A Framework for Hierarchical Traffic Grooming in WDM Networks of General Topology

Bensong Chen, George N. Rouskas, Rudra Dutta
Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534

***Abstract*— We present a framework for hierarchical traffic grooming in mesh networks 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 inter-cluster 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. Our approach scales to large network sizes, and facilitates the control and management of multigranular networks. Comparisons to lower bounds indicate that it is also efficient in its use of the network resources of interest, namely, electronic ports and wavelengths.**

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 sub-wavelength 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 [6].

Traffic grooming research has, in general, followed one of two directions. In *dynamic* grooming [18], 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 on-line 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 [5], [8], [16], 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 [9], [10], [12]–[14], [19]. 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.

In essence, solution approaches based on an ILP formulation 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 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 framework for hierarchical traffic grooming in mesh networks with the objective of minimizing the total number of electronic ports in the network[1]. To this end, we emulate 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 intra-cluster traffic as well as inter-cluster traffic originating or terminating locally. At the second level of the hierarchy, the first-level hubs form another cluster for grooming and routing inter-cluster 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;

[1]Note that a lightpath requires exactly two electronic ports, one at the source and one at the destination. Hence, minimizing the number of electronic ports is equivalent to minimizing the number of lightpaths in the logical topology.

This work was supported by NSF grant ANI-0322107.

- 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; and
- it routes large 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 made in [7], while the use of the blocking island paradigm for tackling a restricted version of traffic grooming in mesh networks was advocated in [4]; this paradigm allows for the abstraction of network resources, and can be applied recursively on the network graph. Our approach is more comprehensive than either [7] or [4], and is quite general, in the sense that it can be extended to a wide range of variants of the grooming problem.

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 obtain lower bounds for the number of lightpaths and the number of wavelengths in Section V, and we present numerical results in Section VI. We conclude the paper in Section VII.

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. A formulation of this traffic grooming problem as an integer linear problem (ILP) is omitted, but is available in [6]. The problem involves the following conceptual subproblems (SPs): (1) *logical topology SP:* find a set R of lightpaths that forms a virtual topology, (2) *lightpath routing and wavelength assignment (RWA) SP:* solve the RWA problem on 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. In the problem formulation, the number W of wavelengths is taken into consideration as a constraint rather than as a parameter to be minimized.

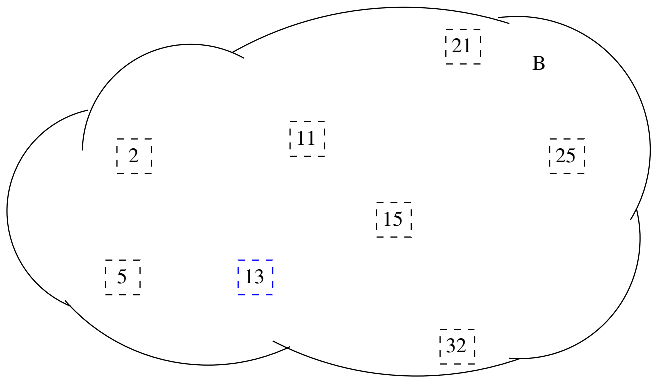
The above traffic grooming problem defined on a general topology is NP-hard, since the RWA subproblem is NP-hard [3]. Next, we outline our hierarchical approach to traffic grooming in general topologies.

A. A Hierarchical Approach to Traffic Grooming

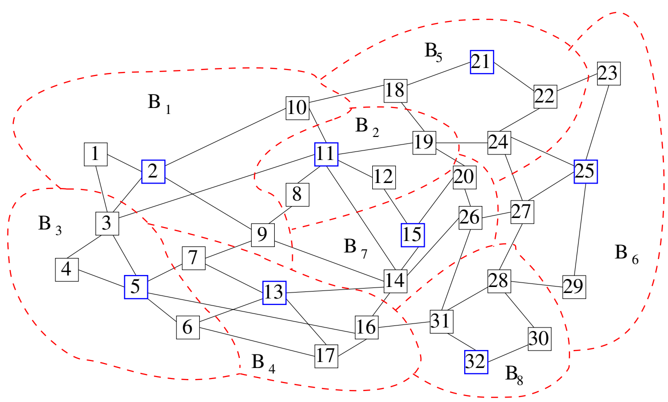
Our approach borrows ideas from the hub-and-spoke paradigm that is widely used within the airline industry. Specifically, we assume that the network is partitioned into clusters (or islands) of nodes, where each cluster consists of nodes in a contiguous region of the network. The clusters may correspond 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 non-hub 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 non-hub 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



(b) Second−level cluster consisting of first−level hubs, and hub node 13



(a) First−level clusters

Fig. 1. A 32-node WDM network, its partition into eight first-level clusters $B_1, \cdots, B_8$, and second-level cluster B consisting of the eight first-level hubs

next section to each cluster; the result of this step is a set of lightpaths within each cluster to route local (intra-cluster) traffic, as well as inter-cluster 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 inter-cluster 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 Figure 1. The bottom part of the figure shows a partition of the network into eight clusters, $B_1, \cdots, B_8$, 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 32 as its hub. Suppose that the logical topology obtained by running the StarTopology algorithm on the corresponding *virtual star* with node 32 as the hub, includes the "one-hop" lightpath $(28, 32)$ and the "two-hop" lightpath $(31, 28)$. After running the RWA algorithm, the "one-hop" lightpath may be routed over the path $28-30-32$ (since node 28 is not directly connected to the hub node 32 of the virtual star), while the "two-hop" lightpath may in fact be routed over the direct link $31-28$, completely bypassing the hub node 32 (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.

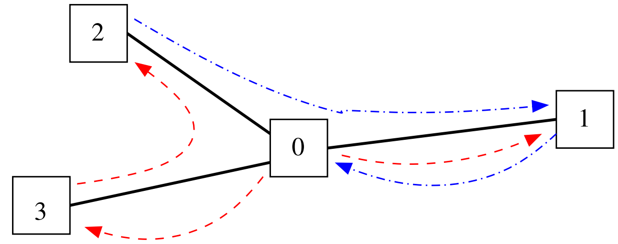


Fig. 2. A 4-node star with five lightpaths

III. TRAFFIC GROOMING IN STAR NETWORKS

Let us consider a network with a star physical topology, as in Figure 2. The network consists of N nodes, a central *hub* node labeled 0, and $N-1$ nodes labeled $1, \cdots, N-1$, each connected to the hub over a bidirectional fiber link (denoted by the thick solid lines in Figure 2) that can carry W wavelengths in each direction. As before, we assume that C is the capacity of each wavelength, and that the traffic demands are provided in the form of the traffic matrix $T = [t^{(sd)}]$. In order to ensure efficient use of wavelength capacity, we further assume that no traffic component is allowed to traverse the same physical link back-and-forth. In other words, non-hub nodes are not allowed to switch traffic, either optically or electronically, and all traffic grooming is performed at the hub node. Under these assumptions, there can be only two types of lightpaths in the logical topology, denoted by the dotted lines in Figure 2. The first type consists of single-hop lightpaths which either originate at a non-hub node and terminate at the hub node, or vice versa. The second type consists of two-hop lightpaths that originate and terminate at non-hub nodes, and are switched optically at the hub node.

Before we proceed, we note that for a star physical topology, given a solution to the logical topology subproblem of the traffic grooming problem we defined in the previous section, the other two subproblems can be easily solved in polynomial

Logical Topology Algorithm for Star Networks
Input: A star network with N nodes, W wavelengths, 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; **or** failure if no feasible solution exists.

procedure **StarTopology**
begin
1. Reduce the traffic matrix T, and record the residual traffic matrix $T_r = [t_r^{(sd)}], t_r^{(sd)} < C \ \forall \ s, d$
2. Create single-hop lightpaths to carry the residual traffic by electronically switching (grooming) it at the hub
3. Check feasibility; if infeasible, exit with failure
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**
12. Find the smallest of $U_0, U_1, \ldots, U_m$, and return the corresponding logical topology R as the solution

end

Fig. 3. Logical topology algorithm for star networks

time. Consider the RWA subproblem. Each of the lightpaths in the logical topology is routed over the corresponding unique path in the star topology, while wavelength assignment can be performed in polynomial time [17]. Also, the routing of traffic components (i.e., a solution to the third subproblem) is implicit in the logical topology: components are routed over two-hop lightpaths, if they exist, otherwise they are packed onto single-hop lightpaths to the hub and then to their destination. Therefore, in the remainder of this section we concentrate on the first subproblem of traffic grooming.

Despite the special nature of the star topology, the problem of finding a logical topology that minimizes the total number of electronic ports (equivalently, the total number of lightpaths) remains NP-hard [2]. We now present a greedy heuristic for constructing a near-optimal logical topology in star networks given the traffic matrix T. The main idea behind the algorithm is to assign direct two-hop lightpaths to as many of the largest traffic demands as possible, while carrying smaller demands on single-hop lightpaths to the hub for grooming. A pseudocode description of the algorithm is provided in Figure 3, and its steps are explained in detail below. As we discuss later, this algorithm is used for obtaining the logical topology within each cluster of a large network of general topology.

As a first step, we *reduce* the original traffic matrix T by assigning direct lightpaths to all traffic demands $t^{(sd)}$ that can fill a wavelength (i.e., such that $t^{(sd)} \geq C$). Doing so does not affect the optimality of the solution, since breaking such lightpaths does not benefit either the wavelength constraints or the objective of minimizing the number of lightpaths.

After reduction, the residual traffic demands to be groomed are less than the wavelength capacity C, for each source-destination pair. 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. If, after this step, the most congested link (the physical fiber that carries the largest number of lightpaths) has more than W lightpaths, we can conclude that no feasible solution exists.

The above all-electronic solution for the reduced traffic matrix 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 re-route traffic demands between non-hub 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 non-hub demands for which to set up direct lightpaths is NP-hard (since the star grooming problem is NP-hard [2]), 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 $O(N^2)$.

Figure 4 shows the evolution of the objective (number of lightpaths in the logical topology) after each iteration of the grooming algorithm, for a problem instance with $N = 25$ nodes. Iteration 0 corresponds to a logical topology in which all traffic (after reduction) is carried on single-hop lightpaths (Step 2 of the algorithm), resulting in a relatively large number of lightpaths. As large traffic components get assigned to direct two-hop lightpaths in subsequent iterations, the number of lightpaths starts to decrease, implying that the creation of a new two-hop lightpath leads to the removal of two single-hop lightpaths and a decrease in the objective. In certain cases, the creation of a two-hop lightpath results in the removal of only one single-hop lightpath, and the objective is not affected. In other iterations, the creation of a two-hop lightpath does not

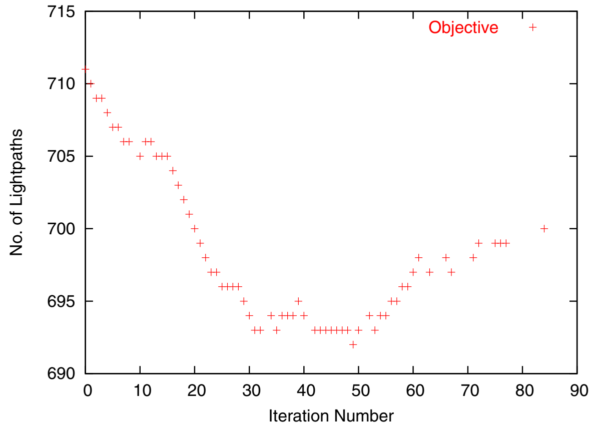


Fig. 4. Evolution of the objective after each iteration of the StarTopology algorithm

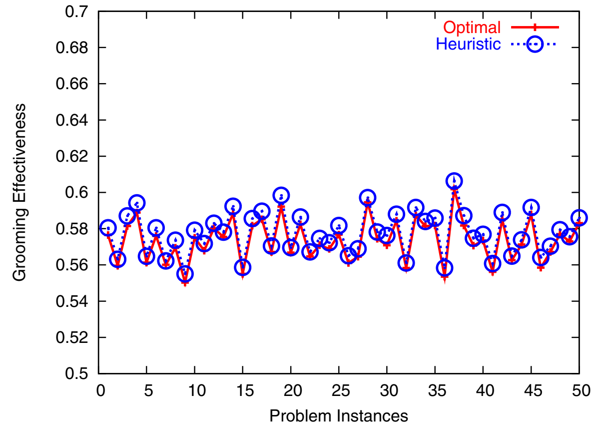


Fig. 5. Grooming effectiveness of the StarTopology algorithm, $N = 10$

lead to the removal of any single-hop lightpath, resulting in an increase in the total number of lightpaths; this situation happens more frequently in later iterations when smaller traffic components are routed over direct lightpaths. Because of these oscillations in the value of the objective, the algorithm in Figure 3 returns the best logical topology found after all iterations have completed. Finally, there are some iterations during which no two-hop lightpath is created since doing so would violate the wavelength constraint on some link (refer to Step 8 of the algorithm). In this case, the value of the objective does not change during the iteration. To distinguish this case from the one in which one two-hop lightpath is created and one single-hop lightpath is removed, in the graph of Figure 4 we do not plot any value for the objective when no lightpath is created during an iteration.

We have conducted a large number of experiments to evaluate the performance of the star grooming algorithm under various traffic patterns and network sizes. Due to space constraints, we show one representative experiment in Figure 5 for 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 within a reasonable amount of time (a few hours per problem instance). Figure 5 plots the *grooming effectiveness* of the optimal solution and the solution obtained by our algorithm, for fifty problem instances with $N = 10$ and random traffic patterns. The grooming effectiveness 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 of grooming effectiveness implies a better solution. As we can see, the solution obtained by our algorithm tracks the optimal solution closely over all fifty problem instances. 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. We have obtained similar results for a wide range of problem instances [2].

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 m 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) intra-cluster traffic grooming; and
   c) inter-cluster 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 are 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 m of clusters, denoted $B_1, \cdots, B_m$. Second, we select one node in each cluster to serve as the hub where grooming of intra- and inter-cluster traffic is performed. Let n_i denote the number of nodes in cluster B_i, $n_1 + n_2 + \cdots + n_m = N$, 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 m of clusters. If m is very small, the amount of inter-cluster traffic will likely be large. Hence, the m 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 m implies a small number of nodes within each cluster. In this case, the amount of intra-cluster traffic will be small, resulting in inefficient grooming (i.e., a large number of lightpaths); similarly, at the second-level cluster, $O(m^2)$ lightpaths will have to be set up to carry small amounts of inter-cluster traffic.

The development of good clustering algorithms that lead to a logical topology with a small number of lightpaths, and which will not require a large number of wavelengths when superimposed on the underlying physical topology, is the subject of ongoing research within our group. In this paper, we manually partition the network in Figure 1, and we experiment with clusters of various sizes in Section VI.

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 $p \in (0.5, 1)$ is a parameter determined 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 of 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_{init} be the set of direct lightpaths created in this step. Let $T_r = [t_r^{(sd)}]$ denote the matrix of residual traffic demands (i.e., excluding those carried by the lightpaths in R_{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 $\{t_r^{(sd)}\}$.

2) Intra-cluster 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 $T_i = [t_i^{(sd)}]$, defined as:

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

In other words, if s and d are non-hub nodes, then $t_i^{(sd)}$ represents the intra-cluster 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 intra-cluster traffic component $t_r^{(sd)}$, but also the aggregate inter-cluster 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 inter-cluster traffic, other than that carried by direct lightpaths set up earlier, is first carried to the local hub, groomed there with inter-cluster traffic from other local nodes, carried on lightpaths to the destination hub (as we discuss shortly), groomed there with other local and non-local traffic, and finally carried to the destination node.

Given traffic matrix $T_i = [t_i^{(sd)}]$, we view cluster B_i as a *virtual star* with hub h_i and $n_i - 1$ non-hub nodes. We apply the StarTopology algorithm in Figure 3 to obtain the set of lightpaths R_i for carrying the demands $\{t_i^{(sd)}\}$. Recall that the lightpaths in R_i are either "single-hop" (i.e., from a non-hub node to the hub, or vice versa), or "two-hop" (i.e., from one non-hub 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 non-hub 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 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 non-hub nodes are connected by a direct link).

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

Logical Topology Algorithm for Mesh Networks

Input: A mesh WDM network with N nodes partitioned in m clusters $B_1, \cdots, B_m$, hub h_i of cluster B_i, W wavelengths per link, capacity C of each wavelength, and traffic matrix $T = [t^{(sd)}]$

Output: The set of lightpaths R in the logical topology, and 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 $> C$
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** $i = 1, \cdots, m$ **do**
6. $\quad T_i = [t_i^{(sd)}] \leftarrow$ intra-cluster traffic matrix for cluster B_i, computed from expression (1)
7. $\quad R_i \leftarrow$ set of lightpaths obtained by running the StarTopology algorithm on virtual star B_i with hub h_i
8. **end for**
9. $R_{intra} \leftarrow R_1 \cup R_2 \cup \cdots \cup R_m$
   // Inter-cluster grooming
10. $B \leftarrow$ cluster consisting of m hub nodes $h_1, \cdots, h_m$
11. $h \leftarrow$ hub of cluster B
12. $T_{inter} \leftarrow$ the $m \times m$ 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. 6. Logical topology algorithm for mesh networks

3) Inter-cluster traffic grooming: At the end of intra-cluster 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 m hub nodes $h_1, \cdots, h_m$, of the first-level clusters. Let $h \in \{h_1, \cdots, h_m\}$ be the node designated as the second-level hub. We view cluster B as a *virtual star* with a $m \times m$ traffic matrix $T_{inter} = [t_{inter}^{(h_i h_j)}]$ representing the inter-cluster traffic demands. This inter-cluster matrix is defined as:

$$t_{inter}^{(h_i h_j)} = \sum_{s \in B_i, d \in B_j} t_r^{(s,d)}, \quad i, j = 1, \cdots, m, \; i \neq j \qquad (2)$$

We now apply the StarTopology algorithm in Figure 3 to the *virtual star* B with hub h, and we obtain the set of lightpaths R_{inter} to carry the traffic demands $\{t_{inter}^{(h_i h_j)}\}$. 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 intra-cluster lightpaths above also apply to the lightpaths in R_{inter}.

Figure 6 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 inter-cluster grooming in Steps 5-8 and 13, respectively. The **for** loop in Steps 5-8 is executed m times, where m 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 $O(n_i^2)$. Since $n_i > 1$ and $n_1 + \cdots + n_m = N$, we have that $N \leq n_1^2 + \cdots + n_m^2 \leq N^2$, hence the **for** loop takes time $O(N^2)$. Step 13 calls the StarTopology algorithm on the second-level cluster with m nodes, taking time $O(m^2)$. Since $m < N$, 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 $R = R_{init} + R_{intra} + R_{inter}$, 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 them using the minimum number of wavelengths. The RWA problem on arbitrary network topologies has been studied extensively in the literature [1], [3], [9], [10], [15]. In this work, we adopt the LFAP algorithm [15] 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 non-increasing 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 pre-computed 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; otherwise, set $w \leftarrow w + 1$ and repeat from Step 2.

V. Lower Bounds

We now obtain lower bounds on both the number of lightpaths and the number of wavelengths required to carry the traffic matrix T. These bounds are obtained *independently* of the manner (e.g., hierarchical or otherwise) in which traffic grooming is performed. Therefore, the bounds are useful in characterizing the effectiveness of our algorithm for problem instances for which it is not possible to solve the ILP directly to find an optimal solution.

A. A Simple Lower Bound on the Number of Lightpaths

A simple lower bound F^l on the total number of lightpaths (our main objective) is given by:

$$F^l = \max\left(\sum_s \left\lceil \frac{\sum_d t^{(sd)}}{C} \right\rceil, \sum_d \left\lceil \frac{\sum_s t^{(sd)}}{C} \right\rceil\right) \quad (3)$$

This bound is 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.

B. A Lower Bound on the Number of Wavelengths

Consider a 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 k be the number of links in the cut set, and C the capacity of each wavelength. Then, the quantity $\lceil t/kC \rceil$ is a lower bound on the number of wavelengths for carrying the given traffic matrix. This bound does not require any information regarding the logical topology or the routing and wavelength assignment of lightpaths.

We used the METIS software [11] to obtain a bisection of the 32-node network in Figure 1 into two groups of roughly equal size with the minimum number of links in the cut set. We use this bisection (whose cut set consists of the five links (10,18), (11,19), (15,20), (14,26), and (16,31), and partitions the network into two groups of 15 and 17 nodes each) to obtain a lower bound on the number of wavelengths for all problem instances we consider.

We also note that, once a logical topology has been determined, a lower bound on the number of wavelengths *for this particular topology* can be obtained by determining the maximum number of lightpaths that travel across the cut in either direction, and dividing it by the cut size. We expect that this bound will be higher than the one that is independent of the logical topology, and the amount of increase is an indication of the performance of the logical topology design algorithm.

VI. Numerical Results

In this section, we present experimental results to demonstrate the performance of our hierarchical grooming algorithm. In all our experiments we use the network with $N = 32$ nodes shown in Figure 1. In order to evaluate the hierarchical grooming approach under various cluster sizes and study the tradeoffs involved, we partition the network into one, two, four, or eight clusters. Table I shows the composition and hub node (shown in bold) of each cluster; the eight-cluster partition $B_1, \cdots, B_8$ is identical to the one shown in Figure 1.

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 σ that depend on the traffic pattern. We consider two traffic patterns here:

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) *Locality pattern.* This traffic pattern is designed to capture the traffic locality property that has been observed in some networks. Specifically, if the mean of the Gaussian 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 of the Gaussian distribution be 20% of the mean.

Figures 7 and 8 and Table II present experimental results obtained by applying the MeshTopology algorithm to each of the four clusterings of the 32-node network, as shown in Table I. For each clustering, we generated thirty instances of the problem, and the traffic matrix of each instance was created according to the random traffic pattern. Figure 7 plots, for each problem instance, the number of lightpaths created by the MeshTopology algorithm for each clustering, as well as the lower bound on the number of lightpaths from expression (3). Figure 8 plots, for each problem instance, the number of wavelengths required to establish the logical topology for each clustering, as well as two lower bounds on the number of wavelengths. The bottom curve in the figure is the lower bound based on the bisection of the 32-node network (refer to Section V-B); this lower bound is independent of how the traffic grooming problem is solved. The next higher curve is the best of the four topology-specific lower bounds. Each bound corresponds to the logical topology of a clustering as explained in Section V-B. Finally, Table II presents aggregate statistics over all thirty problem instances regarding the average lightpath length, the average maximum hub degree, and the average number of wavelengths.

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 (Figure 7). On the other hand, the number of required wavelengths generally decreases as the number of clusters increases (Figure 8), 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 intra-cluster traffic (since the amount of traffic within a cluster is relatively small and lightpaths are not utilized efficiently) and inter-cluster traffic (since each hub has to establish lightpaths to a larger number of hubs in other clusters). Also, intra-cluster lightpaths are shorter when clusters are small, and these short lightpaths are less likely to

#Clusters	Cluster composition and hub nodes
1	$\{B_1 \cup B_2 \cup B_3 \cup B_4 \cup B_5 \cup B_6 \cup B_7 \cup B_8$ **(13)**$\}$
2	$\{B_1 \cup B_2 \cup B_3 \cup B_4$ **(13)**$\}$, $\{B_5 \cup B_6 \cup B_7 \cup B_8$ **(15)**$\}$
4	$\{B_1 \cup B_2$ **(11)**$\}$, $\{B_3 \cup B_4$ **(5)**$\}$, $\{B_5 \cup B_6$ **(21)**$\}$, $\{B_7 \cup B_8$ **(32)**$\}$
8	$\{B_1$, **2**$\}$, $\{B_2$, **11**$\}$, $\{B_3$, **5**$\}$, $\{B_4$, **13**$\}$, $\{B_5$, **21**$\}$, $\{B_6$, **25**$\}$, $\{B_7$, **15**$\}$, $\{B_8$, **32**$\}$

TABLE I

CLUSTER AND HUB SELECTION FOR THE 32-NODE NETWORK IN FIGURE 1

#Clusters	Avg LP Length	Avg Max Hub Degree	Wavelengths
1	3.17	266	60
2	3.07	228	60
4	2.93	183	59
8	2.84	143	56

TABLE II

AGGREGATE STATISTICS FOR THE RANDOM TRAFFIC PATTERN

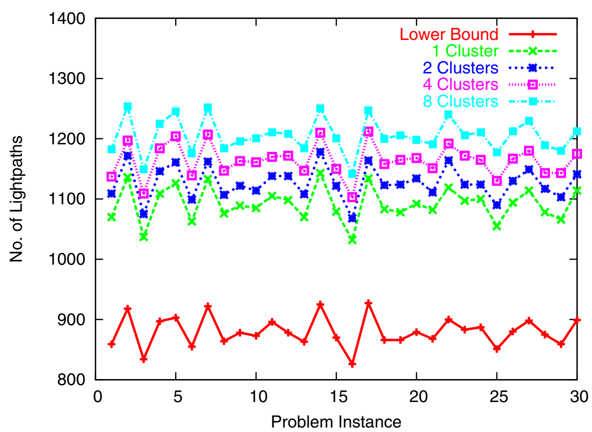


Fig. 7. Number of lightpaths for various numbers of clusters, 32-node network of Figure 1 with random traffic pattern

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.

Form Figure 7, we note that the number of lightpaths created by our hierarchical grooming approach are only about 25-30% above the lower bound, and this behavior is consistent across all problem instances. We believe, however, that this lower bound is rather loose since expression (3) does not take into consideration the underlying physical topology; we are currently working on obtaining a tighter bound. From Figure 8, 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 the topology-specific lower bound (see Section V-B) lies only slightly above the overall lower bound that is independent of the specific grooming approach. In other words, our hierarchical approach does not adversely affect the wavelength requirements for a given traffic matrix.

Figures 9 and 10, and Table III are similar to the ones

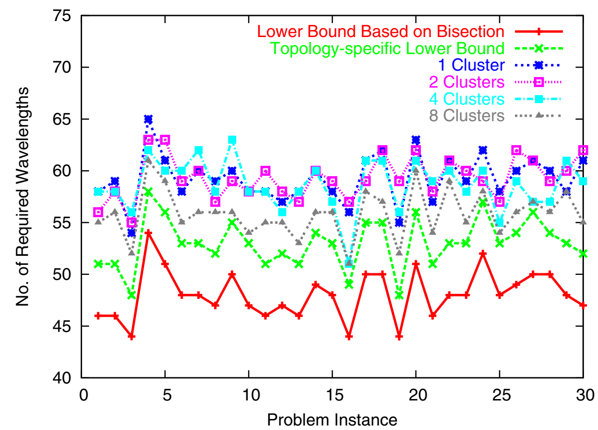


Fig. 8. Number of wavelengths for various numbers of clusters, 32-node network of Figure 1 with random traffic pattern

above, except that they show results for the locality 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 locality pattern, due to the fact that most of the traffic is destined to nodes near by, 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 (Figure 10), that is more pronounced than the one in Figure 8. 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 46%.

#Clusters	Avg LP Length	Avg Max Hub Degree	Wavelengths
1	2.49	484	67
2	2.47	318	62
4	2.34	217	62
8	2.28	164	44

TABLE III

AGGREGATE STATISTICS FOR THE LOCALITY TRAFFIC PATTERN

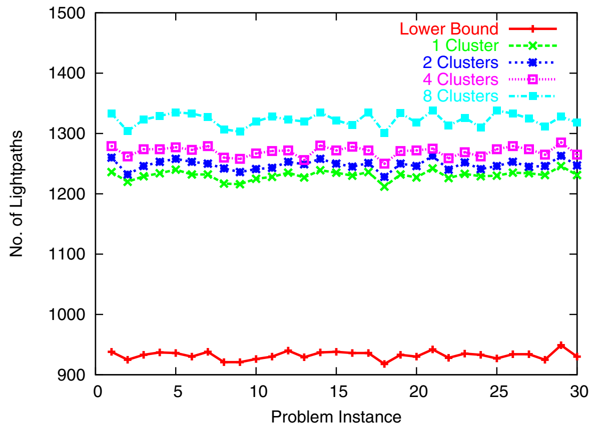


Fig. 9. Number of lightpaths for various numbers of clusters, 32-node network of Figure 1 with the locality traffic pattern

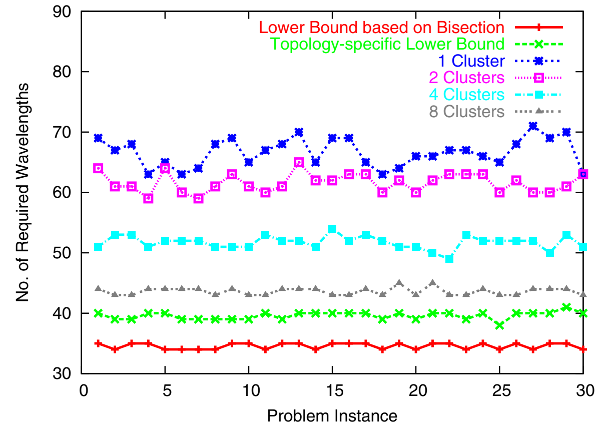


Fig. 10. Number of wavelengths for various numbers of clusters, 32-node network of Figure 1 with the locality traffic pattern

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 lightpaths and wavelength requirements are reasonably close to the corresponding lower bounds. We have also 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 have also shown that the tradeoffs depend on the traffic pattern, as well as on the underlying network topology (although we presented results for a single topology here, the reader is referred to [2] for experiments involving various topologies). Therefore, our current research focus is on developing appropriate clustering techniques for traffic grooming.

VII. Concluding Remarks

We have presented a new framework for efficient and scalable traffic grooming in mesh WDM networks. We are currently developing algorithms which take into account the physical topology and traffic matrix to partition the network nodes into clusters and select hubs. We are also investigating dynamic grooming approaches that leverage the cluster hierarchy and hub infrastructure.

References

[1] S. Baroni and P. Bayvel. Wavelength requirements in arbitrary connected wavelength-routed optical networks. *IEEE/OSA J. Lightwave Technol*, 15(2):242–251, Feb. 1997.

[2] Bensong Chen. *Hierarchical Traffic Grooming in Large-Scale WDM Networks*. PhD thesis, North Carolina State University, Raleigh, NC, August 2005.

[3] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WANS. *IEEE Transactions on Communications*, 40(7):1171–1182, July 1992.

[4] Z. Ding and M. Hamdi. Clustering techniques for traffic grooming in optical WDM mesh networks. In *IEEE GLOBECOM 2002*, volume 3,17-21, pages 2711–2715, Nov 2002.

[5] R. Dutta and G. N. Rouskas. On optimal traffic grooming in WDM rings. *IEEE Journal on Selected Areas in Communications*, 20(1):110–121, January 2002.

[6] R. Dutta and G. N. Rouskas. Traffic grooming in WDM networks: Past and future. *IEEE Network*, 16(6):46–56, November/December 2002.

[7] 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 *NFOEC 2003*, September 2003.

[8] O. Gerstel, R. Ramaswami, and G. Sasaki. Cost-effective traffic grooming in WDM rings. *IEEE/ACM Transactions on Networking*, 8(5):618–630, October 2000.

[9] J.Q. Hu and B. Leida. Traffic grooming, routing, and wavelength assignment in optical WDM mesh networks. *Proceedings of the IEEE INFOCOM 2004*, 23(1):495–501, March 2004.

[10] Z.K.G. Patrocínio Jr. and G. R. Mateus. A lagrangian-based heuristic for traffic grooming in WDM optical networks. In *IEEE GLOBECOM 2003*, volume 5,1-5, pages 2767–2771, Dec 2003.

[11] G. Karypis and V. Kumar. A fast and highly quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 1998.

[12] 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*, pages 218–221, 2001.

[13] C. Lee and E. K. Park. A genetic algorithm for traffic grooming in all-optical mesh networks. In *IEEE SMC 2002*, volume 7, 6-9, Oct 2002.

[14] D. Li, Z. Sun, X. Jia, and K. Makki. Traffic grooming on general topology WDM networks. *IEE Proc. Commun.*, 150(3):197–201, June 2003.

[15] H. Siregar, H. Takagi, and Y. Zhang. Efficient routing and wavelength assignment in wavelength-routed optical networks. *Proc. 7th Asia-Pacific Network Oper. and Mgmt Symposium*, pages 116–127, Oct. 2003.

[16] P-J. Wan, G. Calinescu, L. Liu, and O. Frieder. Grooming of arbitrary traffic in SONET/WDM BLSRs. *IEEE Journal on Selected Areas in Communications*, 18(10):1995–2003, 2000.

[17] D. West. *Introduction to Graph Theory*. Prentice Hall, 1996.

[18] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee. A novel generic graph model for traffic grooming in heterogeneous wdm mesh networks. *IEEE/ACM Transactions on Networking*, 11(2):285–299, April 2003.

[19] K. Zhu and B. Mukherjee. Traffic grooming in an optical WDM mesh network. *IEEE Journal on Selected Areas in Communications*, 20(1):122–133, Jan 2002.