Traffic Grooming in WDM SONET Rings with Multiple Line Speeds

Huan Liu and Fouad A. Tobagi

***Abstract*—We consider the traffic grooming problem in WDM/SONET rings with multiple line speeds. This is motivated by the fact that when traffic demands are non-uniform and are spread over a relatively wide range, a ring using multiple line speeds would lead to a lower cost, owing to the economy of scale seen in devices, in particular, electronic ADMs, running at higher speeds. We give novel Integer Linear Programming (ILP) formulations for the problem. We also propose techniques that exploit the problem structure and, thereby, reduce the computation time. As a result, many problem instances can be solved exactly. For large size problems, we propose an efficient heuristic algorithm that achieves a similar cost using a fraction of the computation time. We show that, by allowing WDM/SONET rings to run at different line speeds, we can greatly reduce the ADM cost. We also study the cost benefits of traffic switching, compare UPSR and BLSR, and study the trade-offs of shortest path routing in BLSR.**

***Index Terms*—Simulations, mathematical programming/optimization.**

I. Introduction

WDM/SONET architecture is gaining popularity because it provides an easy upgrade path, avoiding the need to lay out additional fibers to meet the traffic growth. In WDM/SONET, many rings run in parallel and each ring runs on a separate wavelength. Each node needs one Optical Add-Drop Multiplexer (OADM), which allows the node to add/drop a selected subset of the wavelengths into and out of the fiber. Each node may also need up to W electronic ADMs, one for each wavelength on which the node originates or terminates some traffic demands.

There are two types of SONET rings: Unidirectional Path Switched Rings (UPSR) and Bidirectional Line Switched Rings (BLSR)[4]. In UPSR, traffic is always routed in one direction (e.g., clockwise). To route a symmetric traffic stream, i.e., the same amount of traffic from node A to B and from node B to A, the same amount of capacity is reserved on every fiber along the ring. Therefore, no spatial capacity reuse is possible. UPSR ring consists of 2 fibers where one of the fibers is dedicated for protection purpose. In BLSR, traffic can be either routed in the clockwise or routed in the counter-clockwise direction. BLSR consists of either two or four fibers. They are commonly referred to as BLSR/2 and BLSR/4 respectively. UPSR is commonly used in Metro access networks, whereas BLSR is commonly used in Metro backbone networks.

Manuscript received December 18, 2005; revised December 20, 2006. Preliminary versions of this paper appeared in [1][2][3].

H. Liu is with Accenture Technology Labs (e-mail: huan.liu@accenture.com).

F. Tobagi is with the Department of Electrical Engineering, Stanford University (e-mail: tobagi@stanford.edu).

Digital Object Identifier 10.1109/JSAC-OCN.2007.024405.

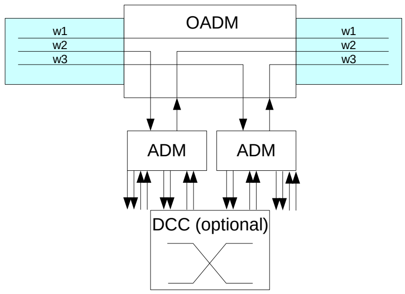

Fig. 1. Node architecture.

A traffic demand can be routed on any one of the wavelengths. If the traffic demand is routed on one wavelength, then an ADM has to be installed on that wavelength at both the source and destination nodes. Fig. 1 shows an example of the node architecture. In the example, a fiber with three wavelengths is connected to the node through an OADM. Wavelength w_2 and w_3 are dropped at the node, and the other wavelength (w_1) is passed through transparently. Thus, only two electronic ADMs are needed at the node. The ADMs can add or drop individual traffic streams into the corresponding wavelength. An optional Digital Cross Connect (DCC) may be also present at the node to switch traffic from one ADM to another.

A major design goal of a WDM/SONET network is to minimize the total equipment cost by intelligently arranging the traffic demands onto the different SONET rings. This is commonly referred to as traffic grooming. Since an OADM is needed at every node, the cost of OADMs is fixed. However, the cost of electronic ADMs varies depending on how traffic is groomed. A naive solution is to install an ADM on every wavelength at each node. Unfortunately, such a solution is very costly. An example is given in [5] which shows that significant cost savings could be achieved by intelligently arranging traffic demands onto different rings. The potential savings can be significant and grow with both the network size and internodal demand [6]. Although only very simple traffic demand matrices are considered in [6], we would expect the savings to be significant for other traffic patterns too.

In addition to determining the routing, it is also important to determine what line speed each wavelength should be set to. If there are enough traffic demands between some nodes, they should be routed on a ring with high line speed to enjoy the economy of scale of high-speed ADMs. On the other hand, if the traffic demands are small, they should be put on a ring with low line speed to reduce ADM cost.

Despite of the recent progress, designing a low cost WDM/SONET network that supports multiple line speeds remains a difficult task. When the network operators design the network, they would have to guess what line speed these rings have to be set to. Choosing the wrong line speed could greatly increase the cost. Even if they can choose the line speed correctly, the cost still could be much higher than if a mixture of line speeds can be chosen, especially when the traffic demands are heterogeneous. One can attempt to guess what line speed each ring should be set to. But, not only it is impossible to enumerate all possible line speeds assignments, but there is also no methodology available to design the network with mixed line speeds, even if the line speeds assignments have been done correctly.

In this paper, for the first time, we propose a comprehensive methodology to solve the traffic grooming problem in WDM/SONET rings with multiple line speeds. Our solution approach includes both Integer Linear Programming (ILP) formulations, which can be used to solve small size problems exactly, and an efficient heuristic algorithm that can solve large size problems quickly. Our solution approach makes no specific assumption regarding the cost and speed ratios between different line speeds, therefore, it can be applied directly to real network designs.

A. Prior work

The traffic grooming problem where all SONET rings are restricted to the same line speed was first considered in [7] [8] [4]. It was first considered in two separate steps: traffic routing [7] and wavelength assignment [8]. In the subsequent paper [4], it was shown that considering the traffic grooming problem in one single step could lead to lower cost. Traffic grooming in SONET UPSR rings with one single line speed has been considered in [9], where heuristic algorithms were proposed for specific traffic patterns, and in [10] [11], where an Integer Linear Programming (ILP) formulation was given.

Traffic grooming in SONET BLSR rings with only one line speed has also been studied. An ILP problem formulation has been given in [12] which imposes the timeslot continuity constraint (A circuit has to occupy the same time slot from its source to its destination). A multi-hop formulation where only one node can switch traffic was given in [12], where it was assumed that the hub node can have as many ADMs as the number of wavelengths. Another ILP formulation has been given in [13] where the time slot continuity constraint is relaxed. Several heuristic algorithms have been proposed before too. In [14], a circle construction algorithm was proposed. It is a two step approach; the first step is to build circles, then the second step is to group circles. This approach is only applicable when the timeslot continuity constraint can not be relaxed. Another simulated annealing based heuristic was also proposed in [12].

These results unfortunately do not apply when a wavelength can choose from a set of available line speeds. Prior work on traffic grooming with multiple line speeds can be found in [4] [15] [16]. Traffic grooming in UPSR rings with two line speeds was first briefly discussed in [4]. The authors compared network cost using a simple lower bound. The lower bound indicates that having two line speeds available might lower network cost under certain cases. Traffic grooming in UPSR rings with a central hub node was considered in [15] [16] where all traffic terminates at the central hub node. The authors did not take the number of wavelengths as a constraint, as a result, the solution may not be feasible if only a limited number of wavelengths are available. No problem formulation or general solution methodology has been given in [4] [15] [16], and furthermore, their analysis assume specific cost and capacity ratios between two different line speeds. The result cannot be easily generalized when the ratios are different.

B. Assumptions

Similar to prior work on traffic grooming [4] [15] [16] [9] [10], we limit ourselves to consider only bidirectional traffic demands. However, we make no specific assumption regarding the cost and speed ratios between different line speeds. Unlike previous work [16] [15] where no limit on the number of wavelengths is assumed, we take the number of wavelengths available as a constraint, since a WDM system typically supports a fixed number of wavelengths, such as 8, 12 or 40 wavelengths, in a single fiber.

C. Organization of this paper

The paper is organized as follows. In section II, we present novel ILP formulations which can be used to solve the traffic grooming problem exactly. The formulations can not only reduce the computation time, but also allow us to study the cost benefits of allowing traffic to switch across rings at some nodes. In section III, we propose several techniques that can reduce the computational complexity. These techniques, along with our novel ILP formulations, allow us to solve much larger problem instances and allow us to further validate our heuristic algorithms. In section IV, we present efficient heuristic algorithms that can obtain comparable results in a much shorter amount of time. In section V, we evaluate our algorithms and approaches. Then, in section VI, we present some numerical results. Finally, in section VII, we conclude the paper.

NOMENCLATURE

- N: The number of nodes in the ring, i.e., the number of traffic add/drop sites around the ring. Nodes are labeled $0, 1, \ldots, N-1$ in the clockwise direction.
- W: The number of wavelengths available in the WDM system.
- R: The number of line speeds available. For example, if OC-3, OC-12 and OC-48 line speeds are available, $R = 3$.
- w: When used as a subscript, it denotes a wavelength. $w \in 1, \ldots, W$.
- r: When used as a subscript, it denotes a line speed. $r \in 1, \ldots, R$.
- i, j, s, d: They denote nodes on the rings. $i, j, s, d \in 1, \ldots, N$.
- f_{ij}: f_{ij} is an integer that describes the total traffic demand from node i to node j in terms of the lowest traffic

granularity. For example, if the lowest traffic granularity is OC-3, and the traffic demand between node i and j is one OC-48 circuit, then $f_{ij} = 16$. We assume f_{ii} is always 0.

- g_r: Capacity of the rth line speed in multiples of the lowest traffic granularity. If the lowest traffic granularity is OC-3, then line speed at OC-3 has capacity of 1, line speed at OC-12 has capacity of 4 and line speed at OC-48 has capacity of 16. We assume that the capacity is an integer number throughout this paper.
- c_r: The cost of an ADM running at the rth line speed. An OC-48 ADM will cost more than an OC-12 ADM, which in turn cost more than an OC-3 ADM.
- M: A large constant.

II. ILP FORMULATIONS

The traffic grooming problem can be formulated as an Integer Linear Programming (ILP) problem and solved using commercial ILP solvers. The ILP formulation for UPSR for the single line speed case was first given in [10] [11]. The formulation can be easily extended to the multiple line speeds case by creating R different rings for each wavelength, each running at one of the R available line speeds. An additional constraint makes sure that only one of them is chosen. We use the term "ring wr" to refer to the ring running at the rth line speed on wavelength w.

One advantage of the ILP approach is that we can easily extend the formulations to accommodate a number of other constraints and considerations. For example, the formulation could take switching cost into consideration, take limited number of ADM equipment as a constraint or disallow bifurcation of traffic. The detail of these can be found in [17].

We use the following notations to describe the formulation.

- We use x variables (e.g. $x^{sd}(w,r)$, $x^{s}_{i+}(w,r)$, $x^{s}_{i,i+1}(w,r)$) to denote the amount of traffic. The superscript refers to one or more traffic demands, which we refer to as a commodity. We use sd to denote the traffic demand between nodes s and d and s to denote all traffic demands from node s. The subscript denotes on which link or node the commodity is routed through. The parameter (w,r) specifies on which wavelength and line speed this traffic is carried on. In general, the x variables are all integers.
- We use y variables to denote where ADMs have to be installed. $y_i(w,r) = 1$ if an ADM is needed at node i on wavelength w with line speed r. y variables are binary variables, i.e., they can only be either 0 or 1.
- We use δ_{wr} variables to denote the line speed of a wavelength. It is 1 if ring wr can carry traffic, otherwise it is 0. δ_{wr} are binary variables.

A. Disaggregate ILP formulation for UPSR

We first present the formulation that extends the formulation in [10][11] to the multiple line speed case. We call it the disaggregate formulation because it treats each traffic demand as a separate commodity. The objective of the traffic grooming problem is to minimize the cost of ADMs. It can be expressed as

$$\min \sum_i \sum_w \sum_r c_r y_i(w,r) \tag{1}$$

The first constraint—the demand met constraint—makes sure that the total traffic demand is routed on some rings.

$$\sum_r \sum_w x^{sd}(w,r) = f_{sd} \quad \forall sd \tag{2}$$

The second constraint—the capacity constraint—makes sure that the total capacity of each ring is not violated.

$$\sum_{sd} x^{sd}(w,r) \le \delta_{wr} g_r \quad \forall w,r \tag{3}$$

The third constraint—called the ADM constraint—relates $y_i(w,r)$ to $x^{sd}(w,r)$ variables. It forces the $y_i(w,r)$ variable to be 1 if an $x^{sd}(w,r)$ variable is nonzero and either $s = i$ or $d = i$.

$$My_i(w,r) \ge \sum_d x^{id}(w,r) + \sum_s x^{si}(w,r) \quad \forall i,w,r \tag{4}$$

Lastly, a wavelength constraint makes sure that only one out of the R rings on wavelength w can carry traffic.

$$\sum_r \delta_{wr} \le 1 \quad \forall w \tag{5}$$

The ILP formulation above is an easy extension of the formulation given in [10] [11] for the single line speed case. This formulation has two shortcomings. First of all, a traffic demand has to stay on the same ring from its source to its destination. If we allow traffic to be switched at any node from one ring to another, potentially large savings are possible. There are two ways of switching traffic at a node as described in [4]. One is by manually connecting one traffic port from an ADM to one traffic port on another ADM sitting on a different wavelength. Another is to utilize DCC which can switch traffic streams through programming control.

The second shortcoming is that the formulation treats each traffic demand as a separate commodity, therefore, there are a large number of commodities. Each commodity corresponds to a separate flow sub-problem that must satisfies the demand met constraint (equation 2), and these sub-problems are connected by the capacity constraint (equation 3). If we could *aggregate* many traffic demands, such as all those originating from a single node, as one single commodity, the number of commodities could be greatly reduced (from $O(N^2)$ to $O(N)$). As a result, the number of flow sub-problems is reduced and the computation time could be much shorter. Aggregate formulation has been used in network design problems [18] and in Logical Topology design problems [19] where it showed promise for reduction in computational complexity.

B. Aggregate ILP formulation with traffic switching for UPSR

In this section, we introduce a novel formulation—the aggregate formulation—which allows all traffic demands originating from a node (or all traffic demands terminating at a node) to be treated as a single commodity. Furthermore, the formulation also allows traffic to switch at every node.

In the formulation, we use a pair of variables—$x^s_{i+}(w,r)$ and $x^s_{i-}(w,r)$. $x^s_{i+}(w,r)$ ($x^s_{i-}(w,r)$) is the amount of traffic originating from node s that leaves from (arrive at) node i on ring wr. Node i is not necessarily the source or destination of some traffic demands because the traffic could be switching from (to) another ring at node i. There are $KNWR$ number of $x^s_{i+}(w,r)$ variables and an equal number of $x^s_{i-}(w,r)$ variables, where K is the number of commodities, e.g., the number of nodes that originate some traffic.

The optimization objective is the same as before, i.e., equation (1). The demand met constraint should be rewritten as follows.

$$\sum_r \sum_w (x^s_{i+}(w,r) - x^s_{i-}(w,r)) = \begin{cases} \sum_j f_{ij} & \text{if } s = i \\ -f_{si} & \text{otherwise} \end{cases} \quad \forall s, i \tag{6}$$

Since we now use a pair of variables, we have to make sure that the traffic generated on a ring also terminates on the same ring.

$$\sum_i (x^s_{i+}(w,r) - x^s_{i-}(w,r)) = 0 \quad \forall w, r, s \tag{7}$$

The capacity and ADM constraints are the same as before, but, need to be modified to use the new variables. Note that, for the capacity constraint, we only need to count flows that originate on the ring, hence, we are only adding $x^s_{i+}(w,r)$ variables.

$$\sum_s \sum_i x^s_{i+}(w,r) \le \delta_{wr} g_r \quad \forall w, r \tag{8}$$

$$M y_i(w,r) \ge \sum_s (x^s_{i+}(w,r) + x^s_{i-}(w,r)) \quad \forall i, w, r \tag{9}$$

Lastly, the wavelength constraint remains the same as before, i.e., equation (5).

The solution time for a mixed integer linear programming problem is heavily dependent on the number of integer variables in the formulation, because a branch and bound algorithm has to branch on each integer variable in turn to arrive at the optimal solution. The following theorem shows that the actual number of integer variables in the formulation is much smaller than it appears. This applies to both the disaggregate and aggregate formulations. The proof is similar to that in [10], thus it is left out for brevity.

Theorem 1: If the integer constraint on the x variables is removed, there is still an optimal solution where the x variables are integers.

The proof for the theorem is constructive. If after removing the integer constraint, the ILP solver finds an optimal solution to the ILP formulation that has fractional x variables, we can easily construct another optimal solution where all x variables are integers.

C. Aggregate ILP formulation with no traffic switching for UPSR

For the case of no traffic switching, we only need to have $x^s_{i+}(w,r)$ variables for nodes that originate some traffic ($i = s$), and $x^s_{i-}(w,r)$ variables if $f_{si} \neq 0$. Therefore, the demand met constraint (equation 6) can be simplified as follows.

$$\sum_r \sum_w x^s_{s+}(w,r) = \sum_j f_{sj} \quad \forall s \tag{10}$$

$$\sum_r \sum_w x^s_{i-}(w,r) = f_{si} \quad \forall s, i, s \neq i \tag{11}$$

Also, equation (7) can be simplified to be

$$x^s_{s+}(w,r) - \sum_i x^s_{i-}(w,r) = 0 \quad \forall w, r, s \tag{12}$$

The objective function and the rest of the constraints remain the same as in the previous section.

In this formulation, we have KWR number of $x^s_{s+}(w,r)$ variables, one for each source node on each ring wr. $x^s_{i-}(w,r)$ variables are used only when f_{si} is non-zero. Therefore, there are only $K'WR$ number of $x^s_{i-}(w,r)$ variables, where K' is the total number of traffic demands.

Note that this formulation solves exactly the same problem as that of the disaggregate formulation. Equation (12) can be viewed as a definition of a new variable which sums up all traffic originating from a source node. This new variable ($x^s_{s+}(w,r)$) is used in equation (10) to restate the demand met constraint and is also used in equation (8) and (9) to replace unnecessary summations.

Although this formulation has both more variables and more constraints, as hinted at by earlier work in a different area [19] [18] and confirmed by our experimental study, this formulation actually greatly reduce the computation time. On average, the new formulation takes half as much time. The reason that this new formulation performs better is that it exposes the problem structure better, as a result, it helps the branch and bound algorithm to find the optimal integer solution faster.

It is also possible to restrict the set of nodes that can switch traffic. We only introduce $x^s_{i+}(w,r)$ and $x^s_{i-}(w,r)$ variables for any node i that is capable of switching traffic, and apply equation (6) to make sure that the traffic demands are met. For other nodes, we apply equation (10) and (11), and only introduce $x^s_{i+}(w,r)$ and $x^s_{i-}(w,r)$ variables if they are non-zero.

In general, the more nodes we allow switching, the higher the potential for ADM cost savings. Fig. 2 shows an example where allowing node 2 and 5 to switch traffic can result in lower total ADM cost. In the figure, a pair of squares are used to denote the source and destination node of a traffic demand. The link between them shows the path the traffic demand follows. The traffic demand between node 1 and 3 does not need a separate ADM at node 3 on ring 3, it can switch to ring 2 at node 2 and then terminate at the ADM at node 3 on that ring. Similarly, for traffic demand between node 4 and 6, we can save one ADM by switching traffic at node 5.

Note that, even if only one node is capable of switching traffic, it is still different from the hub architecture as described in [7][9][15][20]. In a hub architecture, all traffic has to traverse to the hub before it is forwarded to its final destination. Whereas in our formulation, we only switch traffic at the node if it helps to reduce the network cost. Some traffic may bypass the hub (switching) nodes if routing it directly can reduce cost.

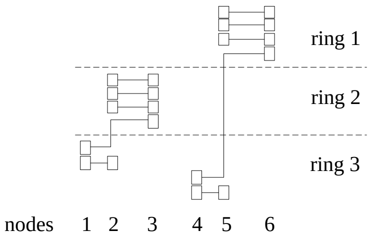

Fig. 2. Allowing two nodes to switch traffic results in lower total ADM cost.

D. Aggregate ILP formulation with traffic switching for BLSR

A disaggregate formulation for BLSR with no traffic switching for the single line speed case was given in [13]. It can be easily extended to the multiple line speeds case. In this section, we present a novel aggregate formulation which also allows traffic to switch across rings. Existing formulations in the literature (only for the single line speed case) either consider only specific scenario or unnecessarily restrict the solution space, which will lead to higher cost [21] [12]. To the best of our knowledge, this is the first time a complete formulation is given for this problem.

The optimization objective remains the same as shown in equation (1).

In BLSR, the two unidirectional traffic demands making up a bidirectional traffic demand are routed on separate fiber rings running in opposite directions (one clockwise, the other counter-clockwise), but along the same arc. Because of symmetry, it is sufficient to consider only one fiber ring and all unidirectional traffic demands on it. The demand met constraint can be expressed as in the following equation. Note that we treat all traffic demands from the same source node as a single commodity. The equation sums up traffic going out of node i (($x^s_{i,i+1}(w,r)$ and $x^s_{i,i-1}(w,r)$), and then subtracts traffic going into node i ($x^s_{i-1,i}(w,r)$ and $x^s_{i+1,i}(w,r)$). The resulting amount is the total traffic originating (or terminating if negative) at node i on ring wr.

$$\sum_r \sum_w (x^s_{i,i+1}(w,r) - x^s_{i-1,i}(w,r) + x^s_{i,i-1}(w,r) - x^s_{i+1,i}(w,r)) = \begin{cases} \sum_j f_{ij} & \text{if } s = i \\ -f_{si} & \text{otherwise} \end{cases} \quad \forall s, i \quad (13)$$

The capacity constraint can be expressed by summing up both the clockwise and counter-clockwise traffic on each link $(i, i+1)$ on each ring wr, and making sure that it is less than the link capacity.

$$\sum_s (x^s_{i,i+1}(w,r) + x^s_{i+1,i}(w,r)) \leq \delta_{wr} g_r \quad \forall w, r, i \quad (14)$$

The ADM constraint can be stated as in the following equation. The terms on the right hand side sum up traffic leaving node i on ring wr. If this sum is non-zero, an ADM has to be installed. Note that, because we are taking the absolute value of the right hand side, this equation corresponds to two separate constraints: one takes the positive value of the right hand side, the other takes the negative value.

$$My_i(w,r) \geq |x^s_{i,i+1}(w,r) + x^s_{i,i-1}(w,r) - x^s_{i+1,i}(w,r) - x^s_{i-1,i}(w,r)| \quad \forall i, w, r, s \quad (15)$$

Lastly, we also need the wavelength constraint (equation 5) to make sure that only one line speed is chosen for each wavelength.

III. Computation techniques for the ILP formulations

Compared to the disaggregate formulation, our novel aggregate formulations greatly reduce the computational complexity. In this section, we further propose several techniques that can be used to reduce the computational complexity of the ILP formulations, and thus allow a much larger problem to be solved.

Our first technique changes how the traffic demands are expressed so that the resulting number of (aggregated) commodities is small, which translates directly into shorter computation time.

The other two techniques exploit the fact that there are a large number of equivalent solutions in the solution space as a result of the line speed assignments and the color assignments. The second technique eliminates the duplication in line speed assignments. The third technique eliminates the duplication in color assignments for wavelengths having the same line speeds. Both techniques allow us to search only the unique solution space, therefore cut down the solution space dramatically. The reason that there are a large number of equivalent solutions is because, in the demand met constraint of the ILP (e.g. equation (2)), we only make sure that the sum of traffic routed on all rings meets the traffic demand, but we are not specifying which set of rings the traffic should be routed on. Therefore, given a solution to the traffic grooming problem, we could permute the traffic assignment from one ring to another and obtain an equivalent solution.

These proposed techniques can be used together to obtain combined computation time reduction. With these techniques, we are able to solve a problem instance in a much shorter amount of time. If we have to terminate the computation early for large size problems, the techniques allow us to explore a larger portion of the solution space in a fixed amount of time, therefore, we are able to obtain better solutions.

It is worth noting that other techniques have also been proposed to reduce the computation time. For example, in [11], the authors attempt to reduce the computation time by exploiting the block diagonal structure in the constraint matrix and use column generation technique to decompose the problem into smaller and easier to solve sub-problems. The column generation technique can be used to speed up the computation of the relaxed Linear Programming problem at any iteration of a branch and bound algorithm.

A. Aggregate commodities

The aggregate formulation treats multiple traffic demands as one single commodity. Since our traffic model is bidirectional, it is equivalent to express a traffic demand from node i to j as a traffic demand from node j to i. Given the choices, we should choose how to express the traffic demands intelligently in order to reduce the total number of commodities. Since the complexity of the formulation grows as the number of commodities increases, reducing the number of commodities will help to keep the problem formulation small, allowing it to be solved quickly. In the following, we show how the run time can be further reduced by changing how a traffic demand is expressed.

The problem of determining how each traffic demand should be expressed such that the total number of commodities is minimized can be easily transformed into a vertex covering problem. The vertex covering problem is NP-complete. Although many heuristic algorithms exist, we propose to solve it exactly because our problem is small enough (SONET can have at most 16 nodes).

The problem can be formulated as the following ILP problem.

$$\min \sum_{i=1}^{N} x_i \tag{16}$$

$$s.t. \quad x_{s(k)} + x_{d(k)} \geq 1 \quad \forall k \tag{17}$$

$$x_i \in \{0, 1\} \tag{18}$$

We use the x_i variables to indicate whether a commodity will be defined for node i. If so, all traffic demands that have node i as one of their end nodes should be expressed as originating from node i, so that they all become part of the commodity. Each traffic demand corresponds to one constraint as expressed by equation (17). The constraint makes sure that at least one commodity is defined at either of the two end nodes of the traffic demand. If a commodity is defined at both end nodes, i.e., $x_{s(k)}$ and $x_{d(k)}$ are both 1, then the traffic demand can be expressed as originating from either node. The objective function (16) tries to minimize the total number of commodities.

B. Exploit symmetry in line speed assignments

The ILP formulation contains a large number of duplicate solutions. Consider a solution to the traffic grooming problem where the ith wavelength is set to the r_ith line speed and the jth wavelength is set to the r_jth line speed, we can construct an equivalent solution by setting the ith wavelength to use the r_jth line speed and setting the jth wavelength to use the r_ith line speed, then moving all traffic routed on wavelength j to wavelength i and moving all traffic routed on wavelength i to wavelength j. The solution thus constructed is clearly identical to the original solution, however, the ILP formulation treats them as separate solutions.

We observe that δ_{wr} are binary variables, and because of equation (5), at most one out of the R number of δ_{wr} variables for any w can be 1. Therefore, the total number of possible assignments for δ_{wr} variables, as would have been enumerated by an ILP solver, is R^W. However, out of the R^W combinations of δ_{wr} variables, only a limited subset is unique. The exact line speed assignment for each wavelength is not important, but rather, the number of wavelengths set to any line speed is more important in determining a unique solution.

For example, if two line speeds are available, then x number of wavelengths can be set to the first line speed and $W - x$ number of wavelengths can be set to the second line speed, where $x \in \{0, 1, \ldots, W\}$. Each value of x corresponds to one unique combination, therefore, there are only $W + 1$ unique combinations out of the 2^W possibilities.

In general, the number of unique combinations when there are many line speeds available is determined by the following formula.

$$\sum_{l_1=0}^{W} \sum_{l_2=l_1}^{W} \cdots \sum_{l_{R-1}=\sum_{i=1}^{R-2} l_i}^{W} 1$$

When two line speeds are available, the above formula becomes $W + 1$, and when three line speeds are available, the above formula becomes $(W + 1)(W + 2)/2$. Consider a network with 10 wavelengths and 3 line speeds, the number of unique line speed assignments is 66 out of the 3^{10} possibilities. If we can examine only these unique assignments, we can greatly reduce the solution space, and at the same time, guarantee that the optimal solution can still be found.

To eliminate the redundant solutions, we propose to decompose the problem into many subproblems, where each subproblem corresponds to one possible way of line speed assignment. Once the line speed assignment is determined, we can formulate the subproblems as ILP problems and solve them directly. Thus, the decomposition allows us to eliminate duplicate solutions by choosing to solve only the subproblems that correspond to unique line speed assignments. The detailed algorithm of this computation technique can be found in [2].

C. Exploit symmetry in color assignments

The technique presented in the last section only eliminates duplicate solutions resulting from the line speed assignments to wavelengths. There are still a large number of duplicate solutions resulting from the equivalency in the color assignments. In this section, we propose to exploit it through a new multi-variable branch and bound technique.

Consider a solution to the traffic grooming problem and any two wavelengths running at the same line speed: w and w', we can easily permute w and w' to obtain another equivalent solution. Again, the ILP formulation treats them as separate solutions.

The number of duplicate solutions resulting from the equivalency is large. Consider the $y_i(w, r)$ variables in the ILP formulation for a single line speed ($R = 1$). Each $y_i(w, r)$ variable can take on the value of either 0 or 1. Therefore, the total number of all combinations of value assignments is $2^{N \times W}$. However, the number of unique combinations after eliminating equivalent solutions is much smaller. Let $C_{N,W}$ denote the number of unique combinations with N nodes and W wavelengths. To derive $C_{N,W}$, let us consider one node at a time. Without loss of generality, let us assume the node is 0. There are 2^W combinations of value assignments for the $y_0(w, r)$ variables. However, the number of unique combinations is small. We note that the number of $y_0(w, r)$ variables that are assigned 1 determines whether a combination is unique. Therefore, there are only $W + 1$ unique combinations. Once we fix an ADM assignment for node 0 (e.g., an ADM on the first w wavelengths and no ADM on the other $W - w$ wavelengths at node 0), we can proceed to consider node 1. The first w wavelengths are still equivalent because there all have an ADM at node 0; hence, the number of unique combinations for these w wavelengths is $C_{N-1,w}$. Similarly, the number of unique combinations for the other $W-w$ wavelengths is $C_{N-1,W-w}$. The total is a product of the two terms summed over all possible assignments at node 0; hence $C_{N,W}$ can be recursively defined as follows:

$$C_{N,W} = \sum_{w=0}^{W} C_{N-1,w} \times C_{N-1,W-w}$$

Note that $C_{N,W}$ is 1 if $N=0$ or $W=0$.

The number of all combinations is much larger than the number of unique combinations. As an example, with 20 wavelengths, the number of all combinations is more than 10^{18} times larger than that of the unique combinations. The difference grows as the number of wavelengths increases. This is expected because the number of duplicated solutions grows exponentially as the number of wavelengths.

We should note that a branch and bound routine will not necessarily explore all combinations because it can frequently prune large solution space based on the current bound. Therefore, the extra time taken for a branch and bound routine to explore the whole solution space rather than only the unique solution space may not be proportionally longer. Still, we expect significant savings if we could limit our search only to the unique solution space.

The branch and bound algorithm normally branches on one fractional integer variable at a time. This works well on many problems. However, it performs poorly on problems with a symmetric structure. This is because even though we force a variable with fractional value to be integer when we branch, the same fractional value will re-appear on other variables because of the symmetry. For example, if a $y_i(w,r)$ variable is fractional and we force it to be either 1 or 0, then another $y_i(w',r)$ variable will become fractional because wavelength w and w' are symmetric and it is equivalent to install an ADM on w or on w'. To force an integer solution, all $y_i(w,r) \quad \forall w$ variables have to be forced to integer values eventually through the branch and bound algorithm. The number of branch nodes created in this process is large and it is proportional to 2^W.

To overcome this problem, we propose a new multi-variable branch and bound technique which will branch on multiple variables at a time instead of only on one single variable. For example, if a $y_i(w,r)$ variable is fractional, we will branch on all $y_i(w,r) \quad \forall w$ variables at the same time. Because of the symmetric structure, we do not have to create 2^W number of branch nodes. Instead, we only need to create $W+1$ nodes, one for each unique value assignment.

When some integer variables have been fixed to some particular values at any step of the branch and bound algorithm, the symmetric structure will change. For example, if $y_i(w,r)$ is fixed to 1 and $y_i(w',r)$ is fixed to 0, i.e., an ADM is installed at node i on wavelength w, but not on wavelength w', then $y_{i'}(w,r)$ and $y_{i'}(w',r)$ variables are no longer equivalent to each other. Our branch and bound algorithm keeps track of what variables have been fixed in order to determine which set of variables to branch on to eliminate the symmetry.

The amount of extra information to keep track of is minimal. Assuming we have v number of variables, and assuming that we use depth first policy in the branch and bound algorithm, then there can at most be v number of branch and bound nodes at any time. For each branch and bound node, we only need to keep $O(W)$ amount of information on variable equivalency. Therefore, we need $O(vW)$ amount of extra storage, which is quite a small amount of overhead.

We refer interested readers to a technical report [17] for the full details of this technique.

IV. Heuristic algorithms

The ILP formulation can be solved exactly to get the optimal solution. Although we can now solve a reasonable size problem optimally with our techniques, it is still time consuming to solve large size problems. For these large problems, we propose efficient heuristic algorithms. Our improvements on the ILP approach allow us to extensively validate our proposed heuristic algorithms, and show that the heuristic algorithms can give near optimal results within a much shorter amount of time.

We note that our algorithms are the first heuristic algorithms to handle arbitrary traffic demand matrices, even in the case where only one line speed is available.

Our heuristic algorithms route traffic onto one ring at a time. When routing traffic onto a ring, we try to minimize the average cost per unit of routed traffic. Once some traffic has been routed on one ring, we proceed to the next ring to route the remaining traffic. Our algorithms are greedy in nature since we route traffic onto a ring based on the best average cost on that ring. We note that such a local optimal decision may not necessarily lead to the global optimal.

In the following, we first describe the CPD ratio which is the essential criteria we use to determine which traffic to route onto each ring. Then we describe how to determine the CPD ratios for both UPSR and BLSR. Lastly, we will describe the heuristic algorithm.

A. The CPD ratio

Our heuristic algorithms are based on the idea of reducing the *Cost Per unit Demand* (CPD) ratio. The CPD ratio has been used in [9][14] to derive a lower bound and in [22] to derive optimal solutions for uniform all-to-all traffic demands. However, our algorithms are the first to use the CPD ratio to guide the search for a solution. We first define the CPD ratio in the following.

Given a traffic matrix T, we want to route some traffic demands on a ring running at the rth line speed. For an arbitrary integer n ($n \leq N$), let $D(n,r)$ denote the maximum amount of traffic among any n nodes in T that could be routed on the ring running at the rth line speed, and let $T(n,r) = [t(n,r)_{sd}]$ denote the corresponding traffic matrix that is routed on the ring. By definition, we have $D(n,r) = \sum_{sd} t(n,r)_{sd}$. Furthermore, because there is no spatial capacity reuse in UPSR, and because a ring running at the rth line speed can support at most g_r traffic streams, $g_r \geq D(n,r)$. The CPD ratio $\rho(n,r)$ is then defined as the ratio between the ADM cost (n ADMs each costing c_r) and the amount of routed traffic ($D(n,r)$).

$$\rho(n,r) = \frac{c_r \times n}{D(n,r)}$$

The CPD ratio is essentially the minimum average cost to route some traffic streams on a ring running at the rth line speed using n nodes. Naturally, the lower the ratio, the lower the cost to route $D(n,r)$ traffic streams. Note that the CPD ratio is defined for a specific traffic matrix T. If the traffic matrix is different, the corresponding CPD ratios will be all different.

In general, there could be many traffic matrices $T(n,r)$ such that $D(n,r) = \sum_{sd} t(n,r)_{sd}$. Any of these traffic matrices could be routed on the ring to achieve the same CPD ratio. In our heuristic algorithms, we arbitrarily pick one of the traffic matrices.

We say a CPD ratio $\rho(n,r)$ *dominates* another CPD ratio $\rho(n',r')$ if $\rho(n,r) \leq \rho(n',r')$ and $D(n,r) \geq D(n',r')$. If no other CPD ratios dominate $\rho(n,r)$, we say $\rho(n,r)$ is a *dominant* CPD ratio, and we say $\rho(n,r)$ is a *non-dominant* CPD ratio if otherwise.

Our heuristic algorithms are greedy in nature. Therefore, there is no reason we should use a non-dominant CPD ratio to route traffic demands. For a non-dominant CPD ratio $\rho(n,r)$, we can always find another lower CPD ratio $\rho(n',r') < \rho(n,r)$ such that it can accommodate more traffic streams on the same ring, i.e., $D(n',r') \geq D(n,r)$. When we route traffic demands onto a ring, we first evaluate the CPD ratio $\rho(n,r)$ for all possible n and all possible r, and remove all non-dominant CPD ratios. We then sort the remaining dominant CPD ratios in increasing order, and number them based on their order as follows:

$$\rho(n_1,r_1) \leq \rho(n_2,r_2) \leq \rho(n_3,r_3) \leq \dots$$

Because we have removed all non-dominant CPD ratios, the corresponding amount of routed traffic should be in sorted order as well, i.e.,

$$D(n_1,r_1) \leq D(n_2,r_2) \leq D(n_3,r_3) \leq \dots$$

It is desirable to use $\rho(n_1,r_1)$ when routing traffic demands onto each ring because it has the lowest average cost. Unfortunately, this may not be possible because we only have a limited number of wavelengths available. Thus we need the higher CPD ratios $\rho(n_i,r_i)$ where $i > 1$ to pack more traffic onto a ring so that we can route all traffic demands on the W wavelengths.

B. Computing CPD ratio for UPSR

For UPSR, $\rho(n,r)$ can be easily computed by examining all $\binom{N}{n}$ combinations of choosing n nodes, and then summing up the traffic among these n nodes . To compute all CPD ratios at a particular line speed, we have to examine $\sum_{n=2}^{N}\binom{N}{n} < 2^N$ combinations. This may appear to be time consuming at first glance. Fortunately, in the SONET architecture, N is limited to at most 16, and examining all 2^{16} combinations only takes a very small amount of time.

C. Computing CPD ratio for BLSR

For BLSR, computing CPD ratios exactly is more difficult. Therefore, we resort to a heuristic algorithm. Recall that $\rho(n,r) = c_r \times n/D(n,r)$. Given an n and r, the only unknown value in the equation is $D(n,r)$. Therefore, computing $\rho(n,r)$ (finding the minimum CPD ratio) is equivalent to finding the maximum $D(n,r)$. For computing $D(n,r)$, i.e., the maximum amount of traffic one can fit on a ring at the rth line speed with n ADMs, we break down the task into two phases.

In the first phase, our heuristic first selects n out of the N nodes to place ADMs. Since our goal is to maximize the amount of routed traffic, the algorithm chooses the set of n nodes that have the most amount of unrouted traffic demands among themselves. Although it is possible that we can accommodate more traffic on the current wavelength if a different set of n nodes is chosen, we find that choosing the n nodes with the most amount of unrouted traffic works well in practice.

In the second phase, we try to fit as much traffic from the n nodes as possible on to the current wavelength. This phase works as follows. The algorithm first routes all unrouted traffic demands among these n nodes on to the current wavelength using the shortest path. If the current wavelength (at line speed r) can accommodate them all, then $D(n,r)$ is simply the sum of all unrouted traffic. Otherwise, the algorithm takes out traffic streams one by one, as described below, until all remaining traffic streams can be accommodated in the current wavelength.

Let l_i denote the *load* on the arc $(i,i+1)$, i.e., the total number of traffic streams on arc $(i,i+1)$. We say an arc $(i,i+1)$ is *overloaded* if $l_i > g_r$. Let o_k denote the *overload factor* of a traffic stream k, which is defined as the number of overloaded arcs that the traffic stream crosses. We say a traffic stream is *completely overloaded* if every arc it crosses is overloaded. Similarly, we say a traffic stream is *partially overloaded* if at least one arc it crosses is overloaded but no all arcs it crosses are overloaded.

We rank all completely overloaded traffic streams based on their overload factor and remove them from the current wavelength one by one. After all completely overloaded traffic streams are removed, we rank all partially overloaded traffic streams, breaking tie by the length of the path (longest path first), then remove the partially overloaded traffic streams one by one until the current wavelength can support the remaining traffic streams. It is important to note that the list of completely overloaded and partially overloaded traffic streams have to be recomputed after a traffic stream is removed. This is because the load on an arc could potentially change as a result of the removal.

Lastly, the algorithm randomly chooses one of the removed traffic streams, and see if it can be routed on the current wavelength using non-shortest path routing. If so, this traffic stream is added back in. This process continues until no more traffic streams can be accommodated. $D(n,r)$ is then the sum of all traffic that was routed.

D. The HCPDF heuristic algorithm

In general, we want to use the lowest dominant CPD ratio as much as possible, and only use the higher dominant ratios as necessary to pack more traffic onto a ring so that we can fit all traffic demands into the W wavelengths. When we are forced to use a higher dominant CPD ratio, it is beneficial to use it early on, i.e., use the higher CPD ratio first on ring 1, then on ring 2, and so on. This is because when we are routing traffic on the first few rings, there is more unrouted traffic, so that there are higher chances for a more optimized solution. Our heuristic algorithm tries to use the higher CPD ratios early, therefore, we call it the High CPD ratio First (HCPDF)is algorithm.

We maintain a set of pointers p_i, one for each ring. They are used to remember which CPD ratios should be used on a particular ring, i.e., on the ith ring, the p_ith dominant ratio should be used.

The HCPDF heuristic is shown in the following algorithm.

```
Algorithm HCPDF_heuristic
Input: T, N and W
Output: Cost of routing traffic demands
T using W wavelengths
1   Initialize
1.1 Set p_i = 1, ∀i
2   Route traffic demands based on the
    current p_i.
2.1 For each ring i from 1 to W, repeat
    step 2.2
2.2 Find the p_ith dominant CPD ratio, then
    route T(n_{p_i}, r_{p_i}) on the ith ring. Set
    T = T - T(n_{p_i}, r_{p_i}), then repeat step 2.2
    for other rings.
3   Check if feasible
3.1 If all traffic demands are routed in
    step 2, a solution has been found,
    return
3.2 Otherwise, find the largest i such
    that p_{i-1} > p_i, and increment p_i. If all
    p_i are equal, increment p_1. Then for
    all j > i, reset p_j = 1. Return to step
    2.
```

Initially, all p_i are 1, meaning that we will use the best CPD ratio for each ring. In the second step, we route all traffic demands based on the set of pointers p_i. We first compute all dominant CPD ratios for the first ring, then we choose the p_1th CPD ratio. In other words, we route $T(n_{p_1}, r_{p_1})$ traffic on the first ring. We repeat the process for other rings as well, i.e., we use the p_ith dominant CPD ratio when routing traffic on the ith ring. This process continues until all traffic demands are routed, or no more rings are left.

In the third step, we first determine if a feasible solution has been found. If all traffic demands have been routed, we return. Otherwise if there is still traffic to route, but there are no more rings left, we will pick a ring i and increase p_i by 1 so that we can route more traffic on that ring. Because more traffic is routed on ring i, there is a chance that we might be able to route the remaining traffic using lower CPD ratios. Therefore, we reset all $p_j = 1, \forall j > i$.

We pick a p_i to increase by 1 such that the relationship of $p_1 \geq p_2 \geq p_3 \ldots$ is always maintained. We do so by choosing the largest i such that $p_{i-1} > p_i$. If all p_i are equal, we choose the first pointer p_1 to increase by 1. The reason we maintain this relationship is because if we need to use a higher CPD ratio to route more traffic streams, we should do so as early as possible when there are more unrouted traffic, so that there are more chances for a more optimized solution.

Note that if a pointer p_i is changed in step 3.2, then the CPD ratios have to be recalculated in step 2.2 in the next iteration for any wavelength j such that $j > i$. This is because when a different CPD ratio is used on wavelength i, a different set of traffic is routed on wavelength i. Since the remaining traffic for the rest of the wavelengths will be different, the CPD ratios will be different.

V. Algorithms and techniques evaluation

To evaluate our proposed formulation and techniques, we consider three SONET networks supporting OC-3 traffic streams. All traffic demands are expressed as multiples of OC-3 circuits. The first network has only one line speed, e.g., OC-48. This is the network considered in [7] [9]. The second network has two line speeds available, e.g., OC-12 and OC-48. This is the network considered in [4] [16]. Lastly, we also consider a network with three line speeds, i.e., OC-3, OC-12 and OC-48. Even though we use OC-3, OC-12 and OC-48 in our examples, the same principle applies to higher line speeds such as OC-192 and OC-768.

We consider several traffic patterns for evaluation, including the uniform traffic pattern with 1 OC-3 demand between every node pair, the central traffic pattern with 1 OC-3 demand between every node and a central hub node and the random traffic pattern where the demands are randomly generated.

We use the same cost ratio assumption as in [4], i.e., OC-48 ADMs are 2.5 times more costly than OC-12 ADMs, and OC-12 ADMs are 2.5 time more costly than OC-3 ADMs. The capacity ratio is 4, i.e., OC-48 has 4 times the capacity of OC-12 and OC-12 has 4 times the capacity of OC-3. We normalize the cost of an OC-3 ADM to be 1. Therefore, an OC-12 ADM will cost 2.5 and an OC-48 ADM will cost 6.25. Even though we use these set of fixed ratios in our experiments, we note that our formulations and heuristic algorithms are general enough to handle any cost and capacity ratio.

We implemented an ILP solver that incorporates the techniques we proposed, and we also implemented the heuristic algorithms. We run all experiments on a Sun Blade 2000 workstation with UltraSPARC III+ 900Mhz CPUs. All computation time shown is in seconds in CPU time on the Sun workstations. We impose an upper limit on the computation time for the ILP solver to be 10000 seconds. If the ILP solver takes more than the time allowed, we terminate the computation and record the best solution it is able to find up to that point.

We only include limited set of computation comparisons. A full set of comparisons are available in [17].

A. Computational feasibility of the ILP approach

With our proposed techniques, we are able to solve a large set of problem instances optimally. For larger size problems, we can terminate the ILP solver early and use the best solution it can find up to that point. We find that the ILP solver can typically get optimal or near optimal solution very early on, but spends a lot of time searching the whole solution space for better solutions. In Fig. 3, we show the computation progress

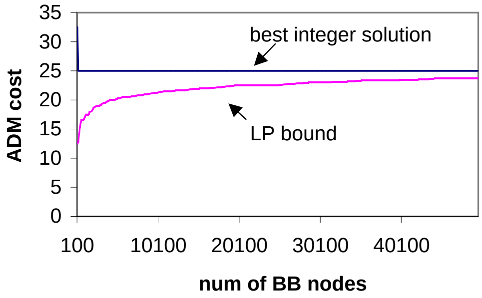

Fig. 3. Computation progress of the ILP solver. UPSR, uniform traffic, W=10, 5 nodes, 2 line speeds.

TABLE I
THE SIZE OF THE NETWORK THAT CAN BE SOLVED. UPSR, UNIFORM TRAFFIC, W=10.

	1 LS	3 LS	3 LS new
Solved exactly	7	4	9
Terminated early but still optimal	13	8	14

TABLE II
HEURISTIC ALGORITHM EVALUATION. UPSR, RANDOM TRAFFIC, W=10, 3 LINE SPEEDS.

# of nodes	ILP cost	HCPDF cost	HCPDF time
4	7.5*	8	0.04
5	15*	16	0.14
6	16.5*	18	0.21
7	18.5*	20	0.3
8	19.5*	20	0.37
9	23.5*	23.5	0.78
10	34*	34	1.68
11	36.5	37	2.31
12	51	51	3.16
13	57.5	59.5	3.99
14	77.5	83.25	8.68
15	95	85.75	9.43
16	97.5	103.25	10.7

of the ILP solver for a 5 node 2 line speeds (OC-12 and OC-48) UPSR network with 10 wavelengths supporting uniform traffic demands. The top line shows the best integer solution it has found so far, and the bottom line shows the best bound it has found through linear relaxation. These are shown against the number of branch and bound nodes that have been explored (x axis). The solver finds an initial solution with cost of 33, and quickly improves it to the optimal solution of 25. This all happens within the first 100 branch and bound nodes, so it is hard to see in the figure. Even though the optimal solution has been found very early on, the solver still takes a lot of time to improve the bound to confirm the optimality of the current solution. We believe this is a very common behavior for the traffic grooming problem. Therefore, by terminating the computation early, we are able to solve a even larger set of problem instances optimally using the ILP approach.

To see what problems can be solved optimally using the ILP approach, we consider uniform traffic pattern and a UPSR network with 10 wavelengths. Table I lists the size of the network in terms of the number of nodes that can be solved.

Under the column “1 LS”, we show what problems can be solved using the ILP formulation in [10][11] for the single line speed case. Under the column “3 LS”, we show what problems can be solved with 3 line speeds using the disaggregate formulation with multiple line speeds. Since using multiple line speeds complicates the formulation, we can only solve a much smaller problem. For problems that can be solved exactly, we can only solve for a network with 4 nodes instead of 7 nodes. For problems where we terminate the computation early but still get the optimal solutions, we can solve for a network with only 8 nodes instead of 13. Under the column “3 LS new”, we show the result of the new formulation and techniques. We now can solve for a network with 9 nodes exactly, and obtain optimal results for a network with 14 nodes if we terminate the computation early.

For central traffic pattern, a UPSR network with 10 wavelengths, 3 line speeds and up to 16 nodes can be solved exactly within 200 seconds of computation time. It is worth noting that the central traffic pattern is exactly the traffic pattern in the single hub architecture as considered in [7] [9] [15] [16]. It is remarkable that we can solve these problems exactly within few hundred seconds.

More detailed comparison of the aggregate formulation and the proposed techniques, including their individual contribution to the run time reduction, can be found in [17].

B. Heuristic algorithms evaluation

To evaluate the heuristic algorithms, we consider the random traffic pattern. In our experiments, we randomly generate one traffic matrix for a network with n nodes. The traffic matrix has $\max\{n(n-1)/8, n-1\}$ traffic demands and the size of each traffic demand is randomly chosen between 1 and 2 OC-3 circuits. Random traffic pattern is a good approximation of real life traffic pattern.

The results from the ILP solver and the heuristic algorithms assuming 10 wavelengths and 3 line speeds for UPSR are shown in Table II. We put an asterisk next to the ILP results when the ILP solver finishes the computation before the allowed time, so those results are known to be optimal. As shown, the HCPDF algorithm can produce solutions that are very close to those from the ILP solver. In one case ($n = 15$), the HCPDF algorithm even produced better result. Also, the HCPDF algorithm takes much less computation time. Even for the largest problem ($n = 16$), it takes only 10 seconds. We see similar results for BLSR networks.

VI. NUMERICAL RESULTS

In this section, we use the ILP formulation and the heuristic algorithm to study the cost benefit of traffic switching and the cost benefit of using multiple line speeds.

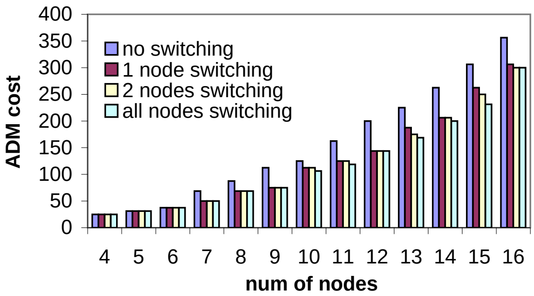

Fig. 4. Cost benefit of traffic switching. UPSR, uniform traffic, one line speed, W=10.

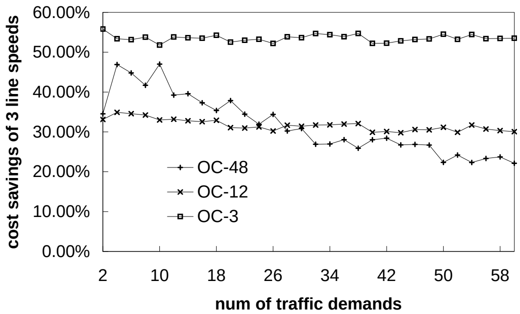

Fig. 5. Cost savings. UPSR, 16 nodes, 1/3 demands of size OC-3, OC-12 and OC-48.

A. *Traffic switching*

Traffic switching can lower the network cost as shown in Fig. 2. The aggregate formulation allows traffic to switch across different rings at a selected subset of nodes. Thus, using our novel formulation, we are able to study the cost benefit of traffic switching. We consider a network with only one line speed (OC-48) and 10 wavelengths available. The cost savings for uniform traffic in UPSR when one node is allowed to switch traffic, when two nodes are allowed to switch traffic and when all nodes are allowed to switch traffic are shown in Fig. 4. For comparison, we also show the case when no traffic switching is allowed. The results are obtained using the ILP solver. As shown, the cost savings when one node can switch traffic are up to 33%. Having more nodes switching only reduces the cost further marginally. It suggests that just setting up switching at one node may be good enough. This observation is consistent with that in [20], where they found that the number of switching nodes should be approximately the same as the number of wavelengths of traffic generated by each node.

For random traffic patterns, we see similar cost savings. However, for central traffic pattern, we are not able to see any cost savings when we introduce switching nodes. This is because traffic switching only helps to reduce cost when an ADM has to be installed at the switching node for other traffic demands anyway. This is not true for central traffic pattern.

For BLSR, we also see large reductions in cost from switching traffic, however, compared to UPSR, the amount of saving is smaller.

B. *Multiple line speeds*

Multiple line speeds can lower cost in two different cases. The first case happens when no lower line speed is available. Therefore, one is forced to use the more expensive ADMs even if the traffic to support is small. If lower and less costly line speed is available, we can use that instead to lower the overall cost. For example, if the only line speed available is OC-48 and the traffic demand between two nodes is only one OC-3 circuit, then we can use OC-3 ADMs to lower the cost and still support the same traffic.

The second case happens when no higher line speed is available. Therefore, one can not achieve enough aggregation to realize the economy of scale. For example, if the only line speed available is OC-12 and the traffic demand between two nodes is one OC-48 worth of traffic, then we have to used 8 OC-12 ADMs. But, if OC-48 line speed is also available, we can use only 2 OC-48 ADMs instead. In general, 2 OC-48 ADMs will cost less than 8 OC-12 ADMs.

We now try to characterize the conditions under which multiple line speeds can lower cost. To make the experiments statistically significant, we run each data point 10 times and then take the average, i.e., for each data point used in this section, we randomly generated 10 different instances and then took the average of their cost. We used the HCPDF algorithm for all experiments because it can produce very good results using very little computation time. For these experiments, we specified a large W in order to guarantee that a feasible solution can be found. This will also result in the lowest cost possible with multiple line speeds, because, otherwise, we may be forced to pack traffic using higher cost ADMs.

We first consider a 16-node UPSR network. The traffic demands are randomly generated, where 1/3 of them require one OC-3 circuit, 1/3 of them require one OC-12 circuit and the remaining 1/3 require one OC-48 circuit. In Fig. 5, we plot the cost savings of using 3 line speeds (OC-3, OC-12 and OC-48) versus using a fixed line speed (OC-3 or OC-12 or OC-48). If only OC-3 line speed is available, the network cost is very high because many ADMs are needed to support one OC-48 traffic. If only OC-12 line speed is available, the network cost is lower because come traffic aggregation can be achieved. If only OC-48 line speed is available, the network cost is high when the number of traffic demands is small because an OC-48 ADM may have to be installed to accommodate an OC-3 traffic. When the number of traffic demands increases, the network cost decreases because it becomes possible to aggregate more OC-3 and OC-12 traffic.

As shown in the figure, no matter which single line speed one chooses, the network cost will always be significantly higher compared to the case where multiple line speeds are used.

We next consider a 16-node UPSR network with 60 randomly generated traffic demands. The size of each traffic demand is randomly generated between 1 and a fixed number S, therefore, the average size of a demand is $S/2$. The cost savings of using 3 line speeds as compared to using 1 single line speed are shown in Fig. 6. When S is small, OC-3 costs the least; and when S is big, OC-48 costs the least. When S is in between, using only OC-12 line speed gives the lowest network cost. Again, no matter which fixed line speed is chosen, there is a large region of S where roughly 20% cost

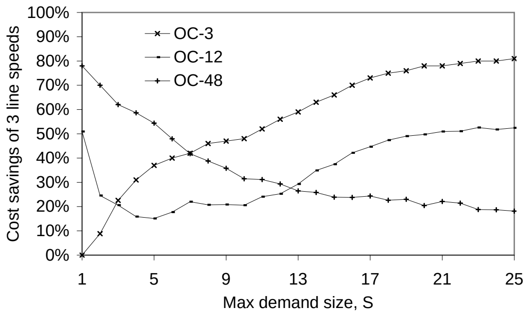

Fig. 6. Cost savings. UPSR, 16 nodes, 60 demands, each demand ranges from 1 to S.

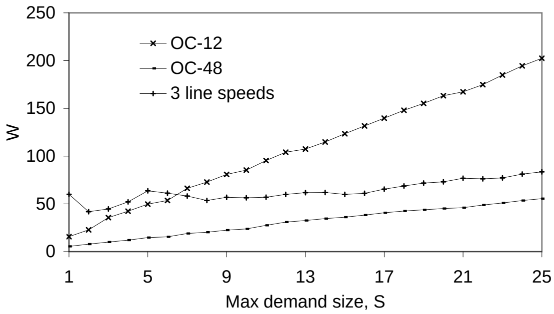

Fig. 7. Number of wavelengths used. UPSR, 16 nodes, 60 demands, each demand ranges from 1 to S.

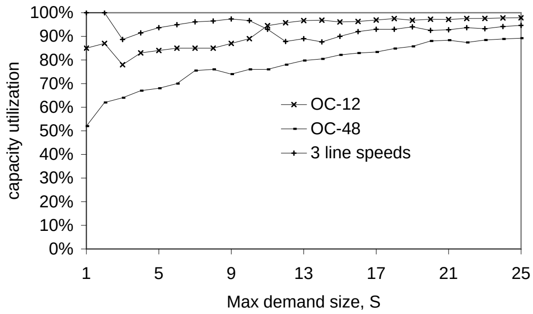

Fig. 8. Utilization ratio. UPSR, 16 nodes, 60 demands, each demand ranges from 1 to S.

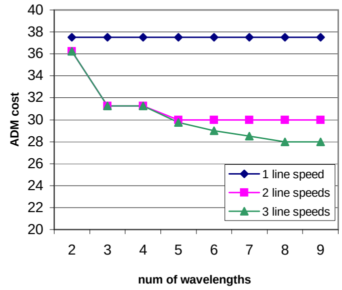

Fig. 9. Reducing ADM cost using more wavelengths. 5 nodes, random traffic, 8 demands, demand size ranges from 1 to 4 OC-3 circuits.

savings can be achieved by using 3 line speeds. If the fixed line speed is not chosen intelligently, much higher cost savings could be achieved by using multiple line speeds.

We also show the number of wavelengths used in Fig. 7. As expected, OC-48 uses the least number of wavelengths and OC-12 tends to use many more wavelengths especially when S is large. In comparison, multiple line speeds only uses slightly more wavelengths than OC-48. Since many wavelengths (at least 40) are available in a typical DWDM system, this seems to be a small price to pay for the large reduction in the network cost.

When S is small (≤ 6), multiple line speeds use more wavelengths than OC-12. This shows that the cost savings are a result of the first case, where introducing lower line speed (OC-3) can lower cost. When S is large (≥ 7), multiple line speeds use fewer wavelengths than OC-12. This shows that the cost savings are a result of the second case, where introducing higher line speed (OC-48) can lower cost.

Multiple line speeds can also increase capacity utilization. The capacity utilization is defined as the ratio between the total amount of traffic routed and the sum of capacity of each wavelength. As shown in Fig. 8, multiple line speeds has much high utilization ratio than OC-48 because aggregation is not always possible. Although multiple line speeds has slightly worst utilization ratio than OC-12 when S is large, we note that OC-12 costs significant more in the same region (Fig. 6).

We observe similar cost savings in BLSR networks. For brevity, the results are not shown.

The results we have shown (Fig. 5-8) assume that a large number of wavelengths are available. When the number of wavelengths available is small, we are forced to use the more costly ADMs at a higher line speed to route all traffic demands. Therefore, the resulting network cost will be higher. We study the cost savings as a function of the number of wavelengths for a 5 nodes network with 8 randomly generated traffic demands ranging from 1 to 4 OC-3 circuits. The result obtained from the ILP solver is shown in Fig. 9. The total ADM cost decreases as more wavelengths become available. However, no further cost savings could be achieved beyond a certain point. For 2 line speeds, there are no cost savings beyond 5 wavelengths, and for 3 line speeds, there are no cost savings beyond 8 wavelengths. We also note that most of the cost savings could be achieved using only 2 line speeds. Adding the third line speed only reduces the cost further marginally.

So far we have assumed that the cost ratio is exactly 2.5. However, in practice, the ratio could vary a lot. We now look at how the cost ratio can affect the savings. We consider a 10 nodes network with 10 traffic demands, each demand ranges from 1 to 16 in size. We vary the cost ratio from 2 to 3 and show the result in Fig. 10.

As cost ratio increases, OC-48 cost increases. This is because multiple line speed has an increasing chance finding lower cost alternatives using lower line speeds. However, as cost ratio increases, OC-12 cost decreases. This is because the cost savings as a result of aggregation tend to diminish as the cost ratio increases.

C. UPSR and BLSR cost comparison

It is proved in [15] that BLSR/2 will never cost more than UPSR in the single hub architecture. In the following, we will prove that BLSR/2 will never cost more than UPSR, regardless of the traffic pattern.

Theorem 2: Given any set of traffic demands and given any set of supported line speeds, BLSR/2 costs no more than UPSR.

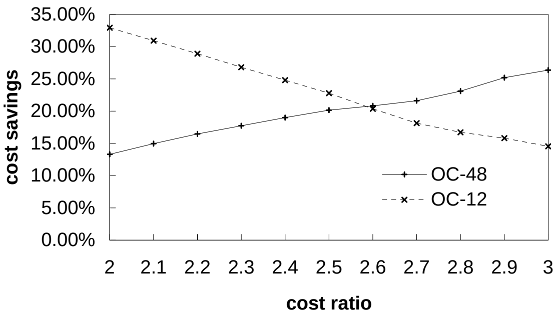

Fig. 10. Cost savings as a function of the cost ratio. 10 nodes, 10 demands.

Proof: Consider the optimal traffic grooming solution in UPSR, we can construct an equivalent BLSR/2 grooming solution as follows. Let w be a ring in the UPSR solution, and let f_k^w be the portion of the kth traffic demand that is carried on wavelength w. We assign wavelength w in BLSR to use the same line speed, and route $f_k^w/2$ traffic in the clockwise direction and route the rest in the counter-clockwise direction on wavelength w. Since the traffic on wavelength w in UPSR is less than the ring capacity, the traffic on wavelength w in BLSR will always be less than 50% of the capacity, guaranteeing that enough capacity will be reserved in BLSR for protection. Therefore, the solution thus constructed is valid and it costs the same as the UPSR grooming solution. ■

Although we prove that BLSR/2 will never cost more than UPSR, it is unknown how much cost savings BLSR/2 could achieve. In Table III, we show the cost comparison between UPSR and BLSR/2 when supporting uniform traffic (1 OC-3 demand between every node pair) in a network with 3 wavelengths. The results are derived by solving the ILP formulations directly using the ILP solver.

We have to install at least one ADM at each node to support uniform traffic. With a single line speed (OC-48), when the total traffic demand is low, one ADM per node is sufficient. Therefore, both BLSR/2 and UPSR will cost the same. As traffic demand goes up, UPSR has to use more wavelengths, and therefore, has to install more ADMs. However, BLSR/2 can more efficiently utilize the capacity and need less wavelengths. Therefore, fewer ADMs are needed, which translates into significant cost savings. The cost savings are up to 50%.

With 2 line speeds (OC-12 and OC-48), BLSR/2 can lower cost even when the total traffic demand is low. This is because BLSR/2 can choose a lower cost line speed and still be able to support all traffic demands. As traffic demand goes up, the cost savings are more significant compared to the single line speed case. Sometimes, the cost savings are more than 50%.

D. Fixed shortest path routing vs. routing in both directions in BLSR

In BLSR, we have a choice of routing traffic either along the shortest path or along the other arc between the source and destination nodes. Routing along the shortest path simplifies the routing decision, as a result, the complexity of the ILP formulations will be greatly reduced and an optimal solution can be found in a much shorter amount of time. In this section, we quantify the tradeoffs of using only fixed shortest path routing.

TABLE III

COST COMPARISON BETWEEN UPSR AND BLSR/2. UNIFORM TRAFFIC, W=3

# of nodes	1 line speed (OC-48)		2 line speeds (OC-12 & OC-48)	
	UPSR	BLSR/2	UPSR	BLSR/2
4	25	25	17.5	10
5	31.25	31.25	25	12.5
6	37.5	37.5	37.5	22.5
7	68.75	43.75	57.5	30
8	87.5	50	85	40
9	112.5	56.25	97.5	45
10	125	62.5	125	62.5

TABLE IV

COST AND RUN TIME COMPARISON OF SHORTEST PATH ROUTING IN BLSR. RANDOM TRAFFIC, W=3, 3 LINE SPEEDS

# of nodes	cost		runtime	
	BD	SP	BD	SP
4	5	5	0.13	0.02
5	10	12.5	0.49	0.09
6	9.5	14.5	0.36	0.16
7	14	14.5	1.51	0.07
8	17	17.5	1.83	0.2
9	18.5	21	5.11	0.51
10	25.5	25.5	138	1.45
11	30.5	30.5	196	1.97
12	37.5	55	1791	140
13	40	42.5	13032	33
14	71.25	76.25	3380	37.32
15	68.25	75.75	18721	215
16	88.75	95	6114	2197

We consider networks with varying number of nodes and supporting random traffic pattern. For networks with N nodes, we randomly generate $\max\{N(N-1)/8, N-1\}$ traffic demands, each demand is either 1 or 2 OC-3 circuits. The cost and run time comparison when 3 line speeds (OC-3, OC-12 and OC-48) are available is shown in Table IV. The columns under "BD" show results when both routes can be used and the columns under "SP" show results when only the shortest path route can be used.

We note that in most cases, shortest path routing will increase cost. In a few cases, the cost increase could be more than 50%. It suggests that using non-shortest path routing for random traffic pattern may be very important. Although the cost is higher, shortest path routing simplifies the problem formulation, therefore, significantly reduces the run time. On average, it takes one tenth of the time to compute an optimal solution using only shortest path routing.

We also studied uniform traffic pattern. For all the cases that we are able to solve exactly, we find the cost remains the same even if we use only the shortest path routing. Even though we can not prove it, we conjecture that this is true in general for uniform traffic demands. This is because the traffic is very symmetric in the uniform traffic pattern so that the shortest path routing can still average out the traffic flowing on each link. It suggests that if the traffic demands are uniform or close to be uniform, then using only the shortest path routing would not increase the cost too much, and at the same time, can greatly reduce the computation time.

VII. Conclusion

We consider the traffic grooming problem in WDM/SONET UPSR rings with multiple line speeds. We study two different approaches to solve the traffic grooming problem. To solve the problem optimally, we propose new aggregate ILP formulations and several new techniques that can reduce the computation time, thereby, allowing many problem instances to be solved exactly. For larger problem instances, we propose an efficient heuristic algorithm that can derive comparable solutions within a much shorter amount of time.

We show that having more line speeds could greatly reduce the network cost, especially when the traffic demands are non-uniform. We quantify the region where the cost savings are most pronounced. We also show that multiple line speeds can increase capacity utilization. Our novel aggregate formulations allow traffic to switch across rings. Using the formulation, we show that traffic switching can reduce cost further and that most of the cost savings could be achieved by allowing only one node to switch traffic.

References

[1] H. Liu and F. Tobagi, "Traffic grooming in WDM SONET UPSR rings with multiple line speeds," in *Proc. Infocom 2005*.

[2] ——, "A novel efficient technique for traffic grooming in WDM/SONET with multiple line speeds," in *Proc. ICC 2004*.

[3] ——, "Traffic grooming in WDM SONET BLSR rings with multiple line speeds," in *Proc. IEEE Global Telecommunications Conference (Globecom) 2005*.

[4] O. Gerstel, P. Lin, and G. Sasaki, "Combined WDM and SONET network design," in *Proc. INFOCOM 1999*.

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

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

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

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

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

[10] J.-Q. Hu, "Traffic grooming in wavelength-division-multiplexing ring networks: a linear programming solution," *J. Optical Networking*, vol. 1, no. 11, pp. 397–408, Nov. 2002.

[11] A. Sutter, F. Vanderbeck, and L. Wolsey, "Optimal placement of add/drop multiplexers: heuristic and exact algorithms," *Operations Research*, vol. 46, pp. 719–728, 1998.

[12] J. Wang, W. Cho, V. R. Vemuri, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM ring networks: ILP formulations and single-hop and multihop connections," *J. Lightwave Technol.*, vol. 19, no. 11, pp. 1645–1653, Nov. 2001.

[13] T. Y. Chow and P. J. Lin, "The ring grooming problem," *Discrete Applied Math*, vol. 0, no. 0, pp. 1–12, Jan. 2000.

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

[15] X. Y. Li, L. Liu, P. J. Wan, and O. Frieder, "Practical traffic grooming scheme for single-hub SONET/WDM rings," in *Proc. 25th IEEE Conf. on Local Computer Networks 2000*.

[16] X. Li, P. Wan, and L. Liu, "Select line speeds for single-hub SONET/WDM ring networks," in *Proc. ICC 2000*, pp. 495–499.

[17] H. Liu and F. Tobagi, "Traffic grooming in WDM SONET rings supporting multiple line speeds," Stanford University, Tech. Rep., 2004.

[18] D. Bienstock and O. Gunluk, "Computational experience with a difficult mixed-integer multi-commodity flow problem," *Mathemat. Program.*, vol. 68, pp. 213–237, 1995.

[19] R. M. Krishnaswamy and K. N. Sivarajan, "Design of logical topologies: a linear formulation for wavelength-routed optical networks with no wavelength changers," *IEEE/ACM Trans. Networking*, vol. 9, no. 2, Apr. 2001.

[20] R. Berry and E. Modiano, "The role of switching in reducing the number of electronic ports in WDM networks," *IEEE J. Sel. Areas Commun.*, vol. 22, no. 8, pp. 1396–1405, Oct. 2004.

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

[22] J.-Q. Hu, "Optimal traffic grooming for wavelength-division-multiplexing rings with all-to-all uniform traffic," *J. Optical Networking*, vol. 1, no. 1, pp. 32–42, Jan. 2002.

Huan Liu received his Ph.D. degree in Electrical Engineering from Stanford University in 2006. He is currently a Research Manager with Accenture Technology Labs in Palo Alto, California. His research interests include network design, optical switching and load balancing.

Fouad A. Tobagi (M'77-SM'83-F'85) received the Eng. degree from the École Centrale des Arts et Manufactures, Paris, France, in 1970 and the M.S. and Ph.D. degrees in computer science from the University of California, Los Angeles, in 1971 and 1974, respectively.

From 1974 to 1978, he was a Research Staff Project Manager with the ARPA project in the Computer Science Department, University of California, Los Angeles, engaged in research in packet radio networks, including protocol design, and analysis and measurements of packet radio networks. In June 1978, he joined the faculty of the School of Engineering, Stanford University, Stanford, CA, where he is currently a Professor of electrical engineering and computer science. In 1991, he co-founded Starlight Networks, Inc., a venture concerned with multimedia networking, and served as Chief Technical Officer until August 1998. He was a co-editor of *Advances in Local Area Networks* (New York: IEEE Press, 1987). He was editor for Packet Radio and Satellite Networks of the *Journal of Telecommunications Networks* from 1981 to 1985, and he is currently serving as editor for a number of journals in high speed networks, wireless networks, multimedia, and optical communications. His research interests have included packet radio and satellite networks, local area networks, fast packet switching, multimedia networking and networked video services, and multimedia applications. His current interests include voice and video communication over the Internet, wireless and mobile networks, network design and provisioning, and network resource management.

Dr. Tobagi is a Fellow of the IEEE for his contributions in computer communications and local area networks. He is the recipient of the 1981 Leonard G. Abraham Prize Paper Award in the field of communications systems for his paper "Multiaccess Protocols in Packet Communications Networks" and co-winner of the IEEE Communications Society 1984 Magazine Prize Paper Award for the paper "Packet Radio and Satellite Networks." He was a corecipient of the 1998 Kuwait Prize in the field of applied sciences. He has served as Associate Editor for Computer Communications of the *IEEE Transactions on Communications* from 1984 to 1986, co-editor of the special issue on local area networks of the *IEEE Journal on Selected Areas in Communications* (November 1983), co-editor of the special issue on packet radio networks of the *Proceedings of the IEEE* (January 1987), and co-editor of the special issue on large scale ATM switching systems for B-ISDN of the *IEEE Journal on Selected Areas in Communications* (October 1991). He is a member of the Association for Computing Machinery (ACM) and served as an ACM National Lecturer from 1982 to 1983.