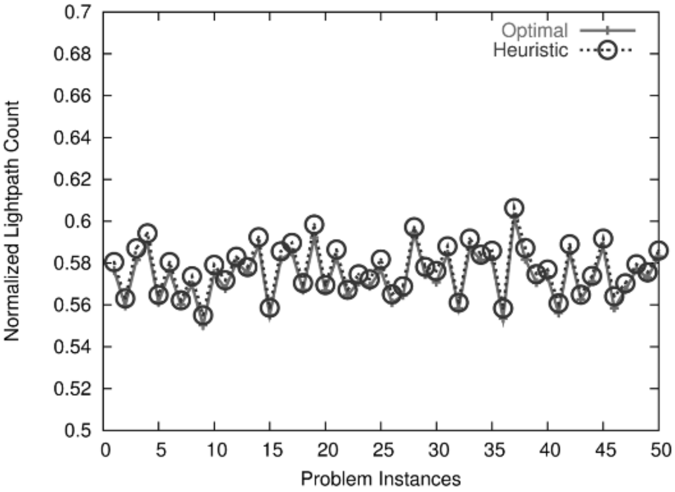


Fig. 3. Grooming effectiveness of the StarTopology algorithm, $N = 10$.

Fig. 3 presents the results of one representative experiment with stars of size $N = 10$; this is the largest number of nodes for which we were able to use CPLEX to obtain the optimal solution to the corresponding ILP. The objective of this ILP is also to minimize the number of lightpaths, and the formulation, which is available in [4], is a specialized version of the formulation for general topologies in [10] adapted to star networks. Fig. 3 plots the *normalized lightpath count* of the optimal solution and the solution obtained by the StarTopology algorithm, for 50 problem instances with $N = 10$ and random traffic patterns (please refer to Section V for a description of how the traffic matrices are generated under the random traffic pattern). The normalized lightpath count is defined as the ratio of the number of lightpaths in a solution of the traffic grooming problem over the number of lightpaths required by the all-electronic solution (i.e., when all traffic is switched electronically at the hub). This normalized value allows us to compare results among problem instances with very different traffic matrices; obviously, a smaller value implies a better solution. As we can see, the solution obtained by our algorithm tracks the optimal solution closely over all 50 problem instances. In absolute terms, our algorithm gives results that are at most four lightpaths more than the optimal; the average difference is 2.96 lightpaths, which is less than 1% of the optimal values for these instances. For a comprehensive evaluation of the StarTopology algorithm over a wide range of problem instances, please refer to [4].

## IV. HIERARCHICAL GROOMING IN MESH NETWORKS

We now present the details of our hierarchical grooming approach for networks with a general topology. Our primary objective is to minimize the number of lightpaths in the logical topology; however, we are also interested in keeping the number of required wavelengths low.

The hierarchical grooming algorithm consists of three phases.

1) **Clustering and hub selection.** Partition the network into $k$ clusters and designate one node in each cluster as the hub.
2) **Logical topology design and traffic routing.** During this phase, the first and third subproblems of the traffic grooming problem are solved in an integrated manner. This phase is further subdivided into three parts:
   a) setup of direct lightpaths for large traffic demands;
   b) intracluster traffic grooming;
   c) intercluster traffic grooming.

   The outcome of this phase is a set $R$ of lightpaths for carrying the traffic demand matrix $T$, and a routing of individual traffic components $t^{(sd)}$ over these lightpaths.
3) **Routing and wavelength assignment.** Each of the lightpaths in $R$ is assigned a wavelength and path on the underlying physical topology of the original mesh network.

The following subsections discuss each of the three phases of the algorithm in depth.

### A. Clustering and Hub Selection

The objective of this phase is twofold. First, we partition the network nodes into some number $k$ of clusters, denoted $B_1, \ldots, B_k$. Second, we select one node in each cluster to serve as the hub where grooming of intra- and intercluster traffic is performed. Let $n_i$ denote the number of nodes in cluster $B_i$, $n_1 + n_2 + \cdots + n_k = N$, and $h_i$ denote the hub of cluster $B_i$.

Clearly, the number of clusters, their composition, and the corresponding hubs must be selected in a way that helps achieve our goal of minimizing the number of lightpaths and wavelengths required to carry the traffic demands. Therefore, the selection of clusters and hubs is a complex and difficult task, as it depends on both the physical topology of the network and the traffic matrix $T$. To illustrate this point, consider the tradeoffs involved in determining the number $k$ of clusters.[1] If $k$ is very small (but greater than one), the amount of intercluster traffic generated by each cluster will likely be large. Hence, the $k$ hubs may become bottlenecks, resulting in a large number of electronic ports at each hub and possibly a large number of wavelengths (since many lightpaths may have to be carried over the fixed number of links to/from each hub). On the other hand, a large value for $k$ implies a small number of nodes within each cluster. In this case, the amount of intracluster traffic will be small, resulting in inefficient grooming (i.e., a large number of lightpaths); similarly, at the second-level cluster, $O(k^2)$ lightpaths will have to be set up to carry small amounts of intercluster traffic. We have been able to capture this tradeoff analytically in the case of uniform traffic demands; the reader is referred to [4, Ch. 6.2] for the details.

Clustering is a function that arises frequently in problems related to network design and organization. A classic book [14] defines clustering as "grouping of similar objects," and discusses many mainstream clustering algorithms. The algorithms are classified as either *minimum cut* or *spanning tree*, depending on the underlying methodology. The input to the algorithms generally consist of a set of nodes and edge weights, while the output is a partition of the nodes that optimizes a given objective function.

Some clustering studies only consider the communication (traffic) pattern between nodes. For instance, an algorithm that can group a nearly completely decomposable (NCD) matrix into blocks, so that the weighted arcs between blocks have values not exceeding a given threshold, was introduced in [6]. The algorithm, called TPABLO, can be used to group the states of large Markov chains. A similar objective exists in the traffic grooming context, as it is desirable for traffic demands within a cluster to be "denser" than intercluster traffic. However, the TPABLO algorithm does not take into account the physical topology; hence it may group together nodes that are far apart, creating clusters that are inappropriate for the hierarchical logical topology we consider. Other work has focused on the physical topology only. Typically, the goal is to partition the nodes into contiguous clusters containing roughly equal numbers of nodes, and at the same time minimize the overall cut size. An example is the work in [23] on multiobjective graph partitioning, which was implemented in the METIS software package. These algorithms were designed for VLSI design, a very different problem, where equality in size and a minimum of cross-layer connections are essential for each module, and are not directly applicable to traffic grooming.

Another family of clustering problems concerned with the physical network topology includes the well-known *K-Center, K-Clustering, K-Median,* and *Facility Location* problems [1], [16], [24], [27]. Unlike the applications targeted by METIS, they do not require clusters to be of equal size. Of all the variants, the *K-Center* problem is of most interest to us. The goal of the *K-Center* problem is to find a set $S$ of $K$ nodes (centers) in the network, so as to minimize the maximum distance from any network node to the nearest center. Thus, the set $S$ implicitly defines $K$ clusters with corresponding hub nodes in $S$.

A solution to the *K-Center* problem may be useful for hierarchical traffic grooming since it is likely to lead to short lightpaths within a cluster, thus requiring fewer wavelengths. Also, this type of clustering tends to avoid physical topologies with a large diameter for each cluster; such topologies are not a good

---

[1] Note that in the special case of $k = 1$, there is a single cluster with one hub and $N - 1$ nonhub nodes, whereas in the special case $k = N$, there are $N$ clusters, each with a single hub and no nonhub nodes.

match for the StarTopology algorithm which treats each cluster as a *virtual star*.

The *K-Center* problem is NP-Complete, and the best approximation ratio that can be obtained in polynomial time is 2 [13], [15]. We use the 2-approximation algorithm in [13] for *K-Center*; for completeness, its steps are listed below.

1) Create a single cluster, $B_1 = \{v_1, \dots, v_N\}$, with hub node $h_1 = v_1$. Calculate the all-pair shortest paths, record the distances in matrix *dist*, and let $x \leftarrow 1$.
2) Let $x$ be the number of clusters, and $d$ be the maximum distance between any node and its hub, i.e., $d = \max\{\mathrm{dist}(v_i, h_j)\}$, $v_i \in B_j$. Let $v$ be a node such that the distance between $v$ and its hub is $d$.
3) Create a new cluster $B_{x+1}$ with $h_{x+1} = v$ as the only node. Then for each node $v'$, if $v'$ is closer to $v$ than to its current hub, move $v'$ from its current cluster to the new cluster $B_{x+1}$. Let $x \leftarrow x + 1$.
4) Repeat Steps 2) and 3) $K - 1$ times, adding one cluster at each iteration, for a total of $K$ clusters.

We note that Step 1) is the preprocessing step when the computation of the all-pair shortest paths is performed; given these paths, the main steps of the algorithm, Steps 2)–4), take time $O(NK)$.

The *K-Center* problem takes only the physical topology as input, and its only goal is to minimize the maximum node-to-hub distance. In the traffic grooming context, on the other hand, hub capacity and lightpath routing should also be considered. Specifically, observe that hubs are responsible for originating and terminating a larger number of lightpaths than nonhub nodes. Therefore, in order to lower the wavelength requirements for the network as a whole, it is generally desirable to select as hubs the nodes with the largest bandwidth capacity, i.e., those with the largest physical degree. Based on this observation, we make the following modification on the above *K-Center* algorithm.

- At any time in Steps 1) or 2) a new hub must be selected, we select the node with the maximum physical degree among all candidate nodes [i.e., all the nodes in Step 1), and all the nodes having the same maximum distance in Step 2)].

We use this modified *K-Center* algorithm in the clustering phase of our hierarchical grooming framework.

### *B. Logical Topology Design and Traffic Routing*

*1) Setup of Direct Lightpaths For Large Traffic Demands:* During this step, we first reduce the traffic matrix $T$ by assigning direct lightpaths to all traffic demands $t^{(sd)}$ that are greater than the wavelength capacity $C$, even if nodes $s$ and $d$ belong to different clusters. Since carrying $C$ units of traffic from source $s$ to the local hub, then to the remote hub (if different), and finally to the destination $d$, would require two or three lightpaths, setting up direct lightpaths for such demands is preferable given our goal of minimizing the total number of lightpaths in the logical topology.

Following the reduction step, we also apply a "direct to the destination hub" rule to set up lightpaths between some node $s$ and a remote hub $h$, if the total amount of traffic from $s$ to nodes $d$ in $h$'s cluster $\sum_d t^{(sd)} \geq p \times C$, where the value of parameter $p$ denotes the minimum fraction of the wavelength capacity required to setup such a direct lightpath. The value of parameter $p$ is configured by the network designer; in our work, we let $p = 0.8$. Setting up such lightpaths for large demands to bypass the local hub node (i.e., the hub in the cluster of node $s$) has several benefits: the number of lightpaths in the logical topology is reduced, the number of electronic ports and switching capacity required at hub nodes is reduced (leading to higher scalability), and the RWA algorithm may require fewer wavelengths (since hubs will be less of a bottleneck).

Let $R_{\text{init}}$ be the set of direct lightpaths created in this step. Let $T_r = \left[t_r^{(sd)}\right]$ denote the matrix of residual traffic demands (i.e., excluding those carried by the lightpaths in $R_{\text{init}}$) that need to be groomed. Obviously, $t_r^{(sd)} < C$ for all $s, d$. Next, we concentrate on setting up lightpaths to groom the demands $\left\{t_r^{(sd)}\right\}$.

*2) Intracluster Traffic Grooming:* Consider the $i$th cluster $B_i$ with $n_i$ nodes, one of which, say, node $h_i$, is designated as the hub. We view cluster $B_i$ as a *virtual star* with a $n_i \times n_i$ traffic matrix $T_i = \left[t_i^{(sd)}\right]$, defined as

$$t_i^{(sd)} = \begin{cases} t_r^{(sd)}, & s \neq h_i, d \neq h_i \\ t_r^{(sd)} + \sum\limits_{x \notin B_i} t_r^{(sx)}, & d = h_i \\ t_r^{(sd)} + \sum\limits_{x \notin B_i} t_r^{(xd)}, & s = h_i. \end{cases} \tag{1}$$

In other words, if $s$ and $d$ are nonhub nodes, then $t_i^{(sd)}$ represents the intracluster traffic from $s$ to $d$. If, on the other hand, node $d$ (respectively, node $s$) is the hub node, then $t_i^{(sd)}$ includes not only the intracluster traffic component $t_r^{(sd)}$, but also the aggregate intercluster traffic originating at node $s$ (respectively, terminating at node $d$). This definition of $t_i^{(sd)}$ when either $s$ or $d$ are the hub node, implements the hierarchical grooming of traffic: all intercluster traffic, other than that carried by direct lightpaths set up earlier, is first carried to the local hub, groomed there with intercluster traffic from other local nodes, carried on lightpaths to the destination hub (as we discuss shortly), groomed there with other local and nonlocal traffic, and finally carried to the destination node.

Given traffic matrix $T_i = \left[t_i^{(sd)}\right]$, we view cluster $B_i$ as a *virtual star* with hub $h_i$ and $n_i - 1$ nonhub nodes. We apply the StarTopology algorithm in Fig. 2 to obtain the set of lightpaths $R_i$ for carrying the demands $\left\{t_i^{(sd)}\right\}$. Recall that the lightpaths in $R_i$ are either "single-hop" (i.e., from a nonhub node to the hub, or vice versa), or "two-hop" (i.e., from one nonhub node to another). Hence, the routing of the traffic components $t_i^{(sd)}$ is implicit in the logical topology $R_i$, as we explained in Section III.

We emphasize that, at this stage, we only identify the lightpaths to be created; the routing of these lightpaths over the physical topology is performed later. Depending on the actual topology of the cluster $B_i$, which may be quite different than that of a physical star, once routed, the lightpaths in $R_i$ may follow paths that do not resemble at all the paths of a physical star. For instance, a "one-hop" lightpath from a nonhub node of the cluster to the hub $h_i$ is routed on the unique link from the node to the hub in a physical star; in our case, however, the path followed by the lightpaths may consist of several

```
Logical Topology Algorithm for Mesh Networks
Input: A mesh WDM network with $N$ nodes partitioned in $k$ clusters $B_1, \cdots, B_k$, hub $h_i$ of cluster $B_i$, capacity $C$ of each wavelength, and traffic matrix $T = [t^{(sd)}]$
Output: The set of lightpaths $R$ in the logical topology, the routing of the traffic components $t^{(sd)}$, such that $|R|$ is minimized
Procedure MeshTopology
begin
    // Set up direct lightpaths
 1. Reduction: create direct lightpaths for demands $\geq C$
 2. Direct to destination hub: create lightpaths to hub nodes when the aggregate traffic to a cluster is large ($\geq 0.8 \times C$)
 3. $R_{init} \leftarrow$ initial set of direct lightpaths
    // Intra-cluster grooming
 4. $T_r = [t_r^{(sd)}] \leftarrow$ residual traffic matrix
 5. for $i = 1, \cdots, k$ do
 6.     $T_i = [t_i^{(sd)}] \leftarrow$ intra-cluster traffic matrix for cluster $B_i$, computed from expression (1)
 7.     $R_i \leftarrow$ set of lightpaths obtained by running the StarTopology algorithm on virtual star $B_i$ with hub $h_i$
 8. end for
 9. $R_{intra} \leftarrow R_1 \cup R_2 \cup \cdots \cup R_k$
    // Inter-cluster grooming
10. $B \leftarrow$ cluster consisting of $k$ hub nodes $h_1, \cdots, h_k$
11. $h \leftarrow$ hub of cluster $B$
12. $T_{inter} \leftarrow$ the $k \times k$ inter-cluster matrix from expression (2)
13. $R_{inter} \leftarrow$ set of lightpaths obtained by running the StarTopology algorithm on virtual star $B$ with hub $h$
14. Return the set of lightpaths $R = R_{init} \cup R_{intra} \cup R_{inter}$
end
```

Fig. 4. Logical topology algorithm for mesh networks.

links, depending on the physical topology of the network and the RWA algorithm (which we discuss in a moment). Similarly, a "two-hop" lightpath is always switched optically at the hub of a physical star; in a *virtual star* cluster, on the other hand, a "two-hop" lightpath will be routed by the RWA algorithm on the actual underlying topology, and its path may not even pass through the hub $h_i$ at all, if doing so is more efficient in terms of resource usage (e.g., if the two nonhub nodes are connected by a direct link).

We perform intracluster grooming in this manner, by applying the StarTopology algorithm to each cluster $B_1, \ldots, B_k$, in isolation. As a result, at the end of this step, we identify a set of lightpaths $R_{\text{intra}} = R_1 \cup R_2 \cup \cdots \cup R_k$ for carrying all intracluster traffic.

*3) Intercluster Traffic Grooming:* At the end of intracluster grooming, all traffic (other than that carried by the initial direct lightpaths) from the nodes of a cluster $B_i$ with destination outside the cluster, is carried to the hub $h_i$ for grooming and transport to the destination hub. In order to groom this traffic, we consider a new cluster $B$ that forms the second-level hierarchy in our approach. Cluster $B$ consists of the $k$ hub nodes $h_1, \ldots, h_k$, of the first-level clusters. Let $h \in \{h_1, \ldots, h_k\}$ be the node designated as the second-level hub. We view cluster $B$ as a *virtual star* with a $k \times k$ traffic matrix $T_{\text{inter}} = \left[t_{\text{inter}}^{(h_i h_j)}\right]$ representing the intercluster traffic demands. This intercluster matrix is defined as

$$t_{\text{inter}}^{(h_i h_j)} = \sum_{s \in B_i, d \in B_j} t_r^{(s,d)}, \qquad i,j = 1, \ldots, k,\ i \neq j. \quad (2)$$

We now apply the StarTopology algorithm in Fig. 2 to the *virtual star* $B$ with hub $h$, and we obtain the set of lightpaths $R_{\text{inter}}$ to carry the traffic demands $\left\{t_{\text{inter}}^{(h_i h_j)}\right\}$. Again, we emphasize that the routing of these lightpaths is performed on the underlying physical topology. Thus, the same observations regarding the routing of the intracluster lightpaths above also apply to the lightpaths in $R_{\text{inter}}$.

Fig. 4 provides a pseudocode description of the hierarchical logical topology algorithm. The time complexity of the algorithm is determined by the application of the StarTopology algorithm for intra- and intercluster grooming in Steps 5–8 and 13, respectively. The **for** loop in Steps 5–8 is executed $k$ times, where $k$ is the number of first-level clusters. During the $i$th iteration of the loop, the StarTopology algorithm is run on a cluster of size $n_i$, taking time $O\left(n_i^2\right)$. Since $n_i > 1$ and $n_1 + \cdots + n_k = N$, we have that $N \leq n_1^2 + \cdots + n_k^2 \leq N^2$; hence the **for** loop takes time $O(N^2)$. Step 13 calls the StarTopology algorithm on the second-level cluster with $k$ nodes, taking time $O(k^2)$. Since $k < N$, the overall complexity of the algorithm is $O(N^2)$.

Finally, we note that we considered only two levels of clusters in our grooming algorithm. However, for networks of very large size, our approach can be extended to three or more levels of hierarchy in a straightforward manner.

### C. Routing and Wavelength Assignment

The outcome of the logical topology design phase is a set of lightpaths $R = R_{\text{init}} \cup R_{\text{intra}} \cup R_{\text{inter}}$, and an implicit routing of the original traffic components $t^{(sd)}$ over these lightpaths. Our objective is to route the lightpaths in $R$ over the underlying physical topology, and color them using the minimum number of wavelengths. The RWA problem on arbitrary network topologies has been studied extensively in the literature [3], [5], [17], [18], [26]. In this work, we adopt the LFAP algorithm [26] which is fast, conceptually simple, and has been shown to use a number of wavelengths that is close to the lower bound. For completeness, we now describe the main steps of the LFAP algorithm.

1) Calculate a shortest path for all source-destination pairs for which a direct lightpath must be set up. List the lightpaths in $R$ in nonincreasing order of the length of their shortest path. Let the current wavelength $w \leftarrow 1$.
2) Consider each lightpath in the ordered list, and assign wavelength $w$ and the corresponding precomputed shortest path to as many lightpaths as possible; remove these lightpaths from the list.
3) Remove from the network topology all the links carrying lightpaths assigned wavelength $w$ in the previous step. Consider the lightpaths remaining in the ordered list and compute a new shortest path on the new topology. Assign wavelength $w$ and the corresponding new shortest path to as many lightpaths as possible. Remove the lightpaths that have been assigned a path and wavelength from the ordered list, and restore the original network topology.
4) If the ordered list of lightpaths is empty, stop and return $w$; otherwise, set $w \leftarrow w + 1$ and repeat from Step 2).

## V. NUMERICAL RESULTS

In this section, we present experimental results to demonstrate the performance of our hierarchical grooming algorithm. We experiment with two network topologies: the 32-node, 53-link network shown in Fig. 1, and a larger 47-node, 96-link topology. In order to evaluate the hierarchical grooming approach under various cluster sizes and study the tradeoffs involved, we use the modified *K-Center* algorithm in Section IV-A to partition the networks into different sets of clusters, and compare the results.

The traffic matrix $T = [t^{(sd)}]$ of each problem instance we consider is generated by drawing $N(N-1)$ random numbers (rounded to the nearest integer) from a Gaussian distribution with a given mean $t$ and standard deviation $\sigma$ that depend on the traffic pattern. To illustrate that our approach works well under different traffic patterns, we consider three patterns in our study.

1) *Random pattern.* We have found that random patterns are often challenging in the context of traffic grooming, since the matrix does not have any particular structure that can be exploited by a grooming algorithm. To generate a traffic matrix for a problem instance, we let the standard deviation of the Gaussian distribution be 150% of the mean $t$. Consequently, the traffic elements $t^{(sd)}$ take values in a wide range around the mean, and the loads of individual links also vary widely. If the random number generator returns a negative value for some traffic element, we set the corresponding $t^{(sd)}$ value to zero.
2) *Falling pattern.* This pattern is such that nodes physically close to each other exchange more traffic than nodes far away from each other. Specifically, if the mean of the distribution for node pairs that have shortest distance 1 is $t$, then the mean for node pairs with shortest distance 2 (respectively, 3) is set to $0.8t$ (respectively, $0.6t$); for all other pairs, the mean is set to $0.2t$. We also let the standard deviation $\sigma$ be 20% of the mean.
3) *Rising pattern.* This is the opposite of the falling pattern. We use $t$ as the mean for traffic demands between node pairs with the longest path. Node pairs with the second and third longest paths have mean values $0.8t$ and $0.6t$, respectively. All other node pairs have mean $0.2t$. The standard deviation is also set to 20% of the mean.

As we stated earlier, our grooming objective is to minimize the total number of lightpaths (equivalently, number of electronic ports) in the network, while keeping the number of wavelengths required to establish these lightpaths low. In order to characterize the performance of the solutions generated by our hierarchical traffic grooming algorithm, we have obtained lower bounds on the number of lightpaths and wavelengths necessary to carry a given traffic demand matrix. In order to be able to compare results among different problem instances, we use two performance metrics to evaluate the effectiveness of our hierarchical traffic grooming algorithm: the *normalized lightpath count* and the *normalized wavelength count*. For a given problem instance, we first compute the lower bound $lp_l$ on the number of lightpaths, and then run our hierarchical grooming algorithm and obtain the number $lp_h$ of lightpaths used in that solution. The normalized lightpath count is computed as the ratio $lp_h/lp_l \geq 1$, and this value is the one we plot in the figures presented in this section.[2] Clearly, the closer this value is to one, the closer our solution is to the optimal. The normalized wavelength count is computed in a similar manner.

Next, we describe how to obtain bounds on the number of lightpaths and wavelengths that are *independent* of the manner (e.g., hierarchical or otherwise) in which traffic grooming is performed; for additional details, the reader is referred to [4].

*1) Lightpath Lower Bounds:* A simple lower bound can be calculated based on the observation that each node must source and terminate a sufficient number of lightpaths to carry the traffic demands from and to this node, respectively. This bound can be determined directly from the traffic matrix $T$. However, we obtain a better lower bound based on the following observations. Let $b_{sd}$ denote the number of direct lightpaths set up from $s$ to $d$. Since all traffic originating at source node $s$ must be carried on some lightpath also originating at $s$, the following constraints must be observed:

$$\sum_d b_{sd}C \geq \sum_d t^{(sd)} \quad \forall\, s. \tag{3}$$

Similarly, for each destination $d$ we have that

$$\sum_s b_{sd}C \geq \sum_s t^{(sd)} \quad \forall\, d. \tag{4}$$

We can obtain a lower bound on the total number of lightpaths by solving the following ILP:

Minimize: $\sum_{s,d} b_{sd}$
Subject to: Constraints (3) and (4).

We emphasize that the above ILP will not necessarily yield a meaningful solution to the original grooming problem, only a lower bound. By configuring CPLEX to use dual steepest-edge pricing, we are able to compute this bound within a few seconds even for the 47-node topology we consider later in this section. Although this bound is better than the simple bound above, we believe that it is somewhat loose. However, we have found that introducing additional constraints on traffic flow and/or routing

---

[2] Note that this definition of normalized lightpath count is different from the one we used in Section III: there we normalize with respect to the cost of the all-electronic solution, while here we normalize with respect to the lower bound.

into the ILP in order to improve the lower bound tends to increase substantially the running time of CPLEX, to the point that it becomes impractical for the large networks we consider in this work. Therefore, we use the lower bound obtained from the above simple ILP in our experimental study to compute the normalized lightpath count.

*2) Wavelength Lower Bound:* Consider a bisection cut of the network, and let $t$ be the maximum amount of traffic that needs to be carried on either direction of the links in the cut set. Let $x$ be the number of links in the cut set, and $C$ the capacity of each wavelength. Then, the quantity $\lceil t/xC \rceil$ is a lower bound on the number of wavelengths for carrying the given traffic matrix. We use the METIS software [19] to find a small-cut bisection of the network such that the number of nodes at each side of the bisection is roughly equal. With such a bisection, it is likely that a large amount of traffic will traverse the cut, resulting in a tighter (higher) lower bound on the wavelength requirements. We also note that computing this bound does not require any information regarding the logical topology or the routing and wavelength assignment of lightpaths.

### A. The 32-Node, 53-Link Network

Let us first consider the 32-node, 53-link network topology shown in Fig. 1. We used the following methodology to obtain the results we present in this section. We employed the modified *K-Center* algorithm in Section IV-A to obtain three different clusterings of this network, with two, four, and eight clusters, respectively; we also consider the special case of a single cluster comprising all network nodes. For each traffic pattern, we generated 30 problem instances (i.e., random traffic matrices following the given pattern). Finally, for each of the 120 instance-clustering pairs corresponding to a traffic pattern, we: 1) used the Mesh-Topology algorithm in Fig. 4 to determine the hierarchical logical topology for grooming the traffic demands; 2) applied the LFAP algorithm [26] to route and assign a wavelength to each lightpath required; and 3) computed a lower bound on the number of lightpaths and wavelengths, as we described above, from which we determined the normalized lightpath and wavelength count.

Figs. 5 and 6 and Table I present experimental results for the random traffic pattern. Figs. 5 and 6 plot the normalized lightpath and wavelength count, respectively, for each problem instance and corresponding clustering, while Table I presents aggregate statistics over all 30 problem instances regarding the average lightpath length, the average maximum hub degree (i.e., the maximum of the number of incoming or outgoing wavelengths at the hub), and the average number of wavelengths.

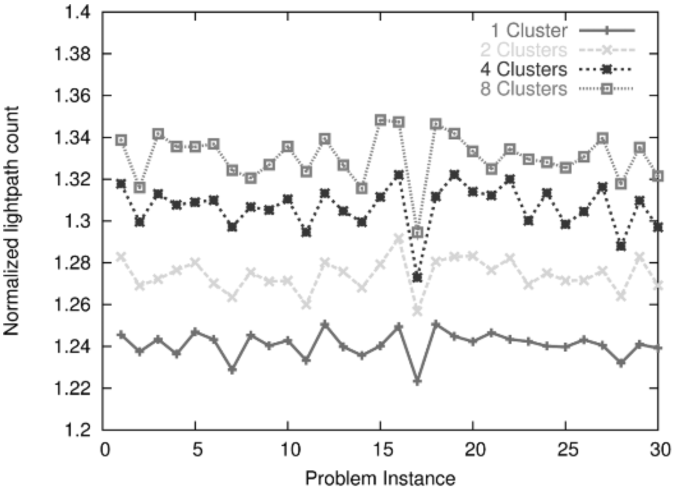


Fig. 5. Lightpath comparison, random pattern, 32-node network.

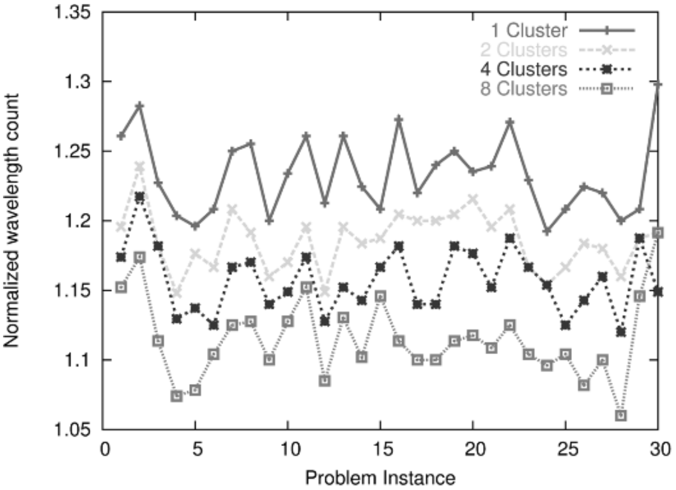


Fig. 6. Wavelength comparison, random pattern, 32-node network.

TABLE I
AGGREGATE STATISTICS OVER ALL 30 INSTANCES, RANDOM PATTERN

| #Clusters | Avg LP Length | Avg Max Hub Degree | Avg #$\lambda$s |
|---|---|---|---|
| 1 | 3.17 | 266 | 60 |
| 2 | 2.93 | 231 | 57 |
| 4 | 2.87 | 182 | 56 |
| 8 | 2.75 | 145 | 53 |

We observe that as the number of clusters into which the network is partitioned increases, the total number of lightpaths in the resulting topology increases gradually (Fig. 5). On the other hand, the number of required wavelengths generally decreases as the number of clusters increases (Fig. 6), and so do the average lightpath length and the maximum hub degree. These results can be explained by noting that, as the number of clusters increases, the size of each cluster decreases. With a smaller cluster size, more lightpaths are necessary for both intracluster traffic (since the amount of traffic within a cluster is relatively small and lightpaths are not utilized efficiently) and intercluster traffic (since each hub has to establish lightpaths to a larger number of hubs in other clusters). Also, intracluster lightpaths are shorter when clusters are small, and these short lightpaths are less likely to share links, resulting in fewer wavelengths. At the same time, there is relatively less traffic to be groomed at each hub; hence hub degrees (and hub cost) decrease; the fact that hubs are less of a bottleneck also reduces the wavelength requirements.

From Fig. 5, we note that the number of lightpaths created by our hierarchical grooming approach are only about 25–35% above the lower bound, and this behavior is consistent across all problem instances. As we mentioned earlier, however, we believe that this lower bound is rather loose since it does not take into consideration the underlying physical topology, hence the performance of our algorithm is better than the curves imply. From Fig. 6, we observe that, with appropriate clustering, the wavelength requirements of our approach are close to the lower bound obtained from the bisection. We also emphasize that both lower bounds have been computed in a

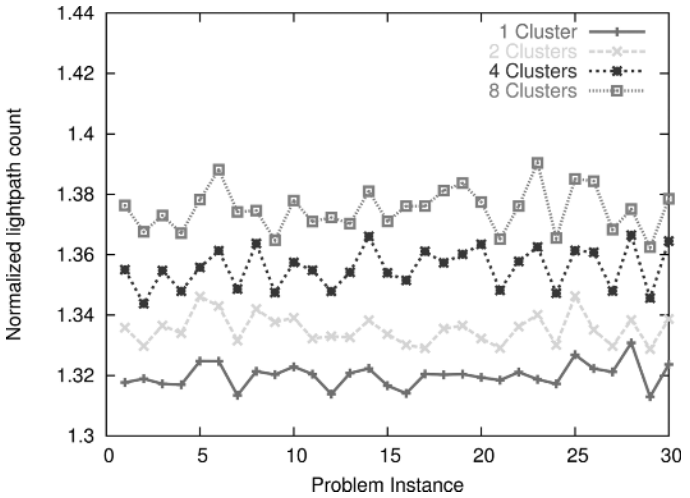


Fig. 7. Lightpath comparison, falling pattern, 32-node network.

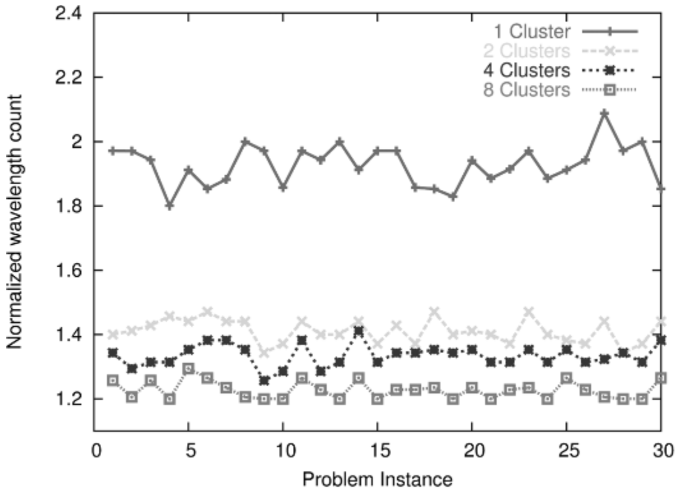


Fig. 8. Wavelength comparison, falling pattern, 32-node network.

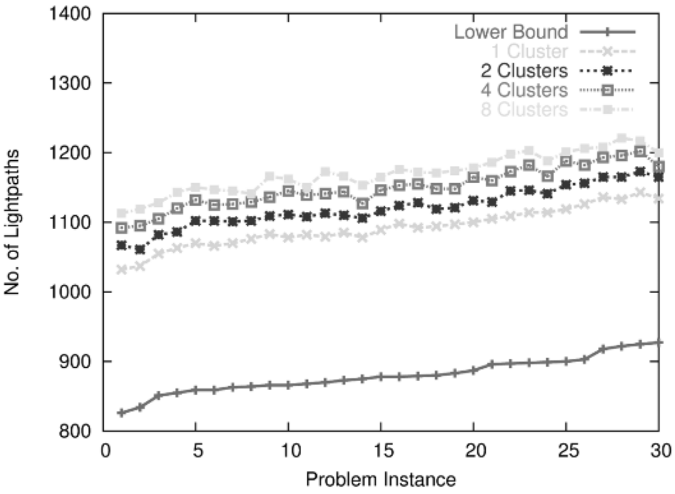


Fig. 9. Lightpath comparison, random pattern, sorted instances, 32-node network.

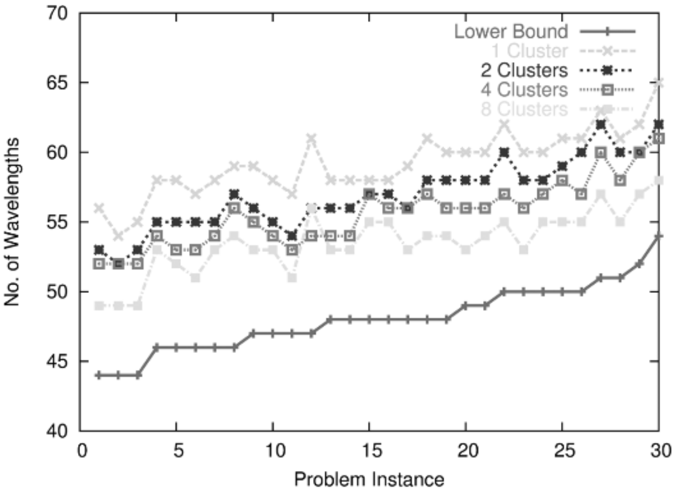


Fig. 10. Wavelength comparison, random pattern, sorted instances, 32-node network.

TABLE II
AGGREGATE STATISTICS OVER ALL 30 INSTANCES, FALLING PATTERN

| #Clusters | Avg LP Length | Avg Max Hub Degree | Avg #$\lambda$s |
|---|---|---|---|
| 1 | 2.49 | 484 | 67 |
| 2 | 2.24 | 312 | 49 |
| 4 | 2.23 | 209 | 46 |
| 8 | 2.15 | 166 | 42 |

manner that is independent of the grooming methodology employed. Consequently, these results demonstrate that despite its hierarchical nature, our approach produces grooming solutions that are close to optimality.

Figs. 7 and 8, and Table II are similar to the ones above, except that they present results for the falling pattern. As we can see, the general trends in these results are very similar to the ones we observed with the random traffic pattern. In particular, as the number of clusters increases, the total number of lightpaths also increases moderately, while the number of wavelengths, the average lightpath length and, the maximum hub degree all decrease. However, comparing the absolute values to the ones obtained with the random traffic pattern reveals the effect of the traffic pattern on the overall solution. For instance, the average lightpath length is significantly smaller under the falling pattern, due to the fact that most of the traffic is destined to nodes nearby; therefore, it is more likely to be confined within a cluster. There is a similar effect on the number of required wavelengths: a large cluster size is likely to force longer, indirect lightpaths which may cause wavelength collisions and require a larger number of wavelengths. Consequently, there is a significant drop in the wavelength requirements as we move from one to eight clusters (Fig. 8), which is more pronounced than the one in Fig. 6. Also, the clustering affects the maximum hub degrees much more dramatically than under the random pattern. In particular, when there are few clusters, even traffic destined locally is forced to travel to a relatively remote hub, increasing the degree of the hub (and the required electronic switching capacity) significantly. Increasing the number of clusters allows most of the traffic to remain within a cluster; as a result, the maximum hub degrees decrease by 66% when there are eight clusters compared to one cluster, while the corresponding decrease for the random pattern is about 45%.

While Figs. 5–8 indicate that hierarchical traffic grooming produces good solutions with respect to the number of lightpaths and wavelengths, it is instructive to also observe the behavior of our hierarchical grooming algorithm as a function of the lower bounds for each problem instance. To this end, in Figs. 9 and 10

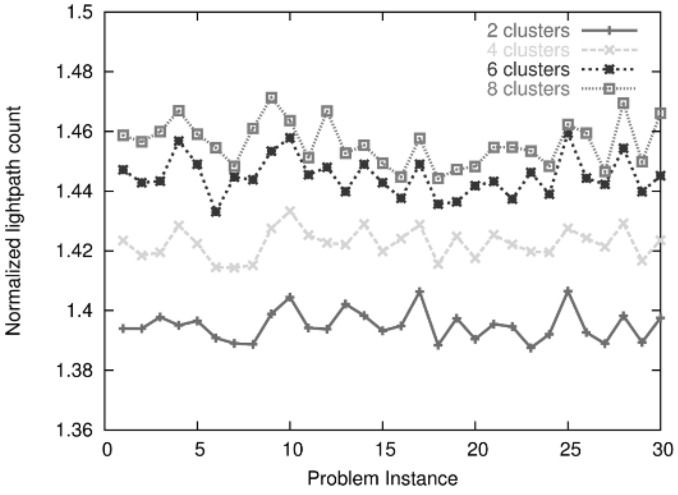


Fig. 11. Lightpath comparison, falling pattern, 47-node network.

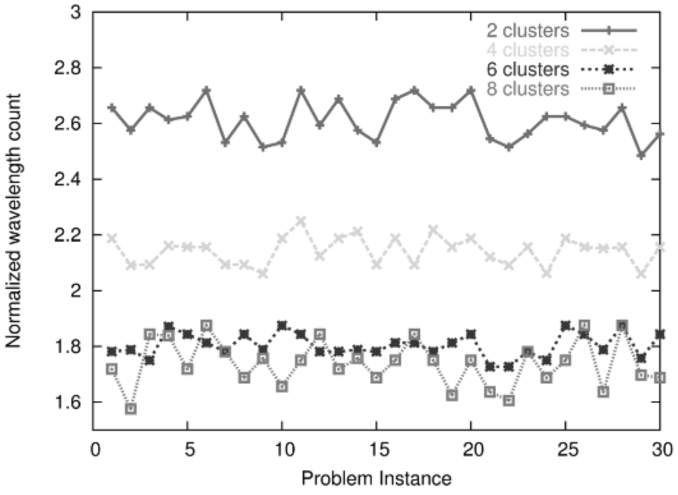


Fig. 12. Wavelength comparison, falling pattern, 47-node network.

we again plot the results from the same 30 problem instances generated according to the random traffic pattern. However, instead of plotting the normalized values (as in Figs. 5 and 6), we now plot the *absolute* values of the numbers of lightpaths and wavelengths, respectively, in the grooming solution, along with the corresponding lower bound values. Furthermore, we sort the problem instances in increasing order of the corresponding lower bounds. As we can observe, the curves of the numbers of lightpaths and wavelengths in our hierarchical solutions generally follow the curves of the lower bounds, increasing as the corresponding lower bound curves trend higher. These results indicate that the grooming solutions produced by our approach are likely to track closely the optimal solution as well.

### *B. The 47-Node, 96-Link Network*

We now consider a larger topology which appeared in a historical paper on network design [2]. The topology consists of 47 nodes and 96 links, the node degree of the network is relatively high, and the topology is balanced, in the sense that there is no bisection with a small cut size that can be a bottleneck for traffic grooming. Our experimental methodology is similar to the one we followed for the 32-node network in the previous section; the only difference is that due to the size of this topology we do not consider the case where all nodes belong to the same cluster, hence, we construct four clusterings with two, four, six, and eight nodes.

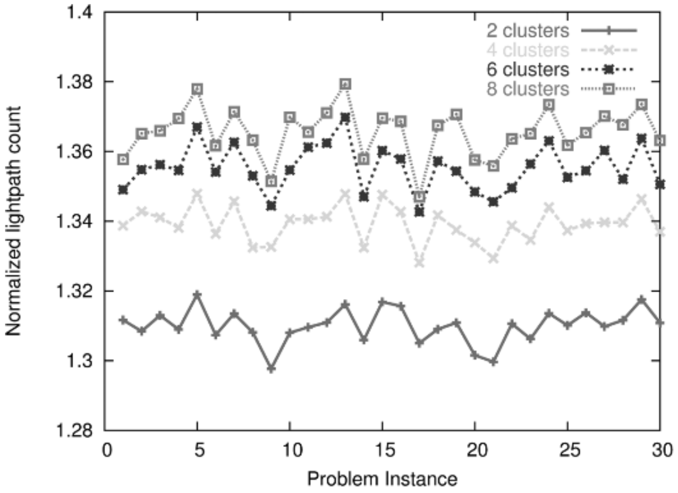


Fig. 13. Lightpath comparison, rising pattern, 47-node network.

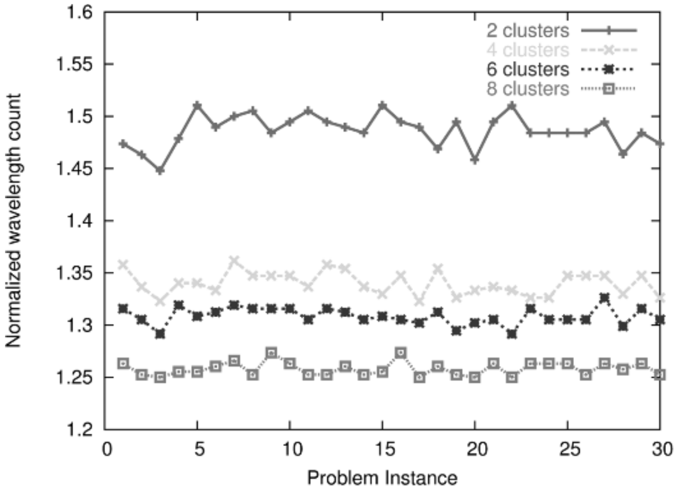


Fig. 14. Wavelength comparison, rising pattern, 47-node network.

Figs. 11 and 12 plot the normalized lightpath and wavelength count, respectively, for 30 problem instances generated according to the falling traffic pattern. Figs. 13 and 14 are similar, but report results for instances following the rising traffic pattern. In general, our earlier observations regarding the tradeoffs between the number of lightpaths and the number of wavelengths as the number of clusters increases remain valid. Furthermore, these results demonstrate the increasing benefit of partitioning large networks into more clusters. While increasing the number of clusters from two to eight only slightly increases the normalized lightpath count, it may significantly reduce the wavelength requirements. This result can be explained by noting that when a large network is partitioned into a small number of clusters, each cluster will be relatively large. Consequently, lightpaths within the cluster will tend to be long and their paths will overlap with each other, increasing the number of wavelengths. Since increasing the number of clusters has a more pronounced effect on reducing the cluster size in large networks, the wavelength requirements are affected accordingly.

Overall, the results we presented demonstrate that our traffic grooming approach can be efficiently applied to large-size networks and produce hierarchical logical topologies whose lightpath and wavelength requirements are close to the corresponding lower bounds. For a more comprehensive set of experiments that confirm this conclusion, the reader is referred to [4].

## VI. CONCLUDING REMARKS

We have presented a new hierarchical framework for efficient and scalable traffic grooming in large-scale WDM networks with a general topology. Our framework consists of three phases: clustering and hub selection, hierarchical topology design, and routing and wavelength assignment, and we have identified important tradeoffs between the number of clusters (or, equivalently, the cluster size) and pertinent performance metrics such as the total number of lightpaths and wavelengths, the average lightpath length, and the hub degrees.

We use a modified version of the *K-Center* algorithm to decompose the network into a set of first-level clusters, and to select a node in each cluster as the hub. We view each cluster as a virtual star, and present an efficient algorithm for grooming the intracluster traffic at the hub. We then groom the intercluster traffic by applying the same algorithm to a second-level virtual star consisting of the hubs of first-level clusters. An important benefit of the virtual-star approach is that the grooming of subwavelength components into lightpaths is implicit in the logical topology obtained. Once the lightpaths for carrying all the traffic demands have been determined, the routing and wavelength assignment is performed directly on the underlying physical topology so as to minimize the number of wavelengths required to establish the lightpaths. We have presented numerical results which demonstrate that hierarchical grooming performs well over a range of network topologies and traffic patterns and scales to networks of realistic size.

Although our techniques were developed for static grooming, we expect that our hierarchical framework is applicable to dynamic grooming as well; this is the subject of ongoing research.

## REFERENCES

[1] J. Bar-Ilan, G. Kortsarz, and D. Peleg, "How to allocate network centers," *J. Alg.*, vol. 15, pp. 385–415, 1993.

[2] P. Baran, "On distributed communications networks," *IEEE Trans. Commun.*, vol. COM-12, no. 1, pp. 1–9, Mar. 1964.

[3] S. Baroni and P. Bayvel, "Wavelength requirements in arbitrary connected wavelength-routed optical networks," *J. Lightw. Technol.*, vol. 15, no. 2, pp. 242–251, Feb. 1997.

[4] B. Chen, "Hierarchical traffic grooming in large-scale WDM networks" Ph.D. dissertation, North Carolina State Univ., Raleigh, NC, 2005.

[5] I. Chlamtac, A. Ganz, and G. Karmi, "Lightpath communications: An approach to high bandwidth optical WAN's," *IEEE Trans. Commun.*, vol. 40, no. 7, pp. 1171–1182, Jul. 1992.

[6] H. Choi and D. B. Szyld, "Application of threshold partitioning of sparse matrices to Markov chains ," in *Proc. IEEE Int. Comput. Performance and Dependability Symp. (IPDS)*, 1996.

[7] Z. Ding and M. Hamdi, "Clustering techniques for traffic grooming in optical WDM mesh networks," in *IEEE GLOBECOM 2002*, Nov. 2002, vol. 3,17–21, pp. 2711–2715.

[8] R. Dutta, S. Huang, and G. N. Rouskas, "Traffic grooming in path, star, and tree networks: Complexity, bounds, and algorithms," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 4, pp. 66–82, Apr. 2006.

[9] R. Dutta and G. N. Rouskas, "On optimal traffic grooming in WDM rings," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 1, pp. 110–121, Jan. 2002.

[10] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Netw.*, vol. 16, no. 6, pp. 46–56, Nov./Dec. 2002.

[11] M. Esfandiari, S. Gloeckle, A. Zolfagheri, G. Clapp, J. Gannett, H. Kobrinski, V. Poudyal, and R. Skoog, "Improved metro network design by grooming and aggregation of STS-1 demands into OC-192/OC-48 lightpaths," in *Proc. NFOEC 2003*, Sep. 2003.

[12] O. Gerstel, R. Ramaswami, and G. Sasaki, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Trans. Netw.*, vol. 8, no. 5, pp. 618–630, Oct. 2000.

[13] T. Gonzalez, "Clustering to minimize the maximum inter-cluster distance," *Theoret. Comput. Sci.*, vol. 38, pp. 293–306, 1985.

[14] A. H. John*, Clustering Algorithms*. Hoboken, NJ: Wiley, 1975.

[15] D. Hochbaum and D. B. Shmoys, "A best possible heuristic for the k-center problem," *Math. Oper. Res.*, vol. 10, pp. 180–184, 1985.

[16] D. Hochbaum and D. B. Shmoys, "A unified apporach to approximation algorithms for bottleneck problems," *J. ACM*, vol. 33, pp. 533–550, 1986.

[17] J. Q. Hu and B. Leida, "Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks," in *Proc. IEEE INFOCOM*, 2004, vol. 23, pp. 495–501.

[18] Z. K. G. Patrocínio, Jr. and G. R. Mateus, "A Lagrangian-based heuristic for traffic grooming in WDM optical networks," in *IEEE GLOBECOM*, 2003, pp. 2767–2771.

[19] G. Karypis and V. Kumar, "A fast and highly quality multilevel scheme for partitioning irregular graphs," *SIAM J. Sci. Comput.*, 1998.

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

[21] C. Lee and E. K. Park, "A genetic algorithm for traffic grooming in all-optical mesh networks," in *IEEE Int. Conf. Systems, Man, and Cybernetics*, Oct. 2002, vol. 7, pp. 6–9.

[22] D. Li, Z. Sun, X. Jia, and K. Makki, "Traffic grooming on general topology WDM networks," *Proc. Inst. Electr. Eng.—Commun.*, vol. 150, no. 3, pp. 197–201, Jun. 2003.

[23] K. Schloegel, G. Karypis, and V. Kumar, "A new algorithm for multi-objective graph partitioning," Dept. Comput. Sci., Univ. of Minnesota, Duluth, MN, Tech. Rep. 99-003, 1999.

[24] D. B. Shmoys, "Approximation algorithms for facility location problems," in *Proc. APPROX 2000*, Lecture Notes in Computer Science vol. 1913, pp. 27–32.

[25] J. Simmons, E. Goldstein, and A. Saleh, "Quantifying the benefit of wavelength add-drop in WDM rings with distance-independent and dependent traffic," *J. Lightw. Technol.*, vol. 17, no. 1, pp. 48–57, Jan. 1999.

[26] H. Siregar, H. Takagi, and Y. Zhang, "Efficient routing and wavelength assignment in wavelength-routed optical networks," in *Proc. 7th Asia-Pacific Network Operations and Management Symp.*, Oct. 2003, pp. 116–127.

[27] M. Thorup, "Quick k-median, k-center, and facility location for sparse graphs," in *Proc. ICALP 2001*, Lecture Notes in Computer Science vol. 2076, pp. 249–260.

[28] P.-J. Wan, G. Calinescu, L. Liu, and O. Frieder, "Grooming of arbitrary traffic in SONET/WDM BLSRS," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 10, pp. 1995–2003, Oct. 2000.

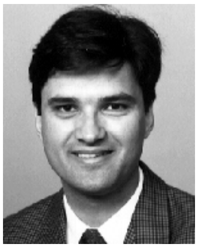
[29] D. West*, Introduction to Graph Theory*. Upper Saddle River, NJ: Prentice-Hall, 1996.

[30] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 285–299, Apr. 2003.

[31] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 1, pp. 122–133, Jan. 2002.

**Bensong Chen** received the B.E. degree in management information systems from Beijing Jiaotong University, Beijing, China, in 1998, the M.E. degree in computer engineering from the Chinese Academy of Sciences, Beijing, in 2001, and the Ph.D. degree from North Carolina State University, Raleigh, in 2005.

In August 2001, he began his research study at North Carolina State University. His research interests have focused on traffic grooming in WDM optical networks. Since September 2005, he has been with Google, Inc., Mountain View, CA, as a Research Engineer and is currently working on web crawl/indexing and Internet search quality.



**George N. Rouskas** (S'92–M'95–SM'01) received the Diploma in computer engineering from the National Technical University of Athens (NTUA), Athens, Greece, in 1989, and the M.S. and Ph.D. degrees in computer science from the Georgia Institute of Technology (Georgia Tech), Atlanta, in 1991 and 1994, respectively.

He is currently a Professor of computer science at North Carolina State University (NCSU), Raleigh. During the 2000–2001 academic year, he spent a sabbatical term at Vitesse Semiconductor, Morrisville, NC, and in May 2000, December 2002, and July 2006 he was an Invited Professor at the University of Evry, France. His research interests include network architectures and protocols, optical networks, multicast communication, and performance evaluation.

Dr. Rouskas received the 2004 ALCOA Foundation Engineering Research Achievement Award, and the 2003 NCSU Alumni Outstanding Research Award. He is a recipient of a 1997 NSF Faculty Early Career Development (CAREER) Award, the recipient of the 1994 Graduate Research Assistant Award from the College of Computing, Georgia Tech, and a coauthor of papers which received the Best Paper Award at the 2006 CSNDSP Conference and the 1998 SPIE Conference on All-Optical Networking. He has received several teaching awards, including induction in the NCSU Academy of Outstanding Teachers in 2004 and the Outstanding New Teacher Award from the NCSU Department of Computer Science in 1995. He is the Co-Editor-in-Chief of *Optical Switching and Networking*, has been on the editorial boards of the IEEE/ACM Transactions on Networking, *Computer Networks*, and *Optical Networks*, and he was a Co-Guest Editor for the IEEE Journal on Selected Areas in Communications special issue on Protocols and Architectures for Next Generation Optical WDM Networks (Oct. 2000). He was Technical Program Co-Chair of the Networking 2004 Conference, Program Chair of the IEEE LANMAN 2004 Workshop, Program Co-Chair of the Traffic Grooming 2004 Workshop, General Co-Chair of the IEEE LANMAN 2005 Workshop, and General Co-Chair of the BROADNETS 2007 Conference. He is a member of the ACM and of the Technical Chamber of Greece.

**Rudra Dutta** (S'00–M'01) was born in Kolkata, India, in 1968. He received the B.E. degree in electricial engineering from Jadavpur University, Calcutta, India, in 1991, the M.E. degree in systems science and automation from the Indian Institute of Science, Bangalore, in 1993, and the Ph.D. degree in computer science from North Carolina State University (NCSU), Raleigh, in 2001.

From 1993 to 1997 he worked for IBM as a Software Developer and Programmer in various networking-related projects. Since 2001, he has been employed in the department of Computer Science at the NCSU, where he is an Associate Professor. During the summer of 2005, he was a visiting researcher at the IBM WebSphere Technology Institute. His current research interests focus on design and performance optimization of large networking systems. His research is supported currently by grants from the National Science Foundation (NSF) and industry, including the one which supported in part this paper and a recent FIND grant from NSF.

Dr. Dutta has served as a Reviewer for many premium journals, on NSF review panels, was part of the organizing committee of BroadNets 2005 and BroadNets 2006, and Program Co-chair for the Second International Workshop on Traffic Grooming. He is serving as Industry Sponsorship Chair for BroadNets 2007, Program Vice-Chair for Optical Networking Symposium, ICC 2008, and Program Chair for the Optical Networking Symposium at Globecom 2008.