# Clustering Methods for Hierarchical Traffic Grooming in Large Scale Mesh WDM Networks

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

***Abstract*— We consider a hierarchical approach for traffic grooming in large multiwavelength networks of a general topology. Inspired by similar concepts in the airline industry, we decompose the network into clusters, and select a hub node in each cluster to groom traffic originating and terminating locally. Clustering and hierarchical grooming enables us to cope with large network sizes and facilitates the control and management of traffic and network resources. Yet determining the size and composition of clusters so as to yield good grooming solutions is a challenging task. We identify the grooming-specific factors affecting the selection of clusters, and we develop a parameterized clustering algorithm that can achieve a desired tradeoff among various goals.**

## I. Introduction

Ongoing advances in optical network and communication technologies continue to expand the capacity of individual wavelengths and increase the availability of wavelength channels for direct optical connections. Traffic grooming, the area of research concerned with efficient and cost-effective transport of sub-wavelength traffic over multigranular networks, has emerged as an important field of study in recent years. Early work on traffic grooming focused on the ring topology, reflecting the technological push in response to the industry's effort to upgrade deployed SONET infrastructure to WDM technology. More recently, several studies have begun to address grooming issues in networks with a general topology. The reader is referred to [8] for a comprehensive survey and classification of research on traffic grooming. Nevertheless, most studies regard the network as a flat entity for the purposes of lightpath routing, wavelength assignment, and traffic grooming. In general, such approaches do not scale well to networks of realistic size for two reasons: first, the running-time complexity of traffic grooming algorithms increases rapidly with the size of the network; and second, the operation, management, and control of multigranular networks becomes a challenging issue in large, unstructured topologies.

We have recently proposed a scalable hierarchical framework for traffic grooming that can be applied to networks of practical size covering a national or international geographical area [3]. Our model borrows ideas from the hub-and-spoke paradigm used within the airline industry. The network is partitioned into clusters, and one node within each cluster is selected as the *hub*. Non-hub nodes route all their traffic to the hub, where it is groomed before it is forwarded to the destination cluster; as a result, the hub is the only node in a cluster responsible for grooming traffic not originating/terminating locally. At the second level of the hierarchy, the first-level hubs form another cluster for grooming and routing inter-cluster traffic. This hierarchical approach is quite scalable and is applicable to both the static and dynamic grooming contexts.

One important yet challenging issue in hierarchical grooming is the selection of clusters and hub nodes. In this paper, we develop a new parameterized clustering algorithm appropriate for traffic grooming. The algorithm is flexible and allows the network designer to achieve a desired balance among a number of conflicting goals.

Following the introduction, we describe the hierarchical grooming approach in Section II. In Section III we present our clustering algorithm for hierarchical grooming in general topologies. We present numerical results in Section IV, and we conclude the paper in Section V.

## II. Hierarchical Grooming in Mesh Networks

We consider a network of general topology with $N$ nodes. Physical links are bidirectional, and support $W$ wavelengths per direction. The capacity $C$ of each wavelength channel is an integer multiple of a basic transmission unit (e.g., OC-3); $C$ is also known as the *grooming factor*. The demands placed on the network are provided in a traffic demand matrix, $T = [t^{(sd)}]$, where integer $t^{(sd)}$ denotes the amount of (forecast) long-term traffic to be carried from node $s$ to node $d$.

The objective of the traffic grooming problem is to configure the network (i.e., determine the lightpaths to be set up) to carry the entire traffic matrix $T$ while minimizing the total number of electronic ports required at the network nodes. Since each lightpath requires exactly two electronic ports (one at the node at each end of the lightpath), this objective is equivalent to minimizing the number of lightpaths in the resulting logical topology. For a more formal definition of the problem and a general-purpose integer linear programming (ILP) formulation, the reader is referred to [8]. The traffic grooming problem in general topology networks is known to be NP-hard, since it contains as a subproblem the lightpath routing and wavelength assignment (RWA) problem which is itself NP-hard [5]. Consequently, for WDM networks with more than a few nodes, it is important to develop heuristic algorithms which are scalable and can be used to obtain provably good solutions in polynomial time.

Our framework for hierarchical traffic grooming was inspired by the hub-and-spoke paradigm that is widely used by the airline industry. In our approach, a large network is partitioned into a number of clusters, each

This work was supported by NSF grant ANI-0322107.

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

Our hierarchical framework consists of three phases:

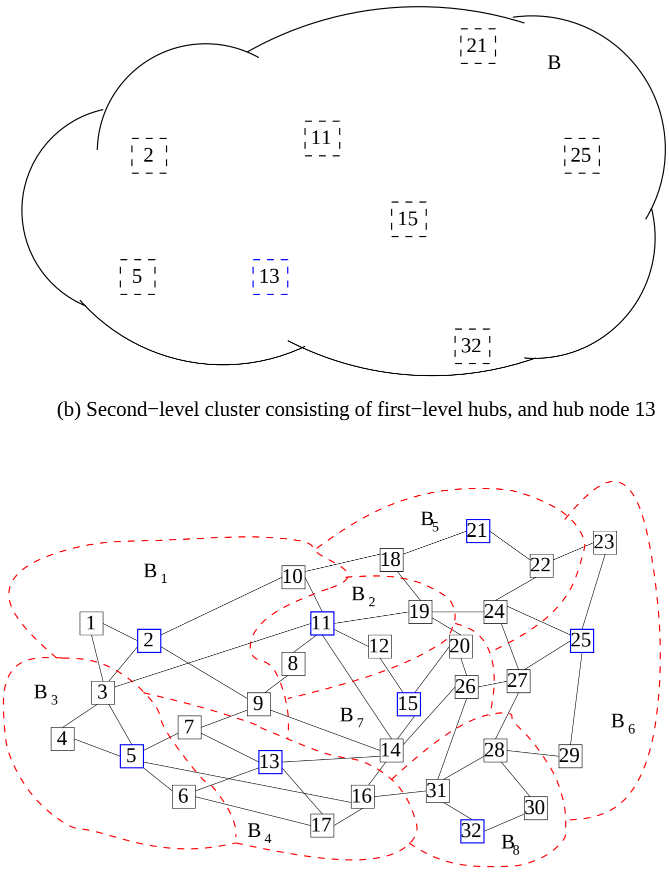
1) **Clustering of network nodes.** In this phase, the network is partitioned into $m$ clusters and one node in each cluster is designated as the hub. The clustering phase is crucial to the quality of the grooming solution.
2) **Hierarchical logical topology design and traffic routing.** The outcome of this phase is a set $R$ of lightpaths for carrying the traffic demand matrix $T$, and a routing of individual traffic components $t^{(sd)}$ over these lightpaths. This phase is further subdivided into three parts:
   a) setup of direct lightpaths for large traffic demands;
   b) intra-cluster traffic grooming; and
   c) inter-cluster traffic grooming.
3) **Lightpath routing and wavelength assignment (RWA).** The goal of the RWA phase is to route the lightpaths in $R$ over the physical topology, and color them using the minimum number of wavelengths.

For a detailed description of each phase, refer to [3]. Figure 1 illustrates the partitioning of a 32-node network into eight first-level clusters, and the second level cluster consisting of the eight first-level hubs.

## III. Clustering for Hierarchical Grooming

Clustering is a function that arises frequently in problems related to network design and organization [11]. Clustering algorithms are classified as either *minimum cut* or *spanning tree*, depending on the underlying methodology. In our case, the goal is to find a clustering that will minimize the number of lightpaths *after* applying the hierarchical grooming (logical design) approach, a fact that adds significant complexity to the problem. Specifically, the input to our problem consists of a traffic demand matrix and several constraints, in addition to the physical network topology; furthermore, unlike typical objective functions considered in the literature (e.g., the physical cut size or the amount of inter-cluster traffic), ours cannot be easily expressed as a function of the resulting clusters. Therefore, most of the existing clustering techniques are not directly applicable to the problem at hand.

Some clustering studies only consider the traffic pattern between nodes. An algorithm that can group a nearly



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

(a) First−level clusters

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

completely decomposable (NCD) matrix into blocks, so that the weighted arcs between blocks have values not exceeding a threshold, was introduced in [6]. Other work has focused on the physical topology only. Typically, the goal is to partition the nodes into contiguous clusters containing roughly equal numbers of nodes, and at the same time minimize the overall cut size. An example is the work on multi-objective graph partitioning [12], which is implemented in the METIS package.

Another family of clustering problems concerned with the physical topology includes the *K-Center, K-Clustering, K-Median* and *Facility Location* problems [2]. Unlike the applications targeted by METIS, they do not require clusters to be of equal size. Of all the variants, the *K-Center* problem is of most interest to us. The goal of the *K-Center* problem is to find a set $S$ of $K$ nodes (centers) in the network, so as to minimize the maximum distance from any network node to the nearest center. Thus, the set $S$ implicitly defines $K$ clusters with corresponding hub nodes in $S$. A solution to the *K-Center* problem may be useful for hierarchical traffic grooming since it is likely to lead to short lightpaths within a cluster, thus lowering the wavelength requirements. The *K-Center* problem is NP-Complete, and the best approximation ratio that can be obtained in polynomial time is 2 [10]. We implemented the 2-approximation algorithm in [10] for *K-Center*, and we compare it to our own clustering method in Section IV.

Some studies have explored clustering techniques in the context of traffic grooming: a hierarchical design for interconnecting SONET rings was proposed in [9], and in [7], the "blocking island" paradigm is used to abstract network resources and find groups of bandwidth hierarchies for a restricted version of traffic grooming. Our work is more comprehensive and it is applicable to many variants of the grooming problem.

We now describe a clustering algorithm tailored to our hierarchical grooming framework. The objective of the algorithm is twofold: to partition the network into some number $m$ of clusters, denoted $B_1, \cdots, B_m$, and to select one node in each cluster to serve as the hub where grooming of traffic is performed. Next, we discuss the tradeoffs involved in selecting the clusters, which set the design principles for our clustering algorithm.

### *A. Important Considerations*

To obtain a good clustering, the number of clusters, their composition, and the corresponding hubs must be selected in a way that helps achieve our goal of minimizing the number of lightpaths and wavelengths required to carry the traffic demands. Therefore, the selection of clusters and hubs is a complex and difficult task, as it depends on both the physical topology of the network and the traffic matrix $T$. To illustrate this point, consider the tradeoffs involved in determining the number $m$ of clusters. If $m$ is small, the amount of inter-cluster traffic will likely be large. Hence, the $m$ hubs may become bottlenecks, resulting in a large number of electronic ports at each hub and possibly a large number of wavelengths (since many lightpaths may have to be carried over the fixed number of links to/from each hub).

On the other hand, a large value for $m$ implies a small number of nodes within each cluster. In this case, the amount of intra-cluster traffic will be small, resulting in inefficient grooming (i.e., a large number of lightpaths); similarly, at the second-level cluster, $O(m^2)$ lightpaths will have to be set up to carry small amounts of inter-cluster traffic. Therefore, the network designer must select the number and size of clusters to strike a balance between capacity utilization and number of lightpaths for both intra- and inter-cluster traffic.

Now consider the composition of each cluster. If the average traffic demand between nodes within a cluster is higher than the average inter-cluster demand, there will tend to be fewer inter-cluster lightpaths, which are typically longer than local lightpaths. Therefore, it is desirable to cluster together nodes with "denser" traffic between each other: doing so reduces the number of longer lightpaths, alleviates hub congestion, and provides more flexibility to the RWA algorithm (since long lightpaths are more likely to collide during the RWA phase).

On the physical topology side, we also need to consider the cut links between clusters. Each cluster has a number of fibers that link to nodes outside the cluster, and all traffic between a node outside the cluster and one within must traverse these cut links. Since the cut links must have sufficient capacity to carry the inter-cluster traffic, it is important to select clusters so that their cut size is not too small, in order to keep the wavelength requirements low.

Another important consideration arises in physical topologies for which there exists a critical small cut set that partitions the network into two parts. In such a topology, all traffic between the two sides of the bisection will have to go through the cut. In this case, creating clusters that consist of nodes on different sides of the cut may be undesirable, because it may generate unnecessary traffic that goes back and forth through the cut. Consider a cluster with nodes $i, j$, on one side of the bisection, and the hub $h$ on the other. Due to the nature of the hierarchical grooming approach, traffic between $i$ and $j$ may need to be sent to the hub first, creating additional traffic across the cut links, with a corresponding increase in the number of required wavelengths. This additional traffic can be eliminated by forcing nodes on different sides of the bisection to be in different clusters. We have employed a pre-cutting technique that can be useful in such situations.

The physical shape of each cluster may also affect the wavelength requirements. In particular, it is important to avoid the creation of clusters whose topology resembles that of a path, since in such topologies the links near the hub can become congested. Since we use a *virtual star* approach for logical topology design within each cluster, topologies with relatively short diameter are more attractive in terms of RWA.

### *B. The MeshClustering Algorithm*

Figure 2 provides a pseudocode description of our MeshClustering algorithm which we use to partition a network of general topology in order to apply our hierarchical traffic grooming framework. The algorithm includes several user-defined parameters that can be used to control the size and composition of clusters, either directly or indirectly. Parameters *MinCS* and *MaxCS* represent the minimum and maximum cluster size, respectively. Our algorithm treats these parameters as an *indication* of the desirable range of cluster sizes, rather than as hard thresholds that cannot be violated.

The parameter $\Delta$ ($0.5 \leq \Delta \leq 0.8$, default value $\Delta = 0.8$) is used to test whether there is sufficient capacity at the hub node, as well as the edges connecting the cluster to the rest of the network, to groom/carry the traffic demands. Specifically, we require that the inter-cluster traffic originating from or terminating at a given cluster do not exceed a fraction $\Delta$ of the hub capacity (this is the HUBTEST in Step 9 of the algorithm); similarly, this intra-cluster traffic must not exceed a fraction $\Delta$ of the capacity of the links connecting the cluster to the rest of the network (the CUTTEST in Step 10 of the algorithm). The algorithm will consider a node to add to a cluster only if doing so will not violate these two constraints.

The parameter $\delta$ controls the ratio of the diameter of a cluster to the number of nodes it contains. In order to avoid cluster topologies that resemble long paths, we require that $0 < \delta \leq 0.75$. We used the value $\delta = 0.75$; this value corresponds to a 4-node path, restricting the

**A Clustering Algorithm for Mesh Networks**

**Input:** A mesh network with a set $V$ of $|V| = N$ nodes, capacity $C$ for each wavelength, and reduced traffic matrix $T_r = [t_r^{(sd)}]$.

**User-defined parameters:** $MinCS, MaxCS$ for the desired minimum and maximum cluster size, respectively, a threshold $0.5 \le \Delta \le 0.8$, a cluster diameter-to-nodes ratio $0 < \delta \le 0.75$, and an intra-to-inter-cluster traffic ratio $0.8 \le \rho \le 1.25$.

**Output:** A partitioning of the node set $V$ into some number $m$ of clusters, $B_1, \ldots, B_m$, and the selection of node $h_i$ as the hub of cluster $B_i$, such that the size of each cluster is roughly between $MinCS$ and $MaxCS$ and the clustering will lower the lightpath and wavelength requirements of the subsequent hierarchical logical topology design and RWA algorithms.

Procedure **MeshClustering**
**begin**
1. **while** $V \neq \phi$ **do**
2. $\quad v \leftarrow$ node in $V$ with maximum remaining capacity
3. $\quad B \leftarrow \{v\}$ // new cluster $B$ with hub $v$
4. $\quad V \leftarrow V - \{v\}$
5. $\quad$ **while** $V \neq \phi$ and $|B| < MaxCS$ **do** // grow cluster $B$
6. $\quad\quad Q \leftarrow$ set of nodes $\in V$ adjacent to nodes in $B$
7. $\quad\quad$ **foreach** node $q \in Q$ **do**
8. $\quad\quad\quad B' \leftarrow B \cup \{q\}$ // assume $q$ is included in $B$
9. $\quad\quad\quad$ HUBTEST: does traffic between $B'$, $\overline{B'}$ occupy more than $\Delta$ of the remaining hub capacity?
10. $\quad\quad\quad$ CUTTEST: does traffic between $B'$, $\overline{B'}$ occupy more than $\Delta$ of the remaining cut link capacity?
11. $\quad\quad\quad$ **if** $q$ passes both tests **then**
12. $\quad\quad\quad\quad x \leftarrow$ total traffic between $q$ and nodes in $B$
13. $\quad\quad\quad\quad y \leftarrow$ total traffic between $q$ and nodes in $\overline{B'}$
14. $\quad\quad\quad\quad \rho_q \leftarrow x/y$ //intra- to inter-cluster traffic ratio
15. $\quad\quad\quad\quad d \leftarrow$ diameter of induced subgraph $B'$
16. $\quad\quad\quad\quad \delta_q \leftarrow d/|B'|$ // diameter-to-nodes ratio
17. $\quad\quad\quad$ **else** $Q \leftarrow Q - \{q\}$
18. $\quad\quad$ **end for**
19. $\quad\quad$ **if** $Q = \phi$ **then break** // cannot grow cluster $B$
20. $\quad\quad$ **else**
21. $\quad\quad\quad q_0 \leftarrow$ node $\in Q$ with largest $\rho_q$ and smallest $\delta_q$
22. $\quad\quad\quad B \leftarrow B \cup \{q_0\}$ // grow cluster $B$ to include $q_0$
23. $\quad\quad\quad V \leftarrow V - \{q_0\}$
24. $\quad$ **end while** // continue until $B$ cannot grow further
25. **end while**
26. Combine clusters of size $< MinCS$ with adjacent clusters

**end**

Fig. 2. Clustering algorithm for mesh networks

longest path within a cluster to no more than three links. Finally, the parameter $\rho, 0.8 \le \rho \le 1.25$, specifies the acceptable range for the ratio of intra- to inter-cluster traffic for a given cluster. Since it is desirable to cluster together nodes that exchange a substantial amount of traffic among themselves relative to traffic they exchange with the rest of the network, we let $\rho = 1.25$.

The MeshClustering algorithm in Figure 2 generates one cluster during each iteration between Steps 1 and 25. Initially, in Steps 2-4, the hub of a new cluster $B$ is selected as the node with the maximum remaining capacity among those not yet assigned to a cluster. We grow the cluster by adding one node during each iteration between Steps 5 and 24. At each iteration, the set $Q$ of candidate nodes for inclusion in $B$ consists of all nodes, not yet assigned to another cluster, which are adjacent to nodes in $B$. For each node $q \in Q$, we check whether including $q$ in $B$ would result in a cluster that passes both the HUBTEST and CUTTEST; if not, node $q$ is removed for consideration for inclusion into cluster $B$ (Step 17). For all nodes $q$ that pass both tests, we compute the diameter-to-nodes ratio $\delta_q$ and intra-to-inter-cluster traffic ratio $\rho_q$, assuming that $q$ is added to cluster $B$ (Steps 11-16). Let $q_0$ be a node that passes both tests and has the largest $\rho_q$ value among the candidates; if there are multiple such nodes, we select the one with the smallest $\delta_q$ value. We include $q_0$ to cluster $B$ (Steps 21-23), and the process is repeated as long as the size of $B$ is less than *MaxCS*.

Once all nodes have been assigned to clusters, it is possible for one or more of the clusters to have fewer than *MinCS* nodes. In this case, at Step 26, the algorithm removes these clusters and includes their nodes into adjacent clusters. As a result, at the end of the algorithm some clusters may contain more than *MaxCS* nodes.

The asymptotic complexity of the algorithm is $O(N^4)$. However, this bound is quite loose; in practice, we have found that the algorithm takes only a few seconds for the 128-node, 321-link network we consider in Section IV.

## IV. Numerical Results

In this section, we present experimental results to demonstrate the performance of our clustering and hierarchical grooming algorithms. The traffic matrix $T = [t^{(sd)}]$ of each problem instance we consider is generated by drawing $N(N-1)$ random numbers (rounded to the nearest integer) from a Gaussian distribution with a given mean $t$ and standard deviation $\sigma$ that depend on the traffic pattern; the details can be found in [4]. We consider a 128-node, 321-link network which corresponds to the worldwide backbone operated by a large service provider; we obtained the topology information from data documented on CAIDA's web site [1]. We emphasize that the size of this topology is about an order of magnitude larger than the typical topology considered in previous grooming studies, a fact that demonstrates the scalability of our hierarchical grooming approach.

For a given network topology and traffic pattern, we generate thirty problem instances and we compare our MeshClustering algorithm to the *K-Center* algorithm [10]. We consider two performance metrics in our study: the *normalized lightpath count* and the *normalized wavelength count*. The former is the ratio of the number of lightpaths required for hierarchical traffic grooming, when clustering is performed by one of the two algorithms above, to the lightpath lower bound, while the latter is the ratio of the number of wavelengths required to the wavelength lower bound; the derivation of the bounds is presented in [4].

Figures 3 and 4 plot the normalized lightpath and wavelength count, respectively, for thirty instances generated according to the random traffic pattern; Figures 5 and 6 show similar plots but for instances generated according to the rising pattern. For the *K-Center* algorithm, we let the number $K$ of clusters be either 9 or 10, and we selected the parameters of the MeshClustering algorithm so that it also produces either 9 or 10 clusters. Our clustering algorithm slightly outperforms *K-Center* in terms of the number of lightpaths, and both algorithms are relatively

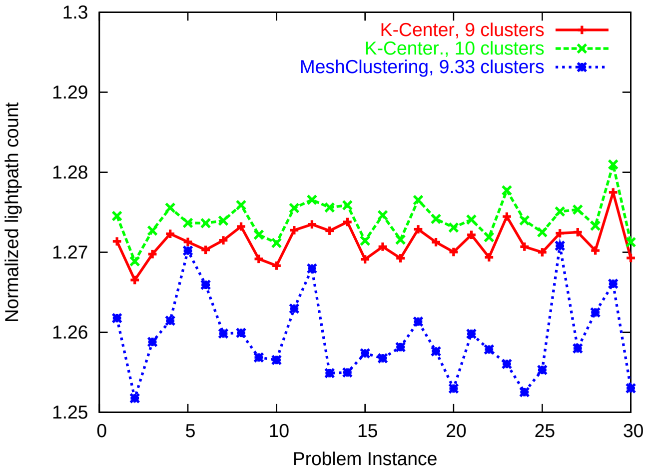



Fig. 3. Lightpath comparison, random pattern, 128-node network

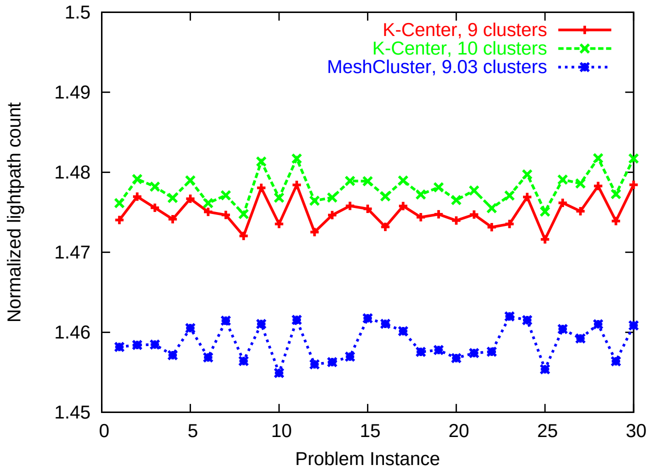



Fig. 5. Lightpath comparison, rising pattern, 128-node network

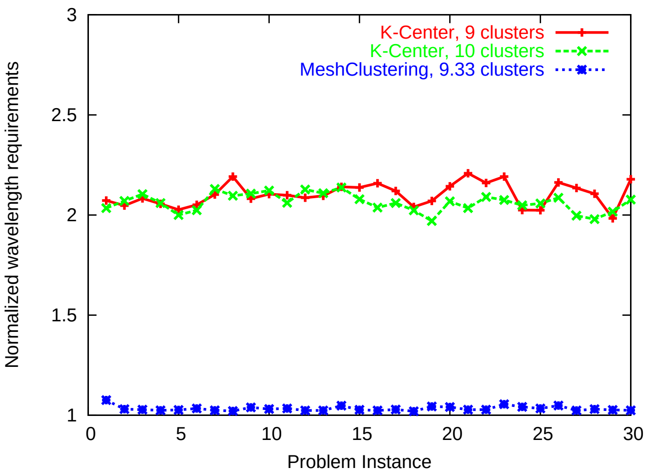



Fig. 4. Wavelength comparison, random pattern, 128-node network

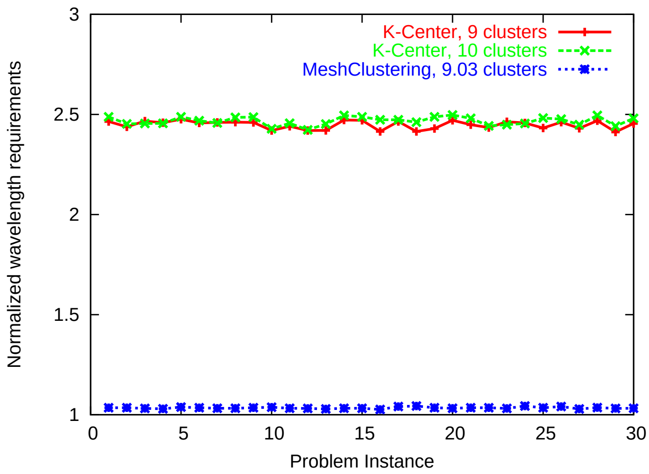



Fig. 6. Wavelength comparison, rising pattern, 128-node network

close to the (loose) lower bound. However, in terms of the number of wavelengths, our algorithm produces results that are within 5% of the lower bound, whereas *K-Center* requires more than twice the number of wavelengths of our algorithm. Similar results for other traffic patterns and network topologies may be found in [4].

## V. Concluding Remarks

We have presented a clustering algorithm for hierarchical traffic grooming that is flexible in balancing various conflicting goals via user-defined parameters. Overall, our experimental results demonstrate that hierarchical grooming combined with specially designed clustering techniques produce logical topologies that perform well in terms of both lightpath and wavelength requirements.

## References

[1] http://www.caida.org.
[2] J. Bar-Ilan, G. Kortsarz, and D. Peleg. How to allocate network centers. *J. Algorithms*, 15:385–415, 1993.
[3] B. Chen, G. N. Rouskas, and R. Dutta. A framework for hierarchical grooming in WDM networks of general topology. In *Proceeding of BROADNETS 2005*, pages 167–176, October 2005.
[4] Bensong Chen. *Hierarchical Traffic Grooming in Large-Scale WDM Networks*. PhD thesis, North Carolina State University, Raleigh, NC, August 2005.
[5] I. Chlamtac, A. Ganz, and G. Karmi. Lightpath communications: An approach to high bandwidth optical WANS. *IEEE Transactions on Communications*, 40(7):1171–1182, July 1992.
[6] H. Choi and D. B. Szyld. Application of threshold partitioning of sparse matrices to markov chains. In *Proceedings of the IEEE International Computer Performance and Dependability Symposium (IPDS)*, 1996.
[7] Z. Ding and M. Hamdi. Clustering techniques for traffic grooming in optical WDM mesh networks. In *IEEE GLOBECOM 2002*, volume 3,17-21, pages 2711–2715, Nov 2002.
[8] R. Dutta and G. N. Rouskas. Traffic grooming in WDM networks: Past and future. *IEEE Network*, 16(6):46–56, November/December 2002.
[9] M. Esfandiari, S. Gloeckle, A. Zolfagheri, G. Clapp, J. Gannett, H. Kobrinski, V. Poudyal, and R. Skoog. Improved metro network design by grooming and aggregation of STS-1 demands into oc-192/oc-48 lightpaths. In *NFOEC 2003*, September 2003.
[10] T. Gonzalez. Clustering to minimize the maximum inter-cluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.
[11] John A. Hartigan. *Clustering Algorithms*. Wiley, 1975.
[12] K. Schloegel, G. Karypis, and V. Kumar. A new algorithm for multi-objective graph partitioning. Technical Report 99-003, Dept. of Computer Science, Univ. of Minnesota, September 1999.