Traffic Grooming in Path, Star, and Tree Networks: Complexity, Bounds, and Algorithms

Shu Huang, Rudra Dutta, and George N. Rouskas

***Abstract*— We consider the problem of traffic grooming in WDM path, star, and tree networks. Traffic grooming is a variant of the well-known logical topology design, and is concerned with the development of techniques for combining low speed traffic components onto high speed channels in order to minimize network cost. Our contribution is two-fold. In the first part of the paper we present a wealth of results which settle the complexity of traffic grooming in path and star networks, by proving that a number of variants of the problem are computationally intractable. Since routing and wavelength assignment in these two topologies is trivial, these results demonstrate that traffic grooming is itself an inherently difficult problem. Our results have implications for ring and other more general topologies, which we explore. In the second part, we design practical grooming algorithms with provable properties. Specifically, for all three topologies, we obtain a series of lower and upper bounds which are increasingly tighter but have considerably higher computational requirements; the series of upper bounds forms an algorithm for the traffic grooming problem with strong performance guarantees. We also present corresponding heuristics with good performance. Our work is a first step towards a formal and systematic approach to the grooming problem in general topologies that builds upon results and algorithms for more elementary networks.**

***Index Terms*— Optical, networks, networking, traffic grooming, virtual topology.**

I. INTRODUCTION

WAVELENGTH division multiplexing (WDM) technology has the potential to satisfy the ever-increasing bandwidth needs of network users on a sustained basis. In WDM networks, nodes are equipped with *optical cross-connects* (OXCs), devices which can optically switch a signal on a wavelength from any input port to any output port, making it possible to establish *lightpath* connections between any pair of network nodes. The set of lightpaths defines a *logical topology*, which can be designed to optimize some performance measure for a given set of traffic demands. The logical topology design problem has been studied extensively in the literature. Typically, the traffic demands have been expressed in terms of whole lightpaths, while the metric of interest has been the number of wavelengths, the maximum congestion level, or a combination of the two. The reader is referred to [7] for a survey and classification of relevant work.

With the deployment of commercial WDM systems, it has become apparent that the cost of network components is one of the dominant costs in building optical networks, and is a more meaningful metric to optimize than, say, the number of wavelengths. Furthermore, with currently available optical technology, the data rate of each wavelength is on the order of 2.5-10 Gbps, while 40 Gbps rates will be commercially feasible in the near future. In order to utilize bandwidth more effectively, new models of optical networks allow several independent traffic streams to *share* the capacity of a lightpath. These observations give rise to the concept of *traffic grooming* [9], [23], a variant of logical topology design, which is concerned with the development of techniques for combining lower speed components onto wavelengths in order to minimize network cost.

Given the wide deployment of SONET/SDH technology and the immediate practical interest of upgrading this infrastructure to WDM, early research in traffic grooming focused almost entirely on ring topologies [2], [8], [13], [14], [22], [25], [28]. More recently, the development of MPLS and GMPLS standards makes it possible to aggregate a set of MPLS streams for transport over a single lightpath. Consequently, traffic grooming in arbitrary physical network topologies is emerging as a research area of both theoretical and practical significance. Although some work in this direction already exists [19], [24], [29], various aspects of the problem remain uninvestigated. In particular, more than heuristics, a formal, systematic approach to the traffic grooming problem is needed whose performance can be characterized, e.g., by tight upper and lower bounds. A worthy goal would be to develop algorithms with formally verified properties, that can be flexibly and efficiently applied in a variety of optical network and cost models.

In this paper we consider the problem of traffic grooming in path, star, and tree networks. Our interest in such elementary topologies is two-fold. Despite their simplicity, these topologies are important in their own right: star networks arise in the interconnection of LANs or MANs with a wide area backbone [21], while passive optical networks (PONs) [20] and cable TV networks (which are increasingly used for high-speed Internet access) are based on a tree topology. Our work can thus be applied to these environments directly. Also, algorithms with provable properties for more elementary networks may be used to attack the traffic grooming problem in general topologies. An example of such an approach can be found in [8], where an algorithm with strong guarantees for ring networks is based on a decomposition into path segments. More recently, we have developed a framework for hierarchical grooming of large general topology networks by decomposing the network into clusters [3]. In this approach,

Manuscript received June 24, 2004; revised June 13, 2005. An earlier version of this paper appeared in the *Proceedings of OPTICOMM 2003*. This work was supported in part by the NSF under grant ANI-0322107.

The authors are with the Department of Computer Science, North Carolina State University, Raleigh, NC 27695-7534 (e-mail: {shuang5, dutta, rouskas}@csc.ncsu.edu).

Digital Object Identifier 10.1109/JSAC-OCN.2006.04006.

we view each cluster as a *virtual* star, and we select a hub node to groom all traffic originating and terminating in the cluster. We apply a grooming algorithm for star networks, similar to the one we describe in this paper, to groom intra-cluster traffic within each cluster, independently of other clusters. At the second level of the hierarchy, the hub nodes form a virtual star for the purpose of grooming intra-cluster traffic. This approach has been shown to provide good grooming solutions for large networks [3]. While a decomposition of general topologies into elementary ones is outside the scope of this paper, the study of path, star, and tree networks can provide insight into the general grooming problem.

We use the *electronic switching cost model* in this paper, described in more detail in Section II-B. We have used the same cost model in previously published research [6], [8]. A related (but not equivalent) cost model is the count of ports required at the electronic switches at network nodes; this is also a useful model and has been used in research reported in the literature. Our cost model is a comparatively finer-grain representation of the electronic switching ability required of a network node, because it does not penalize the termination and origination of traffic at the network nodes, which is after all the purpose of the network and should not be aimed to be minimized by network design.

The electronic switching cost model also has the advantage of representing the traffic-weighted delay in the network exactly. In addition, with current developments in switch technologies, this cost model might well be an accurate representation of actual equipment cost. For example, the Cisco ONS 15454 Multi-Service Transport Platform utilizes *virtual tributary cross-connect* cards in its electronic part which can flexibly switch different traffic components (VTs), and cards can be added to increase the switching capacity. All the above considerations lead us to the conclusion that this cost model is worth study, and we adopt it in this paper.

Our main results are as follows. First, we formally show that traffic grooming is an inherently more difficult problem than logical topology design. Specifically, while routing and wavelength assignment is straightforward in path and star topologies, we prove a number of variants of the grooming problem in these same topologies to be computationally intractable. We also demonstrate the implications of these results to ring and tree networks with nodes capable of full wavelength conversion, i.e., when wavelength assignment is not an issue. We then proceed to provide algorithms for obtaining practical solutions with good properties for path, star, and tree networks. For all three topologies, we obtain a sequence of lower and upper bounds which permit a tradeoff between the quality of the solution and the computational requirements. The sequence of upper bounds yields an algorithm with provable guarantees for the traffic grooming problem in the corresponding topology. We also investigate simple greedy heuristics for each topology, which we find to have good performance.

In Section II we introduce the traffic grooming problem and we present background results on wavelength assignment. In Section III we present the main complexity results. In Sections IV, V, and VI, we present upper and lower bounds on the optimal solution, and we develop corresponding algorithms for path, star, and tree topologies, respectively. We conclude the paper in Section VII.

II. Problem Definition

In this section we review the known results on the complexity of the *routing and wavelength assignment* (RWA) problem, the fundamental problem underlying the logical topology design, and we introduce the traffic grooming problem.

A. The RWA Problem

From a graph theory perspective, an optical network is viewed as a digraph $G = (V, E)$ where each (directed) edge represents an optical fiber link between its endpoints. A request $r = [u, v]$, for a connection from node u to node v, is *satisfied* by (i) assigning to r a path p_r of links in G from u to v, and (ii) assigning to p_r a wavelength to carry the information over the links of p_r (the path p_r and associated wavelength constitute a *lightpath*). Consider a multiset R of connection requests. Then, the RWA problem is to satisfy all requests in R in such a way that if two requests r and r' have paths p_r and $p_{r'}$, respectively, that share a common edge, then they are assigned different wavelengths. The goal is to satisfy all requests in R using the minimum number of wavelengths. In its pure form, the RWA problem assumes no wavelength conversion, i.e., the *same* wavelength is assigned on all links along the path p_r of request r.

First note that if the network G is a tree, then every pair of nodes is joined by a unique path; therefore, part (i) of the RWA problem is trivial for the elementary topologies we consider here. It is known then that wavelength assignment to minimize the number of wavelengths can be solved in polynomial time in paths and stars, but that it is NP-hard in general trees. If G is a path, part (ii) is equivalent to the interval graph coloring problem, which can be solved in linear time by a greedy algorithm [16]. If G is a star, part (ii) is equivalent to finding a minimum edge coloring in a bipartite graph, which is solvable in polynomial time by combining theorems of Hall and Konig [26] (refer also to the proof of Theorem 3.5 in Section III-C). If G is a general tree, the problem of minimizing the number of wavelengths is known to be NP-hard, even in special cases such as binary trees [5]. Finally, the RWA problem is NP-hard in general topologies [17].

B. The Traffic Grooming Problem

In order to utilize bandwidth more effectively, new models of optical networks allow several independent traffic streams to *share the bandwidth* of a lightpath. If the multiplexing and demultiplexing of lower-rate traffic components is performed at the boundaries of the network only (i.e., at edge routers), and the aggregate traffic transparently traverses the optical network of OXCs, this problem is equivalent to RWA. However, it is in general impossible to set up lightpaths between every pair of edge routers, e.g., due to wavelength constraints or constraints on the number of optical transceivers at each router. Therefore, it is natural to consider optical networks in which nodes have both optical (OXC) and electronic switching capabilities. Such nodes let some lightpaths pass through transparently, while they may terminate others. Traffic on terminating lightpaths may then be electronically switched (*groomed*) onto new lightpaths towards the destination node. Introducing some amount of electronic switching within the optical network has two advantages: it significantly enhances the degree of *virtual connectivity* among the edge routers, which otherwise is limited by the number of optical interfaces at each router; it may also drastically reduce the wavelength requirements within the optical network for a given traffic demand. The trade-off is an increase in network cost due to the introduction of expensive active components (i.e., optical transceivers and electronic switches). These observations motivate us to define the following traffic grooming problem.

Let C be the capacity of each wavelength, expressed in units of some arbitrary rate (e.g., OC3); we will refer to parameter C as the *grooming factor*. Let N be the nuimber of nodes in the network, and W be the number of wavelengths that each fiber link in the network can support. We represent a traffic pattern by a demand matrix $T = [t_{ij}]$, where integer t_{ij} denotes the number of traffic streams (each of unit demand) from node i to node j. (We allow the traffic demands to be greater than the capacity of a lightpath, i.e., it is possible that $t_{ij} > C$ for some i, j.) We emphasize that we take the size of a problem instance to be on the order of $\left(N + \sum_{i,j} t_{ij}\right)$, rather than the amount of memory required to store the instance parameters (e.g., the matrix of integers T, the integer N, etc).

Given matrix T on network G, the traffic grooming problem involves the following conceptual subproblems (SPs):

1) *logical topology SP:* find a set R of lightpath requests,
2) *lightpath routing and wavelength assignment SP:* solve the RWA problem on R, and
3) *traffic routing SP:* route each traffic stream through the lightpaths in R.

We emphasize that this is a logical decomposition of the traffic grooming problem, and an optimal algorithm must solve all three subproblems simultaneously.

The goal we consider in this paper is to minimize the *total amount of electronic switching at all network nodes.* In this cost model, every time a lightpath terminates at a network node, one unit of cost is incurred for each unit traffic stream carried by the lightpath if this stream has to undergo electronic switching (i.e., the stream does not have this node as its destination).

We note that the first and third subproblems together constitute the grooming aspect of the problem. Also, in this context, the number W of wavelengths per fiber link is taken into consideration as a constraint rather than as a parameter to be minimized.

A formulation of traffic grooming as an integer linear problem can be found in [9].

III. Complexity Results

A. Path Networks

We consider a network in the form of a unidirectional path $\mathcal{P}$ with N nodes. There is a single directed fiber link from node i to node $i+1$, for each $i \in \{1, 2, \cdots, N-1\}$. An instance of the traffic grooming problem is provided by specifying a number N of nodes in the path, a traffic matrix $T = [t_{ij}], 1 \le i < j \le N$, a grooming factor C, a number of wavelengths W, and a goal F. The problem asks whether a valid logical topology may be formed on the path and all traffic in T routed over the lightpaths of the logical topology so that the total electronic switching over all path nodes is less than or equal to F. As we discussed above, we take the size of a problem instance to be in the order of $\left(N + \sum_{i,j} t_{ij}\right)$.

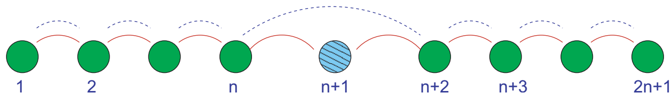

Fig. 1. Example of path construction for the proof of Theorem 3.1, $N = 2n+1$, $W = 2$

We first consider the case where bifurcated routing of traffic is not allowed. Specifically, for any source-destination pair (i, j) such that $t_{ij} \le C$, we require that all t_{ij} traffic units be carried on the *same* sequence of lightpaths from source i to destination j. On the other hand, if $t_{ij} > C$, it is not possible to carry all the traffic on the same lightpath. In this case, we allow the traffic demand to be split into $\lfloor \frac{t_{ij}}{C} \rfloor$ subcomponents of magnitude C and at most one subcomponent of magnitude less than C, and the no-bifurcation requirement applies to each subcomponent independently.

The following theorem settles the complexity of the traffic grooming problem in paths. Our proof is more general, and uses a reduction from a different NP-Complete problem, than the one in [4]; the latter considered a special case where all traffic is destined to the rightmost node in the path. As we shall see shortly, our proof provides insight into the inherent difficulty of the grooming problem.

Theorem 3.1: The decision version of the traffic grooming problem in unidirectional path networks (bifurcated routing of traffic not allowed) is NP-Complete.

Proof. The reduction is from the Subsets Sum problem [12]. An instance of the Subsets Sum problem consists of n elements of size $s_i \in Z^+ \ \forall i \in \{1, 2, \cdots, n\}$, and a goal sum B. The question is whether there exists a subset of elements whose sizes total to B. Let $B_1 = \max\{B, \sum_i s_i - B\}$. (For the purpose of the Subsets Sum problem, posing the instance with B or B_1 is equivalent.) Construct a path network using the following transformation: $N = 2n+1$, $W = 2$, $C = \sum_i s_i + 1$, $F = n\sum_i s_i - B_1$, and traffic matrix:

$$t_{ij} = \begin{cases} C+1, & i \in \{1, 2, \cdots, n-1\} \\ & \quad \cup\{n+1, n+2, \cdots, 2n\}, j = i+1 \\ B_1 + 1, & i = n, j = n+1, \text{or } i = n+1, j = n+2 \\ C - B_1, & i = n, j = n+2 \\ s_i, & i \in \{1, 2, \cdots, n\}, j = i+n+1 \\ 0, & \text{otherwise} \end{cases}$$

Due to the traffic components of magnitude $C+1$, both wavelengths must be used to form single-hop lightpaths over all physical links except the two central ones. Over the two central links, at least one single-hop lightpath each must be formed due to the traffic components of magnitude $B_1 + 1$; this quantity is less than C for $0 < B < \sum_i s_i$, i.e., when the Subsets Sum instance is non-trivial, and hence it can always fit in one wavelength. The other wavelength may be used to form two single-hop lightpaths over these two links, or a single two-hop lightpath over them. The electronic switching cost in the former case is at least as large as in the latter case, thus it suffices to consider the latter case. Thus, the logical topology, i.e., the set of lightpaths R in the first subproblem of Section II-B, may be considered forced on us by the construction; this topology is shown in Fig. 1. On this logical topology, each of the traffic components corresponding to the object sizes of the Subsets Sum problem must be electronically switched exactly $n-1$ times at all nodes other than node $n+1$. At node $n+1$, at most C units of traffic can be optically switched, since only one lightpath passes through optically. The $C-B_1$ units of $t_{n,n+2}$ must be carried on the wavelength that bypasses node $n+1$, since traffic cannot be bifurcated and the other wavelength does not have enough capacity for it. Thus, there remains room for at most B_1 units of the traffic corresponding to the object sizes of the Subsets Sum problem to optically bypass node $n+1$. To satisfy the electronic switching goal F, at least this much traffic must be optically passed through node $n+1$, and because traffic cannot be bifurcated, the electronic switching goal can be satisfied *iff* there is a subset of objects in the Subsets Sum problem instance whose sizes total to B_1, that is, *iff* the Subsets Sum problem instance can be satisfied. Since deciding the satisfiability of the Subsets Sum problem is NP-Complete, the theorem is proved. ■

Because of the construction in the above proof, we have the following corollary. This corollary demonstrates that, even when solutions to the first two subproblems of the traffic grooming problem (refer to Section II-B) are provided, the problem remains NP-Complete by virtue of the third subproblem (traffic routing). Therefore, traffic grooming is inherently more difficult than the well-known NP-Complete RWA problem.

Corollary 3.1: The decision version of the traffic grooming problem in unidirectional path networks (bifurcated routing of traffic not allowed) is NP-Complete even when a logical topology is provided.

Note: Because of the construction in the proof of Theorem 3.1, the only feasible assignment of the traffic to the logical topology is the one that satisfies the grooming goal F. Thus, F can be assigned a larger value without affecting the satisfiability of the instance. In particular, using $F = n\sum_i s_i + C - B_1$ will have the same result. Since this is the maximum possible electronic switching cost for the problem instance (every traffic component is electronically switched at every intermediate node), it is also proved that the problem of deciding whether a *given* logical topology admits of any feasible routing of traffic at all is also NP-Complete.

We now extend the above results to the case where bifurcated routing of traffic is allowed. Specifically, a traffic component t_{ij} is allowed to be split into various subcomponents which may follow different routes (i.e., different lightpath sequences for a path network) from source to destination. The bifurcation is restricted to integer subcomponents.

Theorem 3.2: The decision version of the traffic grooming problem in unidirectional path networks (bifurcated routing of traffic allowed) is NP-Complete.

Proof. The proof is appended as Appendix I.

Again, by the nature of the proof, we have the following corollary.

Corollary 3.2: The decision version of the traffic grooming problem in unidirectional path networks (bifurcated routing of traffic allowed) is NP-Complete even when a candidate logical topology is provided.

Let us now consider the implications of these results for related topologies, *bidirectional* path networks, and *ring* networks (unidirectional and bidirectional). The implications for ring networks are of practical importance, even though the NP-hard nature of traffic grooming for ring networks has already been demonstrated. In particular, it is known that the RWA problem in rings is NP-hard [27]. However, the following lemmas show that *even if all ring nodes are equipped with wavelength converters* (in which case wavelength assignment is trivial), traffic grooming remains a difficult problem. We state below three lemmas which settle the question for these topologies without proof. In each case, the proof is straightforward and can be obtained by an appropriate restriction of the traffic matrix. The interested reader is referred to [18].

For bidirectional paths, we assume that between every two adjacent nodes there are now two links, each carrying W wavelengths, in opposite directions. Traffic components are now allowed from every node to every other node, not just in a single direction, as we have assumed so far. A traffic component is allowed to be carried from source node s to destination node d on a sequence of lightpaths some of which are in one direction and some in the reverse direction; thus a traffic component may traverse the same link multiple times in either direction. It is clear, however, that a traffic component must traverse the outgoing link from node s in the direction in which node d lies at least once, and the incoming link to node d from the direction in which node s lies at least once.

Lemma 3.1: The decision version of the traffic grooming problem in bidirectional path networks (in both the cases of bifurcated routing of traffic allowed and not allowed) is NP-Complete.

Lemma 3.2: The decision version of the traffic grooming problem in unidirectional ring networks (in both the cases of bifurcated routing of traffic allowed and not allowed) is NP-Complete, even when every node has full wavelength conversion capability.

Lemma 3.3: The decision version of the traffic grooming problem in bidirectional ring networks (in both the cases of bifurcated routing of traffic allowed and not allowed) is NP-Complete, even when every node has full wavelength conversion capability.

Considering that the goal of the grooming problem (in unidirectional paths) with bifurcation allowed is bounded by a polynomial of the number of nodes and the maximum traffic component, the problem obviously is not amenable to a fully polynomial time approximation scheme (FPTAS) unless $P = NP$. However, it might be hoped that approximation algorithms may exist for some useful approximation ratios. We show below that this is not true.

Theorem 3.3: Constant-factor approximation of the optimization version of the unidirectional path network grooming problem (bifurcated routing of traffic allowed) is NP-hard.

An instance of the problem is provided exactly as for the proposition for the decision problem of traffic grooming with bifurcation allowed. Now the problem is to find the grooming solution which produces the minimum amount of electronic routing.

Proof. The proof is appended as Appendix II.

Since general network topologies, including most interesting topology families such as spiders, rings, grids, tori etc. contain the path network as a subfamily, the above result shows that it is not practical to attempt optimal or constant ratio approximate solutions to the grooming problem in these cases. The only topology family which does not include the paths is a star topology. We consider star networks next, and show that the problem is again NP-Complete. Whether approximations are possible for star networks is a question that remains open at this time.

Note: An approximation algorithm for ring network grooming is given in [15], and an optimal algorithm for some values in [1]. However, these results do not contradict ours, because in both cases the restrictive assumption is made that traffic cannot be switched between lightpaths of different wavelengths even when two such lightpaths have endpoints at the same node. This is equivalent to the assumption of no SONET cross-connects as in [4]. In addition, the study in [1] addresses only the all-to-all unitary traffic pattern.

B. Star Networks

We consider a network in the form of a star $\mathcal{S}$ with $N+1$ nodes. There is a single *hub* node which is connected to every other node by a physical link. The N nodes other than the hub are numbered from 1 to N in some arbitrary order, and the hub node is numbered 0. Each physical link consists of a fiber in each direction, and each fiber can carry W wavelengths. The grooming factor C, the traffic matrix $T = [t_{ij}]$, and the goal F have the same significance as before.

In a star network, no node except for the hub switches traffic, either electronically or optically. In other words, the hub is the only node which sees traffic neither originated by, nor destined to, itself. Thus, there will be only two kinds of lightpaths in the logical topology: single-hop lightpaths which either originate at a non-hub node and terminate at the hub, or vice versa; and two-hop lightpaths that originate and terminate at non-hub nodes, and are switched optically at the hub. Bifurcation of traffic may or may not be allowed, giving two flavors of the problem; if it is allowed, then the bifurcation must be in integer subcomponents only. The question is whether a valid logical topology and assignment of all traffic in T to the lightpaths of the logical topology can be found which results in F or less electronic switching at the hub. Note that the logical topology design and traffic assignment can be simply expressed as deciding which of the traffic components between non-hub nodes are allocated a lightpath and which are electronically switched through the hub on two single-hop lightpaths: a two-hop lightpath from node i to j can only carry traffic from component t_{ij}, thus the traffic assignment is implicit in the logical topology.

Theorem 3.4: The decision version of traffic grooming in star networks is NP-Complete.

Proof. The proof is appended as Appendix III.

C. Tree Networks

The traffic grooming problem in trees is NP-Complete, since the RWA subproblem is NP-Complete [5]. However, the following result shows that traffic grooming remains intractable even when all interior tree nodes are equipped with wavelength converters.

Corollary 3.3: The decision version of traffic grooming in tree networks is NP-Complete, even when every interior tree node has full wavelength conversion capability.

Proof. This result follows from Theorem 3.4 since the tree can be restricted to the star. In particular, the proof of Theorem 3.4 goes through even if wavelength conversion is available at the hub node of the star. ■

While the RWA problem in tree networks is NP-Complete, even for logical topologies in which no lightpath is longer than three physical hops [17], to the best of our knowledge, the complexity of the RWA problem for logical topologies with single-hop and 2-hop lightpaths has not been settled. The next theorem formally proves that this special RWA problem in trees is, in fact, tractable. We use this result for developing heuristic approaches to the traffic grooming problem in tree networks in Section VI.

Theorem 3.5: Any logical topology containing no lightpaths longer than two physical hops on a tree network has a valid wavelength assignment using no more than L wavelengths, where L is the maximum lightpath loading on any directed link in the tree, and such an assignment can be found in polynomial time.

Proof. Consider all the interior nodes of the tree, and arbitrarily designate one of them to be the root. This establishes a parent-child partial ordering throughout the tree. Traverse the interior nodes of the tree in some order not violating this partial order. At each node, consider only the 2-hop lightpaths which optically pass through that node. From the point of view of the node, these lightpaths are similar to lightpaths in a star network of which this node is the hub. (Note that each 2-hop lightpath is in exactly one of these star networks.)

We know such a set of lightpaths is always possible to color. The wavelength assignment procedure is, in brief, as follows: consider a bipartite multigraph, in which the set of nodes in each part corresponds to the set of star network nodes other than the hub, and the arcs correspond to the lightpaths. Then, the graph is at most L-full because of the lightpath loading condition. Therefore the set of arcs can be partitioned into at most L 1-factors (subsets containing arcs without common nodes) [26]. The arcs of each 1-factor can be colored using the same color, therefore L colors suffice to color all the arcs. However, many actual colorings are possible by permuting the colors among the 1-factors.

Consider the star network seen by the root of the tree. The lightpaths in this network can be colored by arbitrarily picking L_0 colors out of the available ones, where L_0 is the maximum number of 2-hop lightpaths traversing any directed link to which the root is connected. Now consider each child i of the root, and let L_i denote the 2-hop lightpaths passing through this child *and* either terminate or originate at the root. Child i of the root must be able to color these 2-hop lightpaths similarly, using no more than L_i colors. But all the colors may not be available on the link connecting the root to node

i, because they have been used by the coloring chosen by the root. However, *some* L_i colors must be available, because we assumed that L colors are available on each link, and we must have $L_0 + L_i \leq L$, otherwise the loading on that link exceeds L. Thus, some wavelength assignment exists for each child i reconciling that chosen by the root. This strategy continues to hold for each parent and its children, thus the entire topology can be colored. The core of the algorithm consists of at most $N-2$ repetitions of the star wavelength assignment algorithm, which is polynomial; therefore so is this algorithm. ■

IV. Bounds and Algorithms for Path Networks

In order to derive bounds on the optimal solution to the traffic grooming problem, we note that a path can be viewed as a ring network with one link removed. In [8], a sequence of upper and lower bounds on the optimal solution for ring networks was obtained by decomposing the ring into path segments. Each (upper or lower) bound in the sequence improves over the previous one, but it takes increasingly longer computational time to obtain. Furthermore, the sequence of upper bounds represent solutions to the traffic grooming problem whose performance is precisely characterized. The decomposition is effected by considering certain ring nodes to be completely *opaque*, i.e., insisting that no lightpaths optically pass through these nodes; in essence, opaque nodes electronically switch all traffic passing through them. For details, the reader is referred to [8].

The exact same approach can be used to obtain a sequence of increasingly tighter lower and upper bounds for path networks. In this case, the path is decomposed into smaller segments by making some of the nodes completely opaque. In particular, each upper bound $k, k = 1, 2, \cdots$, in the sequence is obtained as the amount of electronic switching in some logical topology in which the distance between adjacent opaque nodes is no more than k links apart; the logical topology is obtained by optimally solving the traffic grooming problem for all path segments between opaque nodes. For $k = 2$, the logical topology is such that either all the odd-numbered or the even-numbered nodes are opaque, i.e., it consists of either single-hop or two-hop lightpaths. It is straightforward to verify (refer also to [8]) that the upper bound corresponding to $k = 2$ is no worse than one-half the worst-case amount of the electronic switching in a completely opaque topology, i.e., when all lightpaths are single-hop. This result is used in the next section to develop an algorithm with performance guarantees.

A. A Greedy Heuristic

We now present a simple greedy heuristic for the case when bifurcation of traffic is allowed. Before we proceed, we introduce the concept of *reduction* of a traffic matrix. Specifically, we reduce the matrix T so that all elements are *less than* the capacity C of a single wavelength, by assigning a whole lightpath to traffic between a given source-destination pair that can fill it up completely. The available wavelengths on the links of the path segment from the source to the destination node are also decremented by the number of lightpaths thus assigned. Since breaking such lightpaths would increase the electronic switching cost by C, and since any benefit we can get by having that wavelength available for grooming traffic cannot exceed C, this procedure does not preclude us from reaching an optimal solution[1]. We continue using the same notation for the traffic matrix and traffic components, but in what follows they stand for the same quantities after the reduction process.

Consider the logical topology obtained by assigning a lightpath to *each* non-zero traffic component t_{ij} of the reduced traffic matrix T. We will call this the completely *transparent* topology. If this logical topology is feasible, i.e., the number of wavelengths required does not exceed the number of wavelengths available at any link of the path network, then it is also optimal since the electronic switching cost is zero. Typically, however, this topology will be infeasible; to obtain a feasible topology traffic must be *groomed*, that is, some traffic components must be carried over a sequence of lightpaths from source to destination, rather than on a direct lightpath. Based on these observations, the greedy algorithm consists of the following steps:

1) List the non-zero multihop components of matrix T in some order, and let t_{ij} be the first component in the list.
2) Determine a sequence of lightpaths $(l_1, \cdots, l_r)$ over which to route component t_{ij}. The sequence of lightpaths is such that the source of l_1 is node i, the destination of l_r is node j, and for each $k, k = 1, \cdots, r-1$, the destination of lightpath l_k is the source of lightpath l_{k+1}.
3) For each lightpath l_k in the sequence of Step 2, let s and d be its source and destination nodes, respectively; add an amount of traffic equal to t_{ij} to the component t_{sd}.
4) Set the traffic component t_{ij} to zero.
5) Reduce the new traffic matrix T using the procedure we described above (note that by adding t_{ij} units of traffic to some components may make them larger than, or equal to, C).
6) If the completely transparent topology corresponding to the new, reduced traffic matrix T is feasible, stop. Otherwise, repeat from Step 1.

The performance of the algorithm depends on (a) the order in which the traffic components are listed in Step 1, and (b) how the sequence of lightpaths $(l_1, \cdots, l_r)$ over which to route a traffic component is determined in Step 2. After extensive experimentation with a number of rules, we have found that the following one works well and is the one we use: list traffic components in the order of the distance they travel; break ties by listing smaller components first, and break further ties arbitrarily.

Let t_{ij} be the component selected at Step 1 of the algorithm; initially, this is carried on a direct lightpath l from i to j. Let $(m, m+1), i < m < j$, be the most congested link in the path of this component. Consider the logical topology that provides the second upper bound, as we discussed in the previous subsection. In this topology either node m or node

[1] Although reduction does not affect optimality in this case, we note that reduction may preclude reaching an optimal solution in general topologies or in other variants of the traffic grooming problem.

$m+1$ is opaque. We have two versions of the algorithm, as follows. In the first version, we break lightpath l at either node m *or* node $m-1$, whichever is opaque; as a result, the component is carried on two lightpaths, one from i to the opaque node among $m-1$ and m, and one from the opaque node to j. In the second version, we break lightpath l at the same node as in the first version, *and* either at node $m+1$ or node $m+2$, whichever is opaque. In this case, the component is carried on three lightpaths, one from i to the opaque node among $m-1$ and m, a two-hop lightpath to the next opaque node, and finally on another lightpath to j. In both versions, if the first (respectively, last) link is the most congested one, we break the lightpath only at the first (respectively, last) opaque node other than the source (respectively, destination) node.

The running time of both versions of the algorithm is $O(N^3)$. The algorithm is guaranteed to do no worse than the logical topology that provides the second upper bound; the proof is straightforward and is omitted. However, the algorithm can actually do much better than this upper bound, since when it terminates some components may be carried over lightpaths longer than two hops.

B. Numerical Results

The performance measure of interest in our study is the *normalized electronic switching cost* of a solution. The normalized electronic switching cost is defined as the total amount of electronic switching for this solution, expressed as a fraction of the amount of electronic switching for the completely opaque logical topology, i.e., when all traffic is carried on single-hop lightpaths. Consequently, the *lower* the value, the *better* the grooming effectiveness of an algorithm.

Figs. 2-3 plot the normalized electronic switching cost for a number of path problem instances with $N = 30$ nodes, $W = 10$ or 80 wavelengths, and grooming factor $C = 32$ or 128. Each figure shows results for 30 different instances. The instances in Fig. 2 were randomly generated such that the traffic pattern is distance-independent (i.e., uniform pattern), while those in Fig. 3 were randomly generated to follow a distance-dependent traffic pattern such that traffic demands increase with the distance between source and destination nodes. The figures do not show the lower bounds because, due to computational considerations, for the large path networks we consider here, we were able to obtain only a part of the series of lower bounds, and these are always zero. Since a path is a special case of a ring, the reader is referred to [8] for an extensive study of the behavior of the lower bounds in rings of various sizes.

As we can see, both versions of the algorithm have very similar performance across all instances, with the first version performing slightly better than the second one; this is due to the fact that the first version carries each component over two lightpaths (instead of three for the second version), incurring less electronic switching cost. We also observe that both versions perform better than the topology that provides the 2-hop opaque upper bound (0.5), and that in some cases the solutions obtained by the algorithms have a normalized cost around 0.1-0.15, well below the upper bound. Through extensive experimentation, we have found that the difference

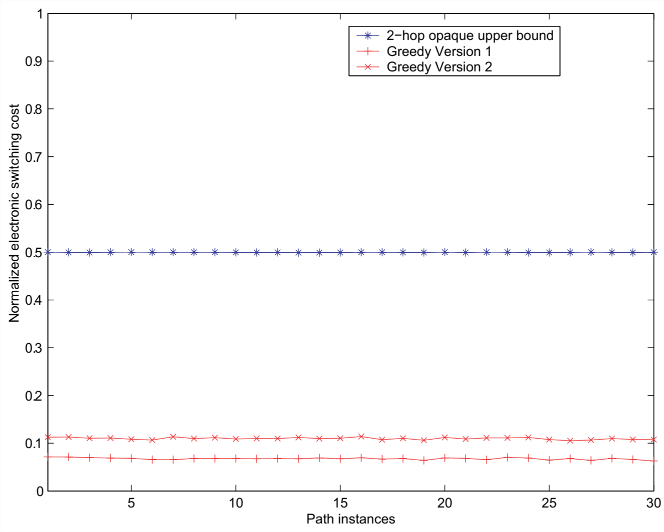

Fig. 2. Path result: $N = 30, W = 80, C = 32$, uniform pattern

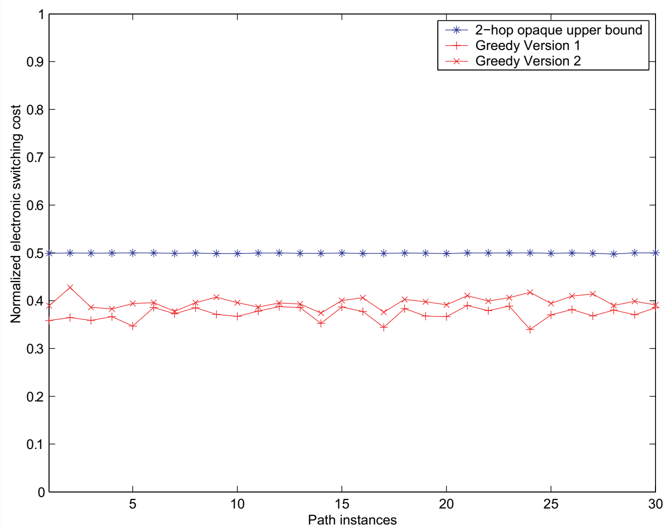

Fig. 3. Path result: $N = 30, W = 10, C = 128$, rising pattern

in the normalized cost values between the two figures is not due to the traffic pattern, but rather due to the grooming factor used. Specifically, a larger grooming factor C leads to higher values of normalized electronic switching cost since when a lightpath is broken, a large number of traffic components will have to be electronically switched.

We have obtained results very similar to the ones shown in Figs. 2-3 for a wide range of values for the parameters N, C, and W, and for various traffic patterns. We conclude that it is possible to obtain good solutions to the traffic grooming problem for long paths with modest computational effort.

V. Bounds and Algorithms for Star Networks

The search space of the problem is quite large: for N non-hub nodes, each of the $N(N-1)$ traffic components may be either electronically switched at the hub or optically bypass it, and a brute-force algorithm would have to evaluate a space of $2^{N(N-1)}$ combinations. Many of these would not be valid solutions; we use the term "valid" to denote solutions which do not violate any wavelength or traffic constraints, and hence can actually be implemented as a logical topology. A *partial* solution (i.e., one which defines *some* lightpaths but leaves others indeterminate) is called valid if there is at least one valid complete solution of which the partial solution is a subset.

We now present an algorithm which builds solutions incrementally, visiting each possible solution exactly once while avoiding any invalid solution. We employ pruning techniques to further reduce the search. While the average case is improved, the worst case running time of the algorithm remains very large. However, the incremental nature of the algorithm allows practical benefits to be obtained without completing the exhaustive search. We demonstrate how increasingly good bounds, both upper and lower, may be obtained with successively more computation.

A. Examining Solutions

Our strategy is to examine partial or complete solutions to a given problem instance in the form of a traversal of a tree formed by the partial solutions. Pruning is applied on this tree to reduce the search space, as in standard tree searches and branch-and-bound methods [12]. However, the key idea of our approach is to construct the tree in a manner that significantly reduces the search space. Specifically, only valid solutions (complete or partial) are allowed to appear in the tree. Among these, our aim is to examine more of those which are maximally useful. In what follows, we consider the traffic matrix T after the reduction process we described in Section IV-A.

Consider an $N \times N$ *mask* matrix $M = [m_{ij}]$, the rows (respectively, columns) of which correspond to the N rows (resp., columns) of the traffic matrix T other than the row (resp., column) that includes traffic components originating (resp., terminating) at the hub. In the mask matrix, every element outside the diagonal has one of two values: "E" or "O." (The traffic from a node to itself is always zero, thus the diagonal elements can be safely ignored.) A mask matrix stands for a proposed solution to a given problem instance. A value of "E" for m_{ij} indicates that the corresponding traffic component t_{ij} is carried on two single-hop lightpaths to its destination in this solution, and thus, it is electronically switched at the hub. On the other hand, a value of "O" indicates that the traffic component is carried directly to its destination node on a two-hop lightpath, and it is optically routed by the OXC at the hub.

There are $2^{N(N-1)}$ possible mask matrices for a given traffic matrix. To create a natural progression in which candidate solutions may be examined, we introduce the concept of *partial* mask matrices representing partial solutions. We introduce a third value of "U" (for "unassigned") that the partial mask matrix elements may take, which simply indicates that the partial solution does not specify whether the corresponding traffic component should be electronically switched or optically routed. It may appear that we have actually made the problem harder, by increasing the search space to $3^{N(N-1)}$. However, this actually allows us to create an efficient search algorithm; since a partial mask matrix stands for multiple complete solutions (all combinations of "E" and "O" for the

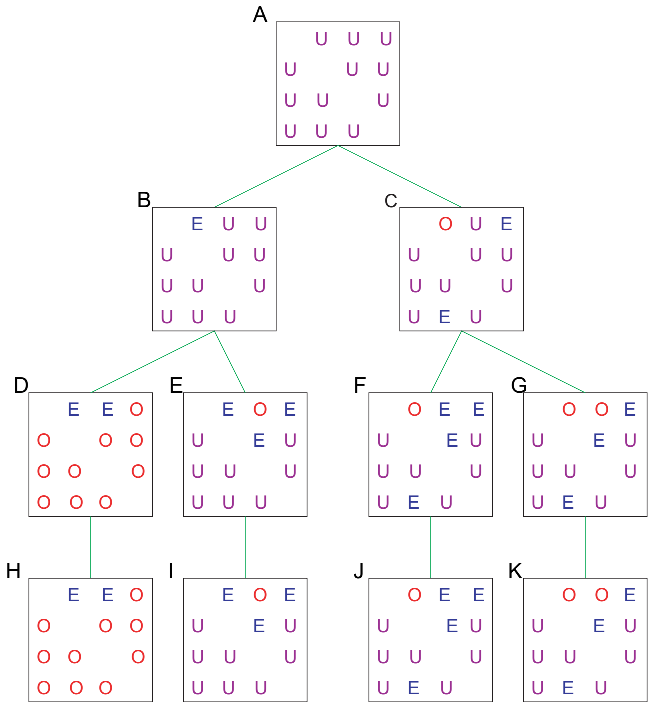

Fig. 4. Generating the search tree for a star network.

"U" elements), pruning such a matrix as we describe below, can eliminate a large number of complete solutions.

We traverse a binary search tree in which each vertex is a mask matrix, such as the one shown in Fig. 4. All vertices other than leaf vertices represent partial mask matrices, whereas leaf vertices are complete mask matrices. (Due to pruning, some non-leaf vertices may become complete mask matrices as well, as explained below.) We start with a partial mask matrix in which all elements have the value "U", or unassigned; this matrix forms the root of the search tree. We generate level $i+1$ of the tree by selecting a *single* traffic component and generating two children for each vertex at level i, one for which the corresponding element of the partial mask matrix is set to "E" and one for which it is set to "O". For instance, in Fig. 4, partial mask matrices B and C (the second level of the tree) are derived from partial matrix A (the root) by changing the element in the first row and second column of matrix A from "U" to "E" and "O," respectively. Let Π_t denote the ordering $(t_1, t_2, t_3, \cdots)$ such that level i of the search tree is generated by setting the mask element corresponding to t_i. A complete search will always generate a full binary tree of depth $N(N-1)$, and will generate all possible complete mask matrices as leaves, no matter what Π_t is. However, we may wish to stop before generating the full tree and extract partial information in the form of bounds; also, pruning will allow us to avoid generating all the leaves even for an exhaustive search. In these cases, changing the ordering Π_t would change the bounds we obtain, or the amount of pruning possible; thus the search is characterized by Π_t.

1) Pruning for Invalidity: Some mask matrices represent invalid solutions to the problem instance. For example, if a mask matrix has W elements of the same row set to "O", it indicates that W two-hop lightpaths are to be set up from

$$\left(\sum_{d \in \{1,\cdots,N\}, d \neq i} \overline{I_O^{(id)}(M)}\, t_{id} \right) + t_{i0} \quad \leq \quad \left(W_o(i) - \sum_{d \in \{1,\cdots,N\}, d \neq i} I_O^{(id)}(M) \right) C \quad , \qquad i = 1, \cdots, N \tag{1}$$

$$\left(\sum_{s \in \{1,\cdots,N\}, s \neq i} \overline{I_O^{(si)}(M)}\, t_{si} \right) + t_{0i} \quad \leq \quad \left(W_t(i) - \sum_{s \in \{1,\cdots,N\}, s \neq i} I_O^{(si)}(M) \right) C \quad , \qquad i = 1, \cdots, N \tag{2}$$

the corresponding node to other non-hub nodes. But if even one of the other $N+1-W$ traffic components sourced by that node is non-zero, then such a solution is invalid because there is no wavelength left to carry this traffic. Hence, for each non-hub node i of the star network, the conditions in Eq. (1) and (2), shown at the top of the next page, must be satisfied for any valid (partial or complete) mask matrix M. In the above expressions, $I_O^{(sd)}(M)$ and $\overline{I_O^{(sd)}(M)}$ indicate whether the mask element m_{sd} is "O" or not. Specifically, $I_O^{(sd)}(M)$ is 1 if m_{sd} is "O", otherwise it is 0; $\overline{I_O^{(sd)}(M)}$ is its inverse indicator. We have also used $W_o(i) \leq W$ and $W_t(i) \leq W$ to denote the number of wavelengths available for node i to originate and terminate lightpaths, respectively, after the reduction process mentioned in Section IV-A. The term in parentheses in the right-hand side of (1) represents the number of wavelengths that are available at non-hub node i to source traffic (the term $W_o(i)$) and the two-hop lightpaths already specified in matrix M (the sum within the parentheses). Therefore, the right-hand side of (1) is the available outgoing traffic capacity on the link from node i to the hub. The left-hand side of (1) is the traffic demand out of node i that has not been assigned to a two-hop lightpath in matrix M, including the traffic to the hub node. This total traffic demand must be at most equal to the capacity of the available wavelengths for matrix M to represent a valid partial solution. Expression (2) is similar, but refers to the traffic demands and capacity to (rather than from) node i.

As we generate the search tree, we can avoid generating the invalid cases by applying (1) and (2). Whenever we generate a new mask matrix by changing the status of a traffic component t_i from "U" to "O", it may be that another element t_j (which currently has the value of "U") cannot be set to "O" as well without violating the above conditions. (Only elements in the same row or column as the mask element t_i need be examined.) In this case, t_i being set to "O" *forces* some t_j in the same row or column to be "E", and we set the mask matrix accordingly. Returning to Fig. 4, we note that matrix C is generated by changing the element in the first column and second row of matrix A to "O" from "U". Assuming an appropriate problem instance, changing this element to "O" forces two other elements, one in the same row and another in the same column to be changed to "E"; thus, matrix C differs from matrix A in three positions rather than one. In doing so, in effect we prune the part of the search tree which contains partial mask matrices with t_i set to "O" and t_j either "O" or "U". A mask element being set to "E" does not force any other traffic element, so there is no corresponding pruning in that case.

A special case of this pruning may be applied at the root of the search tree, if there are traffic elements that cannot be assigned to two-hop lightpaths in any valid solution. Thus, we set the corresponding mask elements to be "E" at the beginning, and start with the resulting partial mask matrix at the root, instead of one with every element set to "U".

2) Pruning for Suboptimality: With the above method, we create a search tree in which only mask matrices corresponding to valid solutions appear. We now outline how to prune the tree of certain suboptimal mask matrices, further reducing the search space.

A partial mask matrix corresponding to an internal vertex of the search tree has some elements still set to "U". In general, *all* of these elements cannot be set to "O" without violating the conditions (1) and (2). However, if the partial mask matrix is "close" to a solution in the sense that most resource conflicts have been already resolved in it, then it may be possible to set all "U" elements to "O". This is equivalent to pruning the entire subtree rooted at the vertex representing the partial mask matrix, and replacing it with the leaf vertex *of that subtree* which yields the optimal solution within that subtree. An example of this pruning in Fig. 4 is in going from matrix B to matrix D.

A second method of pruning suboptimal solutions is the familiar one applied in branch-and-bound techniques. Let $\mathcal{L}$ denote the set of vertices of the tree for which no children have been generated yet. As we show in the next section, we can extract upper and lower bounds on the value of the best solution that can be obtained in the portion of the tree rooted at any vertex $v \in \mathcal{L}$. Hence, we can safely prune the subtree rooted at a vertex v if the lower bound obtained from v is greater than the upper bound obtained from at least one other vertex in $\mathcal{L}$.

The operation of the algorithm using the above procedures is illustrated in Fig. 4. We show the search tree generated up to three levels. In this example, $N=4$ and the first three traffic elements of the ordering Π_t are t_{12}, t_{13}, t_{14}. Matrix A is the starting partial mask at the root with all elements set to "U.". Matrices B and C are generated from A by setting the mask element m_{12} to "E" and "O," respectively. In C, this forces two other mask elements, m_{14} and m_{42}, to become "E," due to pruning for invalidity. The same kind of pruning is observed when generating E from B and G from C respectively, at the next level; in both cases, m_{13} is set to "O," which forces two traffic components to be set to "E" in mask E and one in G. D is generated from B by setting m_{13} to "E"; this results in a partial matrix which yields a valid mask when all remaining "U" elements are set to "O." This new mask matrix replaces D due to pruning for suboptimality. The next level is generated by the traffic element t_{14}; all four partial masks D, E, F and G produce a single child identical to themselves because the mask element m_{14} is already set in each of them.

B. Bounds

Despite the pruning procedures described above, the computation required to complete the exhaustive search for any given problem instance may still be intractably large. We now describe how to extract lower and upper bounds on the optimal value of electronic switching from any intermediate state of the tree. For ease of discussion, we assume that the intermediate trees are those resulting from generating the tree fully up to a given level and no further.

1) Upper Bounds: In general, a matrix at the tree vertex $v \in \mathcal{L}$ will have some elements set to "E" and "O", and some "U." We can always generate a valid solution from such a matrix by converting all the "U" to "E." Furthermore, the best solution in the subtree rooted at v cannot be any worse than this. Let us denote the value of electronic switching obtained from this "pessimistic" solution as $\psi(v)$:

$$\psi(v) = \sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} \overline{I_O^{(sd)}(v)} \tag{3}$$

Then, $\psi(v)$ is an upper bound on the best (lowest) amount of electronic switching that can be obtained from the solutions in the subtree rooted at v. Let us denote by $\mathcal{L}_i$ the set $\mathcal{L}$ obtained after generating the tree completely up to level i. Then we can define a series $\{\Psi_i\}$ of upper bounds as:

$$\Psi_i = \min_{v \in \mathcal{L}_i} \psi(v) \tag{4}$$

Let u_E and u_O be the two children of v, u_E is the matrix in which the traffic component, say m_i, generating the new level is set to "E" and u_O the one in which it is set to "O." There are three possible cases:

1) m_i was set to "U" in v, therefore both u_E and u_O are present; in this case $\psi(u_E) = \psi(v)$, and $\psi(u_O) < \psi(v)$.
2) m_i was set to "E" in v because of an earlier pruning for feasibility; then, u_E is generated, and $\psi(u_E) = \psi(v)$.
3) m_i was set to "O" in v because of an earlier pruning for suboptimality; only u_O is generated, and $\psi(u_O) = \psi(v)$.

Thus, we assert that $\min\{\psi(u_E), \psi(u_O)\} \leq \psi(v)$. Since every vertex $v \in \mathcal{L}_i$ generates children to form $\mathcal{L}_{i+1}$, we have that $\{\Psi_i\}$ is a strong sequence of upper bounds:

$$\Psi_{i+1} \leq \Psi_i, \quad \forall i \in 1, 2, \cdots, N(N-1) - 1 \tag{5}$$

2) Lower Bounds: We can similarly obtain a lower bound on the best objective value under v by taking the most optimistic completion of v, i.e., by turning every "U" into a "O"; no complete valid solution in the subtree rooted at v can yield a lower objective value. Let $I_E^{(sd)}(M)$ be 1 if m_{sd} is "E" in v, and 0 otherwise; $\overline{I_E^{(sd)}(M)}$ is its inverse indicator. We now define the lower bound as the optimistic objective value $\phi(v)$:

$$\phi(v) = \sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} I_E^{(sd)}(v) \tag{6}$$

Whereas $\psi(v)$ represents a valid solution and therefore is an objective value that can actually be attained, $\phi(v)$ in general is not an attainable value. As before, we define $\{\Phi_i\}$ as follows:

$$\Phi_i = \min_{v \in \mathcal{L}_i} \phi(v) \tag{7}$$

Using arguments similar to those made for $\{\Psi_i\}$, we can assert that $\{\Phi_i\}$ is a strong sequence of bounds:

$$\Phi_{i+1} \geq \Phi_i, \quad \forall i \in 1, 2, \cdots, N(N-1) - 1 \tag{8}$$

3) Tightness of Bounds: Consider node $v \in \mathcal{L}_i$. $\phi(v)$ is obtained by setting all the "U" elements of v to "O," while $\psi(v)$ is obtained by setting the same elements to "E." Thus, the difference between these two quantities is equal to the sum of the traffic components corresponding to the mask elements which are set to "U" in v. At level i, the first i traffic components in Π_t have already been used to generate the tree, thus the corresponding mask elements cannot be "U" in v. (The other elements may be "U", but this is not guaranteed because of the pruning methods we follow.) We use $\sigma_i(\Pi_t)$ to denote the sum of the first i elements of $\Pi_t, i = \{1, 2, \cdots, N(N-1)\}$ and let $\sigma_0(\Pi_t) = 0$. Then we have:

$$\psi(v) - \phi(v) \leq \left(\sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} \right) - \sigma_i(\Pi_t) \tag{9}$$

Let v_Φ be the vertex which determines the value of Φ_i, that is, $\phi(v_\Phi) = \Phi_i$. Equation (9) holds for v_Φ, and also $\Psi_i \leq \psi(v_\Phi)$. Combining these observations, we assert that:

$$\Psi_i - \Phi_i \leq \left(\sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} \right) - \sigma_i(\Pi_t) \tag{10}$$

For the fastest convergence, Π_t should be an ordering of the traffic components from the largest to the smallest. Then,

$$\sigma_i(\Pi_t) \geq \frac{i}{N(N-1)} \sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} \tag{11}$$

Thus, the performance of the valid solutions which produce our sequence of upper bounds can be precisely characterized, since we can rewrite the guarantee (10) as:

$$\Psi_i - \Phi_i \leq \left(1 - \frac{i}{N(N-1)} \right) \left(\sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} \right) \tag{12}$$

C. A Greedy Heuristic

The framework of upper bounds allows us to evaluate other heuristic solutions, both as to how closely they approach the optimal, and as to the relative amounts of computation needed. We now describe one greedy heuristic for which we have empirical evidence that it performs quite well (refer to Section V-D).

We start by reducing the traffic matrix as we described in Section V-A. We then create an ordering Π_t of the traffic components in which the traffic components are sorted in order of magnitude, from largest to smallest, with ties broken arbitrarily. The greedy algorithm attempts to assign two-hop lightpaths to traffic components in this order. A traffic component is assigned a lightpath if it is possible without violating the constraints (1 and 2), otherwise it is not. The algorithm terminates when each traffic component has been considered. The complexity of the greedy algorithm is $O(N^4)$.

By combining the framework of upper bounds we propose with the greedy approach above, we can create a scheme which will practically have better performance than the upper bounds themselves, though it is not straightforward to characterize the amount of improvement. As described, the greedy algorithm assumes that no decision regarding electronic or optical routing has been made for any traffic component; this is the equivalent of starting with a mask matrix in which every element is "U." However, it is straightforward to modify the algorithm so that it starts with a matrix in which some elements are "O" or "E." In this case, the algorithm completes the matrix by simply skipping any traffic components which have already been decided (i.e., the corresponding mask element is "O" or "E") when assigning lightpaths. Since the "pessimistic" completion described in Section V-B.1 sets all unassigned mask elements to electronic routing, the greedy completion can do no worse and may possibly do better. Since the greedy completion of the partial solution corresponds to a feasible solution, the value of total electronic switching obtained is still an upper bound on the optimal. Thus, we define a series of upper bounds $\{\Psi_i^{(g)}\}$ based on this idea. As before, we choose to define a new series of upper bounds $\{\Psi_i^g\}$ based on this idea:

$$\Psi_i^{(g)} = \min_{v \in \mathcal{L}_i} \psi^{(g)}(v) \tag{13}$$

where $\psi^{(g)}(v)$ denotes the electronic switching value obtained from the greedy completion of a partial mask matrix v. By a similar argument as used for $\{\Psi_i\}$, $\{\Psi_i^{(g)}\}$ is also a strong sequence of upper bounds.

D. Numerical Results

We characterize a traffic matrix for a star network by two parameters: the *loading factor* and the amount of *hub traffic*. The loading factor is the sum of all the traffic components expressed as a percentage of the total bandwidth available in the network (i.e., the total bandwidth of all the fiber links). For low values of loading, the network is underutilized; such networks are not interesting as it is likely that every traffic component can be given a lightpath. For 100% loading, only traffic components equal to C can be given a lightpath; all other traffic must be electronically switched at the hub. The interesting and most realistic operating condition is when the loading factor is just under 100%, so that opportunities for grooming exist without the problem being trivial; thus, we present results for a loading factor of 90%. The amount of hub traffic is the (average) fraction of the total traffic on each link that is accounted for by traffic to and from the hub. We present results for two values of the hub traffic, 30% and 60%.

Figs. 5 - 7 plot the normalized electronic switching cost of the series of upper and lower bounds against the level i of the search tree. For the results shown in these figures we have used $W = 24$ wavelengths, $C = 16$, and the number of non-hub nodes $N = 6, 10, 20$. Three curves are shown in each figure, one for the series $\{\Phi_i\}$ of lower bounds, one for the series $\{\Psi_i\}$ of upper bounds, and one for the series $\{\Psi_i^{(g)}\}$ of upper bounds (denoted as "Greedy enhanced" in the figures) computed by applying the greedy algorithm to complete a mask matrix rather than the pessimistic completion used for

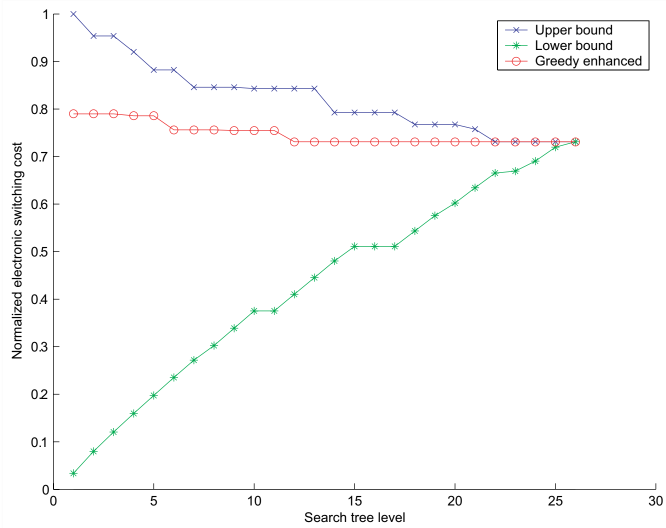

Fig. 5. Star result: $N = 6$, 60% hub traffic

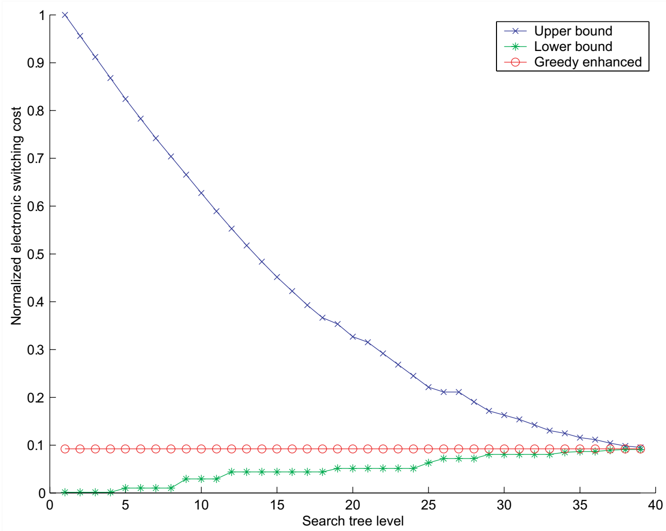

Fig. 6. Star result: $N = 10$, 30% hub traffic

$\{\Psi_i\}$; by definition, this last series will always outperform the original series $\{\Psi_i\}$ of upper bounds. We observe that the sequence of upper and lower bounds do indeed converge to the optimal relatively quickly. For $N = 6, 10$, we were able to reach the optimal within a few minutes of computation on a SUN Sparc-10 workstation; in these cases, the "Greedy enhanced" algorithm also finds the optimal solution, as expected. In fact, the optimal is reached before all levels of the search tree are considered (for $N = 6$ the maximum number of levels is 30, and for $N = 10$ it is 90). For $N = 20$, we terminate the algorithm after examining one million candidate solutions; note that this is a tiny fraction of the 2^{380} possible solutions. For most cases with $N = 20$, the optimal was not reached, and this is demonstrated in Fig. 7, where the lower and upper bounds have not converged. We also observe that the series $\{\Psi_i^{(g)}\}$ of upper bounds outperforms the series $\{\Psi_i\}$. In fact, in all our experiments we have found that the greedy algorithm always performs well.

$$\tau_{ij}^{(p)} = \begin{cases} \sum_{(s,d):s\neq p,d\neq p,t_{sd}\text{traverses links}(q_i,p),(p,q_j)} t_{sd}, & \forall i \neq j \in \{1,\cdots,n\} \\ \sum_{s:s\neq p,t_{sp}\text{traverses link }(q_i,p)} t_{sp}, & \forall i \in \{1,\cdots,n\}, j=0 \\ \sum_{d:d\neq p,t_{pd}\text{traverses link }(p,q_j)} t_{pd}, & \forall j \in \{1,\cdots,n\}, i=0 \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

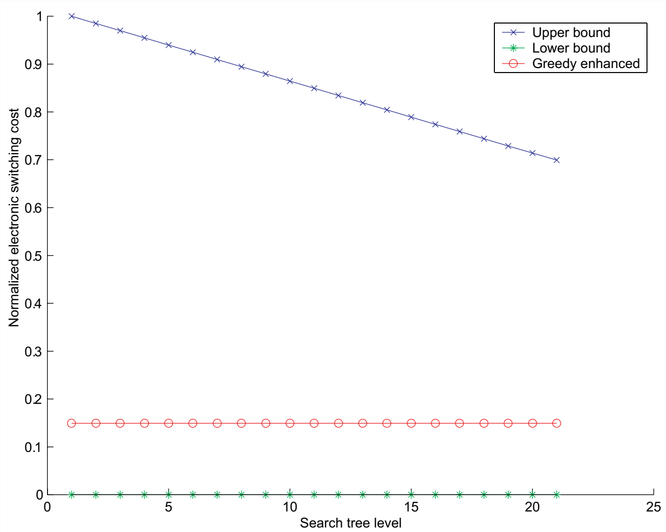

Fig. 7. Star result: $N = 20$, 60% hub traffic

VI. Bounds and Algorithms for Tree Networks

A. Decomposition into Star Networks

The leaf nodes of a tree network do not route traffic either electronically or optically, and we concentrate on the interior nodes. Consider an interior node p of tree $\mathcal{T}$, and the set of nodes $\{q_1, q_2, \cdots, q_n\}$ adjacent to p in $\mathcal{T}$. We define the $(n+1)\times(n+1)$ matrix $T^{(\mathcal{S}_p)} = [\tau_{ij}^{(p)}]$ as in expression (14), shown at the top of the next page. This matrix represents the traffic of the tree $\mathcal{T}$ seen from the point of view of interior node p. Now consider $T^{(\mathcal{S}_p)}$ as the traffic matrix for a star network $\mathcal{S}_p$. The hub node of this star network sees exactly the same traffic scenario as that seen by node p in the tree network, and we refer to $\mathcal{S}_p$ as the "decomposed star network" for node p.

In the star, no node other than the hub does any electronic or optical routing. Thus, the optimal value of electronic switching for the star denotes the optimal (minimum) value of the electronic switching by the hub node of the star only. Since node p is locally in the same traffic scenario as the hub of its decomposed star network, this is the minimum amount of electronic switching that node p can perform in the tree $\mathcal{T}$ under any logical topology and traffic grooming solution. We denote this quantity by $\phi_{\mathcal{T}}(p)$; thus $\phi_{\mathcal{T}}(p)$ is the value of electronic switching that would be obtained by solving the decomposed star $\mathcal{S}_p$ optimally. We also note that $\{\Phi_i\}$ and $\{\Psi_i^{(g)}\}$ for the star $\mathcal{S}_p$ are upper and lower bounds on its optimal electronic routing value and hence on $\phi_{\mathcal{T}}(p)$.

B. Heuristic Based on Star Decomposition

We now show how to obtain a feasible solution to the tree network using the solutions for the star network. We call an interior node of the tree network *opaque* if it switches all traffic electronically. (Conversely, if a node performs optical routing without any restriction other than traffic and wavelength constraints, we call it a *transparent* node.) As with the star, we can create a feasible logical topology in which no node routes any traffic optically. All traffic at all interior nodes is routed electronically, creating a *completely opaque* topology as before. Since this is a feasible topology, the amount of electronic switching performed in this topology is an upper bound on the optimal; in fact, it is the loosest such bound because there is no logical topology in which more electronic switching will need to be performed. Let the amount of electronic switching an interior node p does as an opaque node in the tree be $\psi_{\mathcal{T}}(p)$. Then, the completely opaque upper bound is given by $\Psi_{\mathcal{T}} = \sum_p \psi_{\mathcal{T}}(p)$.

However, realistically we would like to use the optical routing capability of the nodes and create a solution to the tree network in which the amount of electronic switching to be performed is reduced from the maximum at least at some nodes. Recall that $\phi_{\mathcal{T}}(p)$ is the minimum amount of electronic switching node p can do locally. However, to attain this value, the traffic to/from other nodes from/to node p must be groomed according to the optimal solution to $\mathcal{S}^{(p)}$. For two interior nodes p and q which are adjacent, it will not in general be possible to simultaneously attain $\phi_{\mathcal{T}}(p)$ and $\phi_{\mathcal{T}}(q)$ as electronic switching values, because the optimal solutions of the two decomposed stars will in general require the same traffic component in the tree to be groomed differently. For this reason, the lower bound we derived in the last section will in general be unattainable.

To examine what combinations of star decompositions may nevertheless be useful in creating feasible solutions for the tree network, consider a decomposed star network for an interior node p. The hub node corresponds to p, whereas the other nodes of the star correspond to the nodes of the tree that are adjacent to p in $\mathcal{T}$. Some of these nodes may be leaf nodes of the tree, in which case the solution to the decomposed star may be transferred to the tree without any change. However, in general some of the non-hub nodes of the star will be other interior nodes of the tree, and will have their own star decompositions. To create a feasible solution to the tree, we must adopt some method of reconciling the star solutions for adjacent interior nodes of the tree. We now propose one way of doing this.

An opaque node electronically switches all traffic that passes through it. Therefore, traffic components can be rearranged and reassigned to lightpaths arbitrarily at such a node. It is easy to see that the conflict between star solutions to adjacent interior nodes does not arise if the decomposed star for one of the interior nodes is solved optimally while the other one is left as an opaque node. In other words, if we interpose at least one opaque node between every two transparent nodes of

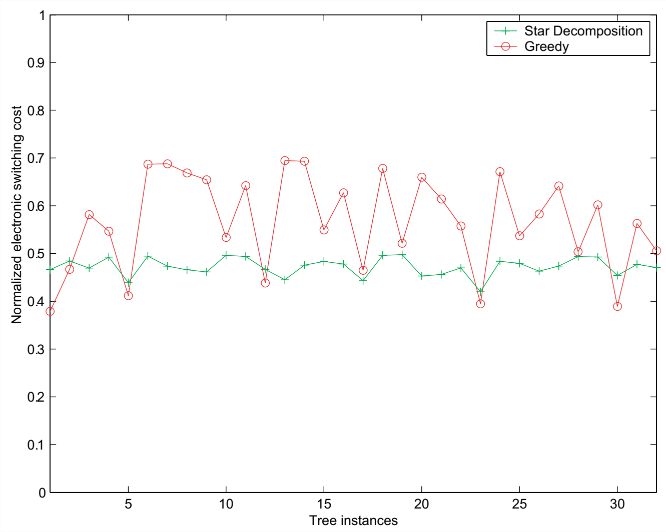

Fig. 8. Tree result: $N = 90 - 150, W = 200, C = 32$

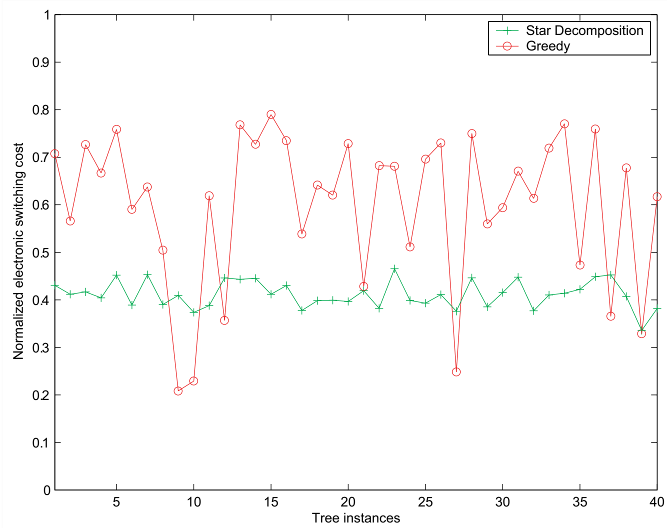

Fig. 9. Tree result: $N = 50 - 150, W = 200, C = 32$

the tree (for which we solve the decomposed star optimally), then there is no problem in combining the corresponding star solutions.

In such a solution, each node p performs either $\phi_{\mathcal{T}}(p)$ amount of electronic switching (the best possible), or $\psi_{\mathcal{T}}(p)$ (the worst). For the best topology which utilizes a combination of transparent and opaque nodes, we would like to choose the nodes such that we get the greatest benefit in terms of electronic switching. Specifically, we would like to find the set of nodes N_t to be designated as transparent nodes, (composed of pairwise non-adjacent interior nodes) such that $\sum_{p \in R}(\psi_{\mathcal{T}}(p) - \phi_{\mathcal{T}}(p))$ is maximized. This problem is equivalent to finding a maximum weight independent set in a tree network, which is solvable in polynomial time using a recursive algorithm. Therefore, using this algorithm, we can obtain the optimal set of nodes N_t, and hence the optimal decomposition of the tree network into stars.

C. A Greedy Heuristic

We now describe a greedy heuristic for tree networks. Because the sequence of feasible solutions we have proposed above get progressively more costly to compute, efficient heuristics would be valuable; and because our solutions never form lightpaths of more than two hops, it is possible that heuristics which are allowed to form longer lightpaths will outperform these solutions in specific cases.

We first list the traffic elements in descending order. The heuristic attempts to optically route the traffic components in this order. In other words, it considers each traffic component in the list and attempts to assign an end-to-end lightpath for it (i.e., one that originates at the source node and terminates at the destination of the component). This attempt may fail for one of two reasons: either there is no free wavelength on the path from source to the destination, or, even if such a wavelength is available, reserving a lightpath for this component would not leave sufficient bandwidth at some intermediate link to accommodate the rest of the traffic that must flow over that link. In the latter case, the algorithm abandons this component, and continues with the next one on the list. If, on the other hand, the failure is due to wavelength availability, a lightpath is formed up to the intermediate node where a continuous wavelength is available, and then the rest of the path is tried similarly. The algorithm terminates when all traffic elements have been examined. Traffic components not assigned to lightpaths because of lack of sufficient bandwidth are carried on single-hop lightpaths from source to destination, undergoing electronic switching at all intermediate nodes.

D. Numerical Results

Figs. 8 and 9 plot the normalized electronic switching cost of the solutions using the star decomposition and the greedy algorithm for thirty problem instances. Each instance in Fig. 8 was generated to have a number N of tree nodes between 90 and 150, $W = 200$, and $C = 32$. Each interior tree node has between 3 to 5 children, thus the number of leaf nodes (likely to represent traffic endpoints) is a large fraction of the total number of nodes. The depth of each tree is at most equal to 4, the fraction of leaf-to-leaf traffic is between 50-60%, and the average link loading is 10%. Traffic demands were randomly generated to follow a distance-dependent traffic pattern such that traffic decreases with the distance between the source and destination nodes (i.e., traffic components were drawn from distributions with means inversely proportional to the path length). Each instance in Fig. 9, on the other hand, has between 50 and 150 nodes, $W = 200$, and $C = 64$; the fraction of leaf-to-leaf traffic is between 80-90%, and the average link loading is 5%.

As we can see, the star decomposition has a normalized electronic switching cost of between 0.3 and 0.5. The greedy heuristic, on the other hand, sometimes performs better than the decomposition, and sometimes worse. It is more opportunistic in taking advantage of the specific nature of a problem instance, whereas the star decomposition provides a more robust guarantee of reasonably good performance. Since the running time of the heuristic is low, it would be reasonable to run both algorithms and select the best solution. We have obtained similar results for a wide range of problem instances, all with the same general nature. More numerical results can be found in [10] by the interested reader.

Overall, the numerical results for all three topologies (paths, stars, and trees), indicate that significant gains in terms of electronic switching costs can be achieved by appropriate traffic grooming.

VII. Concluding Remarks

We considered the traffic grooming problem in WDM path, star, and tree topologies with the objective of minimizing the amount of network-wide electronic switching. We established that a number of variants of the grooming problem are NP-complete for path and stars, and we also showed that it remains so in tree and ring networks even in the presence of wavelength converters. We have obtained lower and upper bounds on the objective function for both star and tree networks, and we have presented a set of heuristics for all three topologies that perform well across a wide range of traffic patterns and loads.

Appendix I
Proof of Theorem 3.2

The reduction is from the Multi-Commodity Flow (MCF) problem in three-stage networks with three nodes in the second stage, which has been proved NP-Complete in [11]. An instance of the problem consists of three sets $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$, of nodes forming the first, second, and third stage, respectively, of a simple staged network (with $|\mathcal{N}_2| = 3$), a set of directed arcs $E \subset (\mathcal{N}_1 \times \mathcal{N}_2) \cup (\mathcal{N}_2 \times \mathcal{N}_3)$, each of unit capacity, and a set of flow requirements $Q \subset (\mathcal{N}_1 \times \mathcal{N}_3)$, each of unit magnitude. The question is whether a feasible flow assignment satisfying the flow requirements exists.

We construct a path network with as many nodes as the three stage network, with a one-to-one correspondence between the nodes of the staged network and those of the path, as illustrated in Fig. 10. We define the following quantities. Let A be the set of all ordered node pairs (i, j) of the path network such that $(s, d) \in E$ for the staged network, where i is the path node corresponding to s and j is the path node corresponding to d. Similarly, let B be the set of all ordered path node pairs (i, j) of the path network such that the corresponding pair $(s, d) \in Q$ for the staged network. For the link from path node i to $i+1, i \in \{1, 2, \cdots, N-1\}$), let $w_i = |\{(l, k) : (l, k) \in A, l \leq i, k \geq i+1\}|$. That is, w_i is the number of arcs that would cross the link from i to $i+1$ if the arcs of the staged network were drawn between corresponding nodes of the path network. We construct a path network using the following transformation: $N = |\mathcal{N}_1| + 3 + |\mathcal{N}_3|$, $C = |Q| + 2$, $W = \max_i\{w_i\}$, $F = |Q|$, and traffic matrix:

$$t_{ij} = \begin{cases} C-1, & (i,j) \in A \\ 1, & (i,j) \in B \\ (W - w_i)C, & i \in \{1, 2, \cdots, N-1\}, j = i+1 \\ 0, & \text{otherwise} \end{cases}$$

That is, each arc of the staged network generates a traffic component of magnitude $C - 1 (= |Q| + 1)$, and each flow requirement of the staged network generates a traffic component of magnitude 1, between the corresponding nodes of the path network.

Because the magnitude of the traffic components corresponding to the arcs of the staged network are each $|Q|+1$, the goal cannot be achieved if even one of these traffic components is completely electronically switched. Thus, at least one unit of traffic for such a traffic component must be optically routed, and this is true of every such traffic component. Hence, for the goal to be achieved, the logical topology must include at least one direct lightpath for each of these traffic components. That is, any logical topology satisfying the goal must include at least one lightpath with source node i and destination j for each node pair $(i, j) \in A$. However, exactly one such lightpath for each $(i, j) \in A$ can be formed, since, together with the $W - w_i$ single-hop lightpaths that must be formed over each link from node i to $i+1$ to carry the single-hop traffic, they will occupy every wavelength on every link. In other words, a complete logical topology is forced. On this logical topology, the single-hop lightpaths are completely occupied with the single-hop traffic. Each of the lightpaths $(i, j) \in A$ must carry the entire traffic from i to j, since there are only lightpaths from one "stage" to another in the path, and so are the traffic components. The remaining bandwidth and remaining traffic components are exactly the ones corresponding to the arc capacities and flow requirements, respectively, of the MCF 3-stage problem instance. Every possible path for every traffic component involves exactly one intermediate node from source to destination, thus if it is at all feasible to route the traffic, the electronic switching goal will be met. Therefore, the path network grooming problem instance is satisfiable *iff* the MCF problem instance is. Since the MCF problem is known to be NP-Complete, so is the path grooming problem. ■

Appendix II
Proof of Theorem 3.3

The reduction is from the same Multi-Commodity Flow used in the proof of Theorem 3.2.

Suppose that we have a polynomial time approximation algorithm M which has approximation ratio $R_M(I) \leq \infty$ for the traffic grooming problem instance I with $OPT(I) > 0$, where $OPT(I)$ is the optimal value. (Excluding the cases where $OPT(I) = 0$ does not change the intractability of the problem, since those cases are trivially solvable.) It implies that there would be a polynomial time algorithm M satisfying $R_M(I) = \frac{M(I)}{OPT(I)} \leq K$, where, $M(I)$ is the result returned by the algorithm M, for some positive integer $K > 1$. Then construct the instance I as follows: For any given instance I^{MCF} of the MCF problem, we add $K|Q|$ dummy nodes to $\mathcal{N}_2$. We name them as D=$\{D_1, \ldots D_{K|Q|}\}$. First we define the following sets: A and B are sets of node pairs exactly as before. Let H be the set of all ordered node pairs (s, d) of the path network such that either ($s \in \mathcal{N}_1$ and $d = D_1$) or ($s = D_{K|Q|}$ and $d \in \mathcal{N}_3$). Let L be the set of all ordered node pairs (s, d) of the path network such that $s = D_i, d = D_{i+1}, \forall i \in 1, 2, \ldots, K|Q| - 1$.

Then we construct a path network with as many nodes as the three stage network, exactly as in the proof of Theorem 3.2, with the following additions.

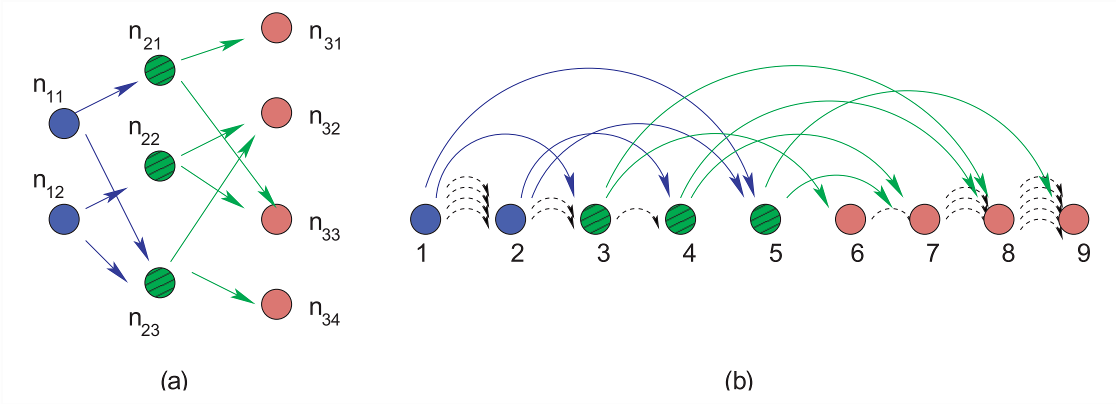

Fig. 10. Example of path construction for the proof of Theorem 3.2: (a) 3-stage network instance, (b) corresponding path instance together with the forced logical topology, $W = 6$

For the link from node i to $i+1$ ($i \in \{1, 2, \ldots, \ldots N-1\}$), let $w_i = |\{(s,d) : (s,d) \in A \cup H \cup L, s \leq i, d \geq i+1\}|$. Construct a path network using the following transformation: $N = |\mathcal{N}_1|+3+K|Q|+|\mathcal{N}_3|$, $C = K|Q|+2$, $W = \max_i\{w_i\}$,

$$t^{(sd)} = \begin{cases} C-1, & (s,d) \in A \cup H \\ 1, & (s,d) \in B \\ (W - w_i + 1)C - |Q|, & (s,d) \in L \\ (W - w_i)C, & d = s+1, (s,d) \notin L \end{cases}$$

All other traffic components are zero, and $F = |Q|$. Since K is independent of I, this construction is in polynomial time. Since the traffic $t^{(sd)} = 1, (s,d) \in B$, can always be routed as $\{s, D_1, \ldots, D_{K|Q|}, d\}$, $M(I)$ will always return a feasible solution.

If $M(I) \leq K|Q|$, it implies that none of the traffic components of magnitude $K|Q|+1$ is completely electronically routed. Thus at least one unit of traffic for such a traffic component must be optically routed, and this is true of every such traffic component. Hence the virtual topology must include at least one direct lightpath for each of these traffic components. That is, any virtual topology satisfying the goal must include at least one lightpath with source node s and d for each node pair $(s,d) \in A \cup H \cup L$. However, exactly one such, together with the $W - w_i$ single-hop lightpaths that must be formed over each link from node i to $i+1$ to carry the single-hop traffic, will occupy every wavelength on every link; thus a complete virtual topology is forced.

Furthermore, if even one traffic component in B is routed through the route $\{s, D_1, \ldots, D_{K|Q|}, d\}$, it will introduce exactly an amount of electronic routing $K|Q|$. Since $|Q| \geq 2$ (again we exclude only trivial cases), we have $K|Q| > 2$. Therefore the amount of electronic routing is at least $K|Q| + |Q| - 1$, which is larger than $K|Q|$. Hence, $M(I) \leq K|Q|$ implies that no traffic component in B is routed by the route $\{s, D_1, \ldots, D_{K|Q|}, d\}$, which, in turn, implies that the instance I^{MCF} is satisfiable.

On the other hand, if $M(I) > K|Q|$, since $\frac{M(I)}{OPT(I)} \leq K$, then $|Q| < OPT(I)$, i.e., the instance of I^{MCF} is not satisfiable. Thus, using the algorithm M, we can solve the MCF problem in polynomial time. Obviously, if we assume that $P \neq NP$, then M cannot exist. ■

Appendix III
Proof of Theorem 3.4

Recalling that the problem consists of deciding which traffic elements are routed optically and which are not, we recognize that requiring the electronic switching at the hub to be F or less is equivalent to requiring that the optical routing at the hub be Q or more, where $Q = \sum_{i,j=1}^{N} t_{ij} - F$. In what follows, we use Q rather than F for notational convenience.

We reduce the decision version of the Knapsack problem [12] to the grooming problem. An instance of the Knapsack problem is given by a finite set U of cardinality n, for each element $u_i \in U$ a weight $w_i \in Z^+$, and a value $v_i \in Z^+, \forall i \in \{1, 2, \cdots, n\}$, a target weight $B \in Z^+$, and a target value $K \in Z^+$. The problem asks whether there exists a binary vector $X = \{x_1, x_2, \cdots, x_n\}$ such that $\sum_{i=1}^{n} x_i w_i \leq B$, and $\sum_{i=1}^{n} x_i v_i \geq K$. Given such an instance, we construct a star network using the following transformation: $N = n+2$, $W = n$, $C = \max_i(w_i+v_i)+1$, $Q = K + \sum_{i=1}^{n}(C - w_i - v_i)$, and traffic matrix:

$$t_{ij} = \begin{cases} C - w_j, & i = n+1, j = 1, 2, \cdots, n \\ C - w_j - v_j, & i = n+2, j = 1, 2, \cdots, n \\ (n-2)C + w_j, & i = 0, j = 1, 2, \cdots n, \\ \sum_{k=1}^{n} w_k - B, & i = n+1, j = 0 \\ 0, & \text{otherwise} \end{cases}$$

In the resulting star network, the only traffic components switched through the hub optically or electronically are those from one of the *source nodes* $n+1$ and $n+2$ to one of the *destination nodes* $1, 2, \cdots, n$. The amount of traffic of each such component is less than the capacity of a wavelength. There is also traffic from the hub node to each destination node, and traffic from source node $n+1$ to the hub. Due to the traffic from the hub, any one, but not both, of the traffic components from the source nodes may be optically routed for each destination node. Not all traffic sourced by source node $n+1$ may be optically routable, due to the traffic to the hub, which requires terminating some lightpaths at the hub. There is no such restriction for source node $n+2$, so a lightpath may be formed from it to every destination node which does not sink a lightpath from source node $n+1$. Therefore, we need only to consider candidate solutions in which there is a lightpath from exactly one of nodes $n+1$, $n+2$, to each node $i \in \{1, 2, \cdots, n\}$ to determine the satisfiability of the instance.

Let X denote a candidate solution of the Knapsack instance. Consider the solution of the star network in which X (respectively, $\overline{X}$) represents the indicator vector of the lightpaths formed from node $n+1$ (resp., $n+2$). Applying the transformation to the satisfiability criteria of Knapsack, we obtain:

$$
\begin{aligned}
& \sum_{i=1}^{n} x_i w_i \le B \\
\Rightarrow\ & \sum_{i=1}^{n} x_i (C - t_{n+1,i}) \le \sum_{i=1}^{n} (C - t_{n+1,i}) - t_{n+1,0} \\
\Rightarrow\ & \sum_{i=1}^{n} (\overline{x_i} t_{n+1,i}) + t_{n+1,0} \le (n - \sum_{i=1}^{n} x_i) C && (15) \\
& \sum_{i=1}^{n} x_i v_i \ge K \\
\Rightarrow\ & \sum_{i=1}^{n} x_i (t_{n+1,i} - t_{n+2,i}) \ge Q - \sum_{i=1}^{n} t_{n+2,i} \\
\Rightarrow\ & \sum_{i=1}^{n} (x_i t_{n+1,i} + \overline{x_i} t_{n+2,i}) \ge Q && (16)
\end{aligned}
$$

Thus, the weight constraint translates to the requirement that the lightpaths from source node $n+1$ to the hub can carry the hub traffic as well as all traffic components which have not been given a lightpath, i.e., the logical topology is feasible. The value criterion translates to the requirement regarding the minimum amount of optical routing. Therefore, a given vector X either satisfies both the Knapsack and the grooming instance, or fails to satisfy both. Hence, the grooming instance is satisfiable *iff* the Knapsack instance is. ∎

REFERENCES

[1] J-C. Bermond and D. Coudert, "Traffic grooming in unidrirectional WDM ring networks: the all-to-all unitary case," in *Proc. 7th IFIP Working Conference on Optical Network Design and Modeling*, Feb. 2003.

[2] R. Berry and E. Modiano, "Reducing electronic multiplexing costs in SONET/WDM rings with dynamically changing traffic," *IEEE J. Select. Areas Commun.*, vol. 18, no. 10, pp. 1961–1971, Oct. 2000.

[3] B. Chen, G. N. Rouskas, and R. Dutta, "A framework for hierarchical grooming in WDM networks of general topology," in *Proc. BROADNETS 2005*, to appear.

[4] A. L. Chiu and E. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *IEEE/OSA J. Lightwave Technol.*, vol. 18, no. 1, pp. 2–12, Jan. 2000.

[5] S. Corteel, M. Valencia-Pabon, D. Gardy, D. Barth, and D. Denise, "The permutation-path coloring problem on trees," *Theoretical Computer Science*, vol. 297, pp. 119–143, 2003.

[6] R. Dutta, S. Huang, and G. N. Rouskas, "Traffic grooming in path, star, and tree networks: complexity, bounds, and algorithms," in *Proc. ACM SIGMETRICS*, pp. 298–299, June 2003.

[7] R. Dutta and G. N. Rouskas, "A survey of virtual topology design algorithms for wavelength routed optical networks," *Optical Networks*, vol. 1, no. 1, pp. 73–89, Jan. 2000.

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

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

[10] Rudra Dutta, "Virtual topology design for traffic grooming in WDM networks," PhD thesis, North Carolina State University, Raleigh, NC, Aug. 2001.

[11] E. S. Elmallah and J. Culberson, "Multicommodity flows in simple multistage networks," *Networks*, vol. 25, pp. 19–30, 1995.

[12] M. R. Garey and D. S. Johnson, *Computers and Intractability*. New York: W. H. Freeman and Co., 1979.

[13] O. Gerstel, P. Lin, and G. Sasaki, "Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings," in *Proc. IEEE INFOCOM 1998*, pp. 94–101.

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

[15] O. Goldschmidt, D. S. Hochbaum, A. Levin, and E. V. Olinick, "The SONET edge-partition problem," *Networks*, vol. 41, no. 1, pp. 13–23, Jan. 2003.

[16] U. Gupta, D. Lee, and J. Leung, "Efficient algorithms for interval graphs and circular-arc graphs," *Networks*, vol. 12, pp. 459–467, 1982.

[17] E. Harder, S.-K. Lee, and H.-A. Choi, "On wavelength assignment in WDM optical networks," in *Proc. 4th International Conference on Massively Parallel Processing Using Optical Interconnections*, April 1997.

[18] S. Huang, "Traffic grooming in wavelength routed path networks," Master's thesis, North Carolina State University, 2002.

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

[20] G. Kramer and G. Pesavento, "Ethernet passive optical network (EPON): Building a next-generation optical access network," *IEEE Commun. Mag.*, vol. 40, no. 2, pp. 66–73, Feb. 2002.

[21] M. Kuznetsov *et al.*, "A next-generation optical regional access network," *IEEE Commun. Mag.*, vol. 38, no. 1, pp. 66–72, Jan. 2000.

[22] G. Li. and R. Simha, "Efficient routing to reduce the cost of add-drop multiplexers in WDM optical ring networks," in *Proc. SPIE Conference 1999*, vol. 3843, pp. 258–267.

[23] E. Modiano and P. J. Lin, "Traffic grooming in WDM networks," *IEEE Commun. Mag.*, vol. 39, no. 7, pp. 124–129, July 2001.

[24] G. Sasaki and T. Lin, "A minimal cost WDM network for incremental traffic," in *Proc. SPIE 1999*, vol. 3843.

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

[26] D. West, *Introduction to Graph Theory*. Englewood Cliffs, NJ: Prentice Hall, 1996.

[27] G. Wilfong and P. Winkler, "Ring routing and wavelength translation," in *Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms 1998*, pp. 333–341.

[28] X. Zhang and C. Qiao, "On scheduling all-to-all personalized connections and cost-effective designs in WDM rings," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 435–445, June 1999.

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

Shu Huang received the B.E. degree in electrical engineering from Beijing University of Posts and Telecommunications, Beijing, China, in 1995. He spend the next several years in China Academy of Telecommunictions as an engineer. He received the M.S. degree in Computer Networking in 2002 from North Carolina State University, and went back for a short spell in the industry. Since Fall 2003 he has been working towards his Ph.D degree in Computer Science at North Carolina State University in Raleigh, NC, USA.

Shu's research interests focus on the application of optimization and graph theoretic methods on network design; in particular, optical and wireless multihop networks.

Rudra Dutta (S '00, M '01) was born in Kolkata, India, in 1968. After completing elementary schooling in Kolkata, he received a B.E. in Electricial Engineering from Jadavpur University, Kolkata, India, in 1991, a M.E. in Systems Science and Automation from Indian Institute of Science, Bangalore, India in 1993, and a Ph.D. in Computer Science from North Carolina State University, Raleigh, USA, in 2001.

From 1993 to 1997 he worked for IBM as a software developer and programmer in various networking related projects. He is currently employed as Assistant Professor in the department of Computer Science at the North Carolina State University, Raleigh. During the summer of 2005, he was a visiting researcher at the IBM WebSphere Technology Institute in RTP, NC, USA. 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, including the one which supported in part this paper. He has served as a reviewer for many premium journals and was part of the organizing committee of BroadNets 2005, and Program Co-chair for the Second International Workshop on Traffic Grooming.

Rudra is married with two children and lives in Cary, North Carolina with his family. His parents and his sister's family live in Kolkata, India.

George N. Rouskas (S '92, M '95, SM '01) is a Professor of Computer Science at North Carolina State University. He 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 College of Computing, Georgia Institute of Technology, Atlanta, GA, in 1991 and 1994, respectively. During the 2000-2001 academic year he spent a sabbatical term at Vitesse Semiconductor, Morrisville, NC, and in May 2000 and December 2002 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, a co-author of a paper that received the Best Paper Award at the 1998 SPIE conference on All-Optical Networking, and the recipient of the 1994 Graduate Research Assistant Award from the College of Computing, Georgia Tech. Dr. Rouskas is especially proud of his teaching awards, including his induction in the NCSU Academy of Outstanding Teachers in 2004, and the Outstanding New Teacher Award he received from the Department of Computer Science in 1995.

Dr. Rouskas 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, published in October, 2000. He was Technical Program Co-Chair of the Networking 2004 conference, Program Chair of the IEEE LANMAN 2004 workshop, and Program Co-Chair of the Traffic Grooming workshop 2004. He also serves as General Co-Chair of the IEEE LANMAN 2005 workshop. He is a member of the ACM and of the Technical Chamber of Greece.