On Optimal Traffic Grooming in Elemental Network Topologies

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

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 problem, 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 settle the complexity of traffic grooming in path and star networks by proving that a number of variants of the problem are computationally hard. 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 form 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.

1. 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 congestion (maximum traffic flowing over any link), or a combination of the two.

With the deployment of commercial WDM systems, it has become apparent that the cost of network components, especially *line terminating equipment* (LTE), 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* [1], 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.

Traffic grooming has enjoyed significant attention in the literature recently. Due to lack of space, we have not included a full bibliography in this paper; we refer the reader to our recent survey work of the area [1] and references thereof. 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. 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, 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

* {dutta,shuang5,rouskas}@csc.ncsu.edu

right: star networks arise in the interconnection of LANs or MANs with a wide area backbone, while passive optical networks (PONs) 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 [2], where an algorithm with strong guarantees for ring networks is based on a decomposition into path segments. 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 problem.

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 hard. 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 2 we present the main complexity results. In Sections 3, 4, and 5, 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 6.

2. COMPLEXITY RESULTS

In this section we present our results regarding the computational complexity of the traffic grooming problem. 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 a 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). This problem has been described previously in the literature. In particular, [1] provides a complete formulation of the problem as an integer linear program, and also discusses the *routing and wavelength assignment* (RWA) problem, with a review of known results.

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 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.) 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 lightpaths,
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 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.

2.1. 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 $T = [t_{ij}], 1 \leq i < j \leq N$, 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.

We first consider the case where bifurcated routing of traffic is not allowed. Specifically, for any source-destination pair (i, j) such that $t_{ij} \leq C$, 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 than the one in [3] (which does not allow traffic to be transferred from one wavelength to another), and uses a reduction from a different NP-Complete problem. As we shall see shortly, our proof provides insight into the inherent difficulty of the grooming problem.

Theorem 2.1. *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. 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 given by (1).

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

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, may be considered forced on us by the construction; this topology is shown in Figure 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 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 2.1. *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.*

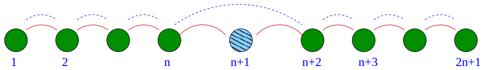

Figure 1. Example of path construction for the proof of Theorem 2.1, $N = 2n+1$, $W = 2$

Note: Because of the construction in the proof of Theorem 2.1, 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 2.2. *The decision version of the traffic grooming problem in unidirectional path networks (bifurcated routing of traffic allowed) is NP-Complete.*

The proof of this theorem is rather long and we have not included it here for the sake of brevity. It is based on a reduction from the Multi-Commodity Flow (MCF) problem in three-stage networks with three nodes in the second stage, which has been proved NP-Complete. The proof can be found in [4]. Again, by the nature of the proof, we are able to state the following:

Corollary 2.2. *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.*

Considering that the goal of the above problem 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 in the next theorem that this is not true.

Theorem 2.3. *Approximating the optimization version of the unidirectional path network grooming problem (bifurcated routing of traffic allowed) is NP-hard, unless P=NP.*

The proof is very similar to the one for Theorem 2.2, and also involves a reduction from the 3-stage MCF problem. It is also available in [4], and is omitted here.

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 cosider 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 optimal algorithm for ring network grooming for some values is given in [5], and other works provide approximation results. 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 [3]. In addition, the study in [5] addresses only the all-to-all unitary traffic pattern.

Our results on the complexity of traffic grooming in path networks have implications for bidirectional path and ring networks as well. In particular, it is known that the RWA problem in rings is NP-hard. 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.

Lemma 2.4. *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 2.5. *The decision version of the traffic grooming problem in unidirectional and 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.*

The proofs are straightforward and are in each case based on an appropriate restriction of the traffic matrix. They are omitted for lack of space, and can be found in [4].

2.2. 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. Hence, requiring that the electronic switching at the hub 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 the proof below, we use Q rather than F for notational convenience.

Theorem 2.6. *The decision version of traffic grooming in star networks is NP-Complete.*

Proof. We reduce the decision version of the Knapsack problem 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 $\{t_{ij}\}$ given by (2).

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

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 consider 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 (3) and (4). 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 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. Note that the proof works whether bifurcation is allowed or not. ■

$$\begin{aligned} & \sum_{i=1}^{n} x_i w_i \leq B \\ \Rightarrow\ & \sum_{i=1}^{n} x_i (C - t_{n+1,i}) \leq \sum_{i=1}^{n} (C - t_{n+1,i}) - t_{n+1,0} \\ \Rightarrow\ & \sum_{i=1}^{n} (\overline{x_i} t_{n+1,i}) + t_{n+1,0} \leq (n - \sum_{i=1}^{n} x_i) C \end{aligned} \tag{3}$$

$$\begin{aligned} & \sum_{i=1}^{n} x_i v_i \geq K \\ \Rightarrow\ & \sum_{i=1}^{n} x_i (t_{n+1,i} - t_{n+2,i}) \geq Q - \sum_{i=1}^{n} t_{n+2,i} \\ \Rightarrow\ & \sum_{i=1}^{n} (x_i t_{n+1,i} + \overline{x_i} t_{n+2,i}) \geq Q \end{aligned} \tag{4}$$

2.3. Tree Networks

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

Corollary 2.3. *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 follows from Theorem 2.6, since the tree can be restricted to the star, which is now known to be NPC. Since wavelength assignment is always possible for the star, wavelength conversion does not help. ■

3. 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 is a special case of a ring network. In [2], 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 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 [2].

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 [2]) 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.

3.1. A Simple Heuristic

We now present a simple 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. Therefore this process does not make the problem inherently easier or more difficult. 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 multi-hop non-zero 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 (since 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 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 $m+1$ are 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 $m-1$ or 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, 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.

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 omitted due to lack of space. 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.

3.2. Numerical Results

The performance measure of interest in our study is the *grooming effectiveness* of a solution. The grooming effectiveness 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.

Figures 3-4 plot the grooming effectiveness 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 Figure 3 were randomly generated such that the traffic pattern is distance-independent (i.e., uniform pattern), while those in Figure 4 were randomly generated to follow a distance-dependent traffic pattern such that traffic demands increase with the distance between source and destination nodes.

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 grooming effectiveness around 0.1-0.15, well below the upper bound. Through extensive experimentation, we have found that the difference in the grooming effectiveness 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 grooming effectiveness 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 Figures 3-4 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.

4. 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 completion of this partial solution.

We now present an algorithm which builds solutions incrementally, visiting each possible solution exactly once while avoiding any invalid solution. We demonstrate how the incremental nature of the algorithm allows increasingly good bounds, both upper and lower, to be obtained with successively more computation.

4.1. Examining Solutions

Consider an $N \times N$ *mask* matrix $M = [m_{ij}]$, the rows and columns of which correspond to the rows and columns of the traffic matrix T other than the row (resp., column) that includes traffic components originating (resp., terminating) at the hub. Every element (other than the diagonals) is either "E" (electronic switching), "O (optical switching)", or "U" (unassigned), thus a mask matrix represents a complete or partial proposed solution to the instance. There are no "U" elements in complete solutions. We traverse a binary search tree in which each vertex is a mask matrix, such as the one shown in Figure 2, starting with the completely unassigned one.

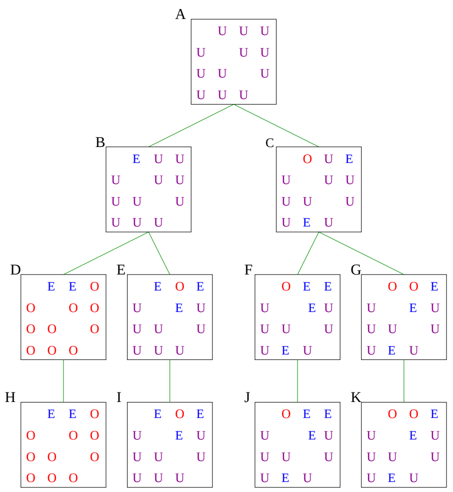

Figure 2. Generating the search tree for a star network.

All vertices other than leaf vertices represent partial mask matrices, whereas leaf vertices are complete mask matrices. We generate level $i+1$ of the tree by selecting a *single* mask element and generating children for each vertex at level i, one each setting the element to "E" and "O" if it was "U" in the level i vertex, a single child identical to the level i vertex otherwise. 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. Then Π_t determines the order in which partial solutions will be generated, and thus characterises the search, but every possible mask matrix will be generated when the full tree of depth $N(N-1)$ is generated.

4.1.1. Pruning

We prune the search in several ways, only one is described here. 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 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 (5, 6) must be satisfied for any valid mask matrix M. In the above expressions, $I_O^{(sd)}(M)$ 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 3.1. The term in parentheses in the right-hand side of (5) represents the number of wavelengths that are available at non-hub node i to source traffic (the term $W_o(i)$) after creating the two-hop lightpaths already specified in matrix M (the sum within the parentheses). The left-hand side of (5) is the traffic demand out of node i that has not been assigned to a two-hop lightpath in matrix M, which must now be carried to the hub; hence inequality. (6) is the same constraint with respect to traffic to (rather than from) node i.

As we generate the search tree, we can avoid generating the invalid cases by applying (5) and (6). 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.) That is, 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. This is illustrated by the generation of G from C in Figure 2. In doing this, 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 it results in no corresponding pruning.

$$\left(\sum_{d \neq i} \overline{I_O^{(id)}(M)} t_{id}\right) + t_{i0} \leq \left(W_o(i) - \sum_{d \neq i} I_O^{(id)}(M)\right) C \qquad (5)$$

$$\left(\sum_{s \neq i} \overline{I_O^{(si)}(M)} t_{si}\right) + t_{0i} \leq \left(W_t(i) - \sum_{s \neq i} I_O^{(si)}(M)\right) C \qquad (6)$$

The other types of pruning we perform are on suboptimal solutions which can be easily replaced with another solution which is closer to optimal. One of the ways to do this is problem-specific and the other is similar to branch-and-bound techniques. They are fully described in [6].

4.2. Bounds

While the pruning cannot affect the essential complexity of the problem, our algorithm can be used incrementally to obtain bounds on the optimal with less computation.

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 $\psi(v)$, defined by (7). 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 in (8). Let u_E and u_O be the two children of v, a little reasoning (see [6]) convinces us that $\min\{\psi(u_E), \psi(u_O)\} \leq \psi(v)$. Hence $\{\Psi_i\}$ has the desirable property of being a strong sequence of bounds: $\Psi_{i+1} \leq \Psi_i, \quad \forall i \in 1, 2, \cdots, N(N-1)-1$.

$$\psi(v) = \sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} \overline{I_O^{(sd)}(v)} \quad (7)$$

$$\Psi_i = \min_{v \in \mathcal{L}_i} \psi(v) \quad (8)$$

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 $\phi(v)$ as in (9), defining $\{\Phi_i\}$ as before. 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. Using arguments similar to those made for $\{\Psi_i\}$, we can assert that $\{\Phi_i\}$ is a strong sequence of bounds: $\Phi_{i+1} \geq \Phi_i, \quad \forall i \in 1, 2, \cdots, N(N-1)-1$.

$$\phi(v) = \sum_{s,d \in \{1,\cdots,N\}, s \neq d} t_{sd} I_E^{(sd)}(v) \quad (9)$$

$$\Phi_i = \min_{v \in \mathcal{L}_i} \phi(v) \quad (10)$$

Tightness of Bounds: We can show that as i increases, Ψ_i and Φ_i converge at a rate that can be guaranteed, by properly choosing Π_t. The details can be found in [6].

4.3. A Greedy Heuristic

We describe a greedy heuristic with empirical evidence of good performance. After reduction, the traffic components of the traffic matrix are considered in decreasing order of magnitude, and assigned lightpaths if (5) and (6) allow. This scheme has the desirable characteristic that it can be applied to any partially generated search tree also, because any partial mask has a "greedy completion". Thus, we define a series of upper bounds $\{\Psi_i^{(g)}\}$ based on this idea: $\Psi_i^{(g)} = \min_{v \in \mathcal{L}_i} \psi^{(g)}(v)$, where $\psi^{(g)}(v)$ denotes the objective value obtained from the greedy completion of a partial mask matrix v. Clearly, $\{\Psi_i^{(g)}\}$ is also a strong sequence of upper bounds.

4.4. 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. For low values of loading, the network is underutilized; for 100% loading, only traffic components equal to C can be given a lightpath. 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%.

Figures 5-7 plot the grooming effectiveness 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 $W = 24$, $C = 16$, and $N = 8, 10, 20$. 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). We observe that the sequence of upper and lower bounds do indeed converge to the optimal relatively quickly. For $N = 8, 10$, we were able to reach the optimal within a few minutes of computation on a SUN Sparc-10 workstation. In fact, the optimal is reached before all levels of the search tree are considered (for $N = 8$ the maximum number of levels is 56, 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, the optimal was not reached, as in Figure 7, where the bounds have not converged. We also observe that the series $\{\Psi_i^{(g)}\}$ of upper bounds outperforms the series $\{\Psi_i\}$. In fact, we have consistently found that the greedy algorithm performs well.

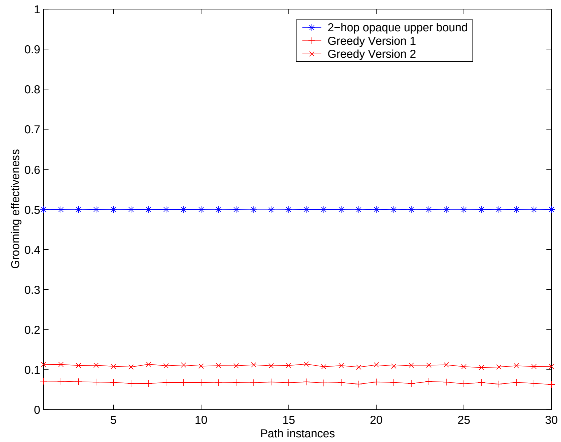

Figure 3. Path result: $N = 30, W = 80, C = 32$, uniform pattern

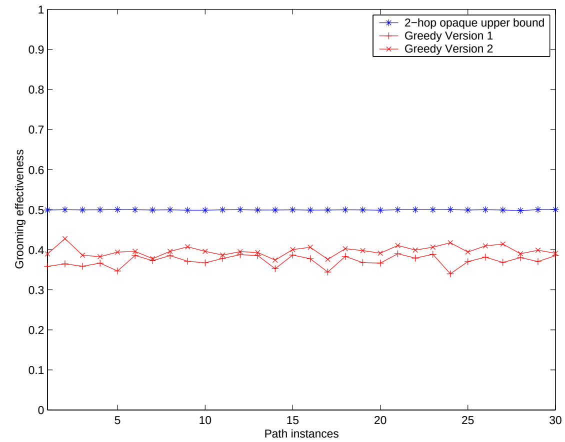

Figure 4. Path result: $N = 30, W = 10, C = 128$, rising pattern

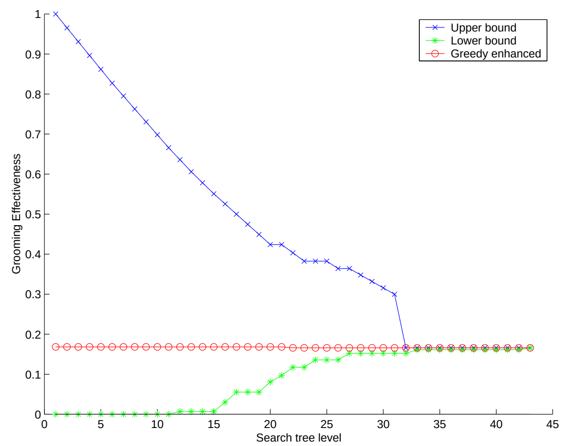

Figure 5. Star result: $N = 8, W = 24, C = 16$, 60% hub traffic

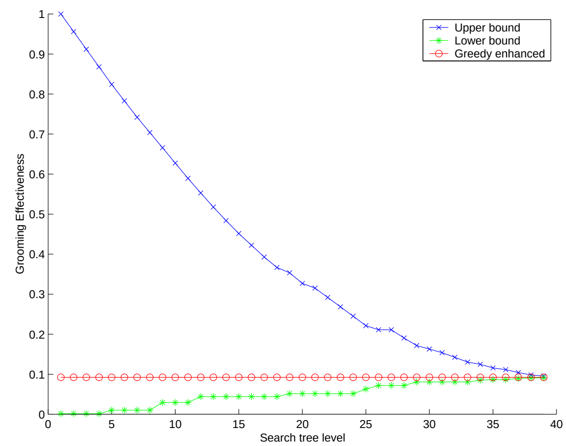

Figure 6. Star result: $N = 10, W = 24, C = 16$, 30% hub traffic

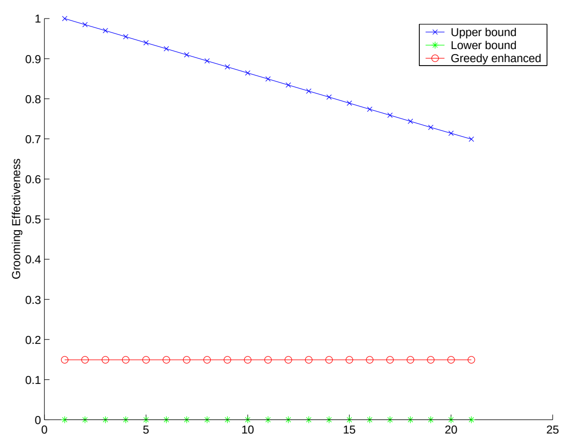

Figure 7. Star result: $N = 20, W = 24, C = 16$, 60% hub traffic

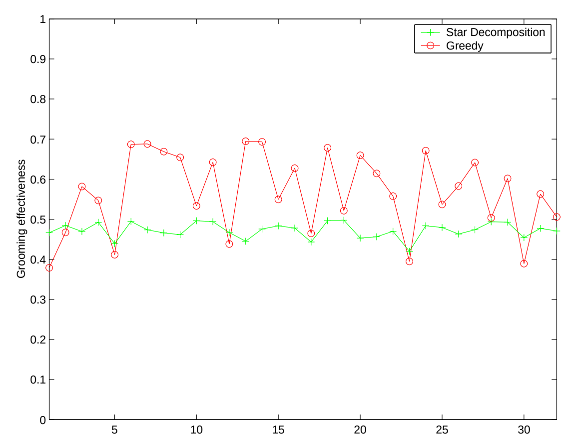

Figure 8. Tree result: $N = 90 - 150, W = 200, C = 32$

5. BOUNDS & ALGORITHMS FOR TREE NETWORKS

5.1. 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 $T^{(\mathcal{S}_p)} = [\tau_{ij}^{(p)}]$ by (11). This matrix represents the traffic of the tree $\mathcal{T}$ seen from the point of view of interior node p. The term $\tau_{ij}^{(p)}$ for this decomposed star corresponds to the term t_{ij} for an actual star network. There is one such decomposed star network for every interior node of the tree. 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.

$$\boxed{\tau_{ij}^{(p)} = \begin{cases} \sum t_{sd} \text{ over } (s,d) : s \neq p, d \neq p, t_{sd} \text{ traverses} \\ \quad \text{links } (q_i, p), (p, q_j), \forall i \neq j \in \{1, \cdots, n\} \\ \sum t_{sp} \text{over } s : s \neq p, t_{sp} \text{traverses the} \\ \quad \text{link } (q_i, p), \forall i \in \{1, \cdots, n\}, j = 0 \\ \sum t_{pd} \text{ over } d : d \neq p, t_{pd} \text{ traverses the link} \\ \quad (p, q_j), \forall j \in \{1, \cdots, n\}, i = 0 \\ 0, \quad \text{otherwise} \end{cases} \qquad (11)}$$

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 $\phi_{\mathcal{T}}(p)$.

5.2. Bounds

Since the amount of electronic switching performed by the different interior nodes of the tree are disjoint quantities, the quantity $\sum_p \phi_{\mathcal{T}}(p)$ is a lower bound on the amount of total electronic switching performed in any logical topology, and hence on the optimal value of electronic switching in the tree $\mathcal{T}$. Further, if obtaining optimal values is not computationally practical, lower bounds can be used in the same way. Let us denote the actual lower bound used on $\phi_{\mathcal{T}}(p)$ by $\phi_{\mathcal{T}}^{(i)}(p)$. This quantity could be chosen out of the quantities $\{\Phi_i\}$ for the decomposed star network of the node p. The closeness with which we approach $\phi_{\mathcal{T}}(p)$ may be different for every p. Now the quantity $\sum_p \phi_{\mathcal{T}}^{(i)}(p)$ is still a lower bound on the optimal electronic routing for tree $\mathcal{T}$, though a weaker one. However, such a bound requires much less computation to determine. Thus, $\{\sum_p \phi_{\mathcal{T}}^{(i)}(p)\}$ is a sequence of bounds on the optimal electronic switching for the tree network, where each $\phi_{\mathcal{T}}^{(i)}(p) = \Phi_i$ for the decomposed star of p is an increasingly better lower bound on $\phi_{\mathcal{T}}(p)$. Since $\{\Phi_i\}$ for the star is a strong sequence of bounds, so is the sequence of bounds for the tree that we have described.

Similarly, we can obtain an upper bound by creating a feasible solution in which every interior node forwards all traffic electronically. (We call such a node *opaque*.) 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, this solution can be significantly improved upon. Consider that 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. While two adjacent interior nodes p and q are unlikely to attain $\phi_{\mathcal{T}}(p)$ and $\phi_{\mathcal{T}}(q)$ in the same feasible solution to the tree (because the optimal solutions of the two decomposed stars will in general require the same traffic component in the tree to be groomed differently), if q is opaque then it can reconcile with whatever grooming decisions p makes. In other words, if we interpose at least one opaque node between every two transparent nodes of 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). Ideally, we would like to find the set of nodes N_t to be designated as non-opaque nodes, (composed of pairwise non-adjacent interior nodes) such that $\sum_{p \in R}(\psi_{\mathcal{T}}(p) - \phi_{\mathcal{T}}(p))$ is maximized. However, this is equivalent to finding a maximal independent set in a graph, which is NP-Complete. An efficient way to pick N_t is to utilize the level ordering of the tree $\mathcal{T}$; this approach leads to an algorithm with a good performance guarantee. Details of this approach can be found in [6].

5.3. A Greedy Heuristic

Because the sequence of feasible solutions we have proposed above get progressively more costly to compute, efficient heuristics would be valuable. Our solutions never form lightpaths of more than two hops, so it is possible that heuristics which are allowed to form longer lightpaths will outperform these solutions in specific cases.

The heuristic attempts to optically route the traffic components in decreasing order of magnitude: it considers each component and attempts to assign an end-to-end lightpath for it. 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 this case, the traffic component is carried on single-hop lightpaths from source to destination, undergoing electronic switching at all intermediate nodes. The algorithm terminates when all traffic elements have been examined. The above algorithm can be extended to consider other possibilities, e.g., carry a given component on a lightpath as long as possible before resorting to electronic switching. We have investigated such extensions but do not present any results here, since they are similar to the ones we discuss next.

5.4. Numerical Results

Figure 8 plots the grooming effectiveness of the solutions using the star decomposition and the greedy algorithm for thirty problem instances. Each instance 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).

As we can see, the star decomposition has a grooming effectiveness of at most 0.5, as expected. The greedy heuristic, on the other hand, sometimes performs better than the decomposition, and sometimes worse; 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 instances which we cannot include due to lack of space.

6. 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 is 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, indicating that significant gains in terms of electronic switching costs can be achieved by appropriate traffic grooming.

One of our immediate research goals is the extension of these results to the traffic grooming problem in general network topologies. Previous research in the area of logical topology design and/or traffic grooming in general topologies has been experimental in nature, with little or no formal analysis of the heuristics developed. We believe that our results are a first step towards a more formal and systematic approach to the traffic grooming problem which combines graph-theoretic techniques with algorithms for elementary networks.

REFERENCES

1. R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Network* **16**, pp. 46–56, November/December 2002.
2. R. Dutta and G. N. Rouskas, "On optimal traffic grooming in WDM rings," *IEEE Journal on Selected Areas in Communications* **20**, pp. 110–121, January 2002.
3. A. L. Chiu and E. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *IEEE/OSA Journal of Lightwave Technology* **18**, pp. 2–12, January 2000.
4. S. Huang, "Traffic grooming in wavelength routed path networks," Master's thesis, North Carolina State University, 2002. Publicly available via WWW: *http://www.lib.ncsu.edu/ETD-db/*.
5. J.-C. Bermond and D. Coudert, "Traffic grooming in unidrirectional WDM ring networks: the all-to-all unitary case," in *Proceedings of the 7th IFIP Working Conference on Optical Network Design and Modeling*, Feb 2003.
6. R. Dutta, *Virtual Topology Design for Traffic Grooming in WDM Networks*. PhD thesis, North Carolina State University, Raleigh, NC, August 2001.