9-1-2003

Dynamic Traffic Grooming Algorithms for Reconfigurable SONET Over WDM Networks

Shu Zhang
IEEE

Byrav Ramamurthy
University of Nebraska - Lincoln, bramamurthy2@unl.edu

Dynamic Traffic Grooming Algorithms for Reconfigurable SONET Over WDM Networks

Shu Zhang and Byrav Ramamurthy, *Associate Member, IEEE*

***Abstract*—The emergence of wavelength-division multiplexing (WDM) technology provides the capability for increasing the bandwidth of synchronous optical network (SONET) rings by grooming low-speed traffic streams onto different high-speed wavelength channels. Since the cost of SONET add–drop multiplexers (SADM) at each node dominates the total cost of these networks, how to assign the wavelength, groom the traffic, and bypass the traffic through the intermediate nodes has received a lot of attention from researchers recently. Moreover, the traffic pattern of the optical network changes from time to time. How to develop dynamic reconfiguration algorithms for traffic grooming is an important issue. In this paper, two cases (best fit and full fit) for handling reconfigurable SONET over WDM networks are proposed. For each approach, an integer linear programming model and heuristic algorithms (TS-1 and TS-2, based on the tabu search method) are given. The results demonstrate that the TS-1 algorithm can yield better solutions but has a greater running time than the greedy algorithm for the best fit case. For the full fit case, the tabu search heuristic yields competitive results compared with an earlier simulated annealing based method and it is more stable for the dynamic case.**

***Index Terms*—Optical network, reconfiguration, SONET add–drop multiplexers (SADM), synchronous optical network (SONET), traffic grooming, wavelength-division multiplexing (WDM), WDM add–drop multiplexer (WADM).**

I. Introduction

MOST OF today's optical networks are built on synchronous optical network (SONET) rings [1]. Using wavelength-division multiplexing (WDM) technology, multiple rings can be supported on a single fiber ring [3]. In this architecture, each wavelength independently carries a SONET ring. Each SONET ring can further support multiple low-speed streams (e.g. an OC-48 SONET ring can support 4 OC-12 or 16 OC-3 streams at the same time). At each node a WDM add/drop multiplexer (WADM) adds and drops or bypasses traffic on any wavelength. At each node, there are SONET add/drop multiplexers (SADM) on each wavelength to add/drop low-speed streams. So the number of SADMs per node will increase linearly with the number of wavelengths that a single fiber ring can carry. The cost of SADMs will dominate the total cost of the optical network. But in fact, it is not necessary for each node to be equipped with SADMs on each wavelength. There is a need for an SADM on a wavelength at a node only if there is traffic terminating at this node on this wavelength. For example (see Fig. 1), if there is a traffic stream from node A to node B through node C on wavelength λ_1, there should be an SADM on λ_1 at node A and node B. The traffic can bypass node C without add/drop capabilities for traffic streams on λ_1. So node C need not be equipped with an SADM on λ_1. The problem of combining different low-speed traffic streams into high-speed traffic streams in such a way that the number of SADMs is minimized is called traffic grooming. Several studies have been done on traffic grooming [2], [5], [6].

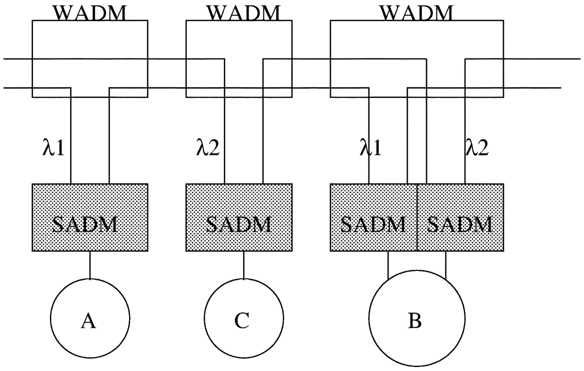

Fig. 1. SONET/WDM with bypass traffic at node C on λ_1.

Here, is an example to show that traffic grooming can reduce the number of SADMs [3]. Consider a ring network with five nodes. Each wavelength could carry two traffic streams. The traffic pattern for this example is bidirectional uniform traffic. That is, there exists the same amount of traffic in both directions for each node pair. The traffic matrix of this example is given as follows:

$$T = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

Fig. 2 shows the SONET ring without traffic grooming. The traffic assignment is shown in Table I. 1 ↔ 2 in Table I and II means that the traffic between nodes 1 and 2 is groomed on the wavelength λ_1. If there is a traffic stream from node 1 to node 3 assigned on a certain wavelength, we say there is a virtual connection setting up from node 1 to node 3 on this wavelength. The total number of SADMs required is 19. Fig. 3 shows the SONET ring with traffic grooming. The traffic assignment is shown in Table II. The total number of SADMs is 15. In fact,

Manuscript received August, 2002; revised March, 2003. This work was supported in part by National Science Foundation (NSF) under Grant ANI-0074121 and Grant EPS-0091900. This paper appeared in part at the *Proc. IEEE Global Communication (GLOBECOM) Conf.*, Taipei, Taiwan, Nov. 2002.

S. Zhang is with the Microsoft Corporation, Redmond, WA 98052 USA (e-mail: shuzhang@microsoft.com).

B. Ramamurthy is with the Department of Computer Science and Engineering, University of Nebraska, Lincoln, NE 68588 USA (e-mail: byrav@cse.unl.edu).

Digital Object Identifier 10.1109/JSAC.2003.815844

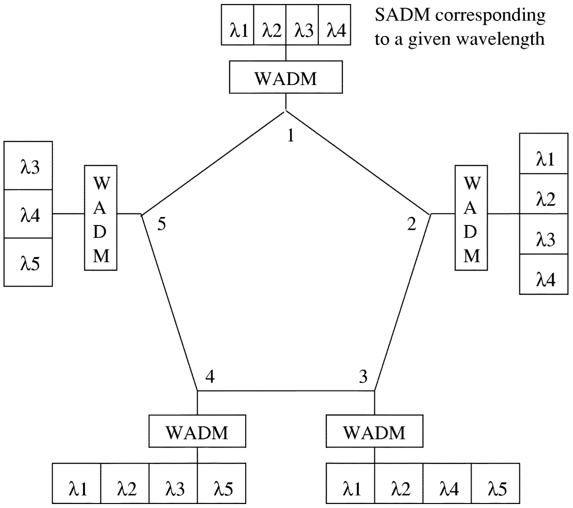

Fig. 2. SONET ring without traffic grooming.

TABLE I
TRAFFIC ASSIGNMENT WITHOUT TRAFFIC GROOMING

Wavelength	Traffic
λ_1	$(1 \leftrightarrow 2), (3 \leftrightarrow 4)$
λ_2	$(1 \leftrightarrow 3), (2 \leftrightarrow 4)$
λ_3	$(1 \leftrightarrow 4), (2 \leftrightarrow 5)$
λ_4	$(1 \leftrightarrow 5), (2 \leftrightarrow 3)$
λ_5	$(3 \leftrightarrow 5), (4 \leftrightarrow 5)$

TABLE II
TRAFFIC ASSIGNMENT WITH TRAFFIC GROOMING

Wavelength	Traffic
λ_1	$(1 \leftrightarrow 2), (1 \leftrightarrow 3)$
λ_2	$(1 \leftrightarrow 4), (1 \leftrightarrow 5)$
λ_3	$(2 \leftrightarrow 4), (2 \leftrightarrow 3)$
λ_4	$(4 \leftrightarrow 5), (2 \leftrightarrow 5)$
λ_5	$(3 \leftrightarrow 5), (3 \leftrightarrow 4)$

if we do not know the traffic pattern and suppose there is an SADM at each node on each wavelength, the total number of SADMs is $5 \times 5 = 25$. Thus, traffic grooming can reduce the number of SADMs greatly.

However, most algorithms assume the traffic matrix to be static; actually, the traffic pattern over SONET rings changes from time to time. In [7], dynamic traffic is described by a multiple set of the traffic matrices and a traffic grooming solution is proposed to meet the multiset instead of a single matrix. However, it is common that a change of traffic matrix happens after the configuration is established. In this paper, we consider the dynamic traffic grooming problem incorporating reconfiguration. That is, based on the current wavelength assignment, when the traffic pattern of the networks changes, we propose a dynamic traffic grooming algorithm to reconfigure the wavelength assignment according to the new traffic pattern without disrupting the old traffic assignment. Two cases, best fit and full fit are studied. Two heuristic algorithms, greedy and tabu search (TS-1) are presented for the best fit approach. A two-phase algorithm based on tabu search (TS-2) is presented for the full fit approach. The static traffic grooming problem, on which many studies have been done so far, is a special case of the *dynamic* problem (specially, the full fit case). The simulation results illustrate that the tabu search algorithm can yield better solutions but takes more running time than greedy algorithms for the best fit case. The tabu-search algorithm for the full fit case is more stable compared to the earlier simulated annealing heuristic [5]. Some results for the static grooming problem are found to be better than earlier results in [5].

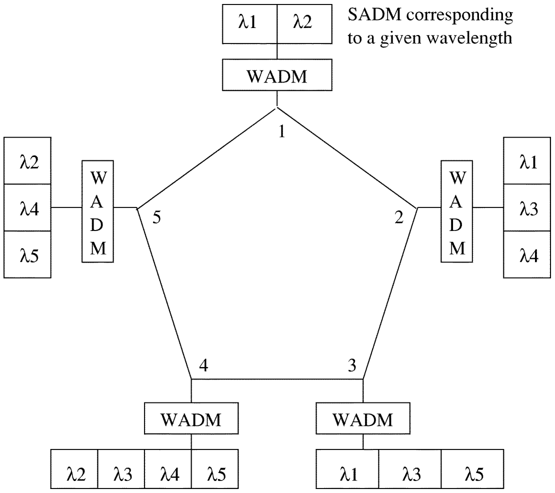

Fig. 3. SONET ring with traffic grooming.

II. PROBLEM DEFINITION: DYNAMIC TRAFFIC GROOMING

In this section, integer linear programming (ILP) models are proposed for dynamic traffic grooming. They are based on the static models proposed in [5]. There are two assumptions.

1) The old traffic matrix is known. By using a heuristic algorithm or integer linear programming solver [5], [6], the current configuration of the network is obtained according to the old matrix. This corresponds to the initial assignment of traffic to wavelengths in the network.
2) The traffic matrix changes. The new matrix is different from the original matrix. The objective is to disrupt as few current connections as possible and fit the new traffic requests in.

The following are the notations that we will use later.

1) There are N nodes numbered $0, 1, 2, \ldots, N-1$ in the SONET ring.
2) W is the number of wavelengths in the original traffic grooming matrix.
3) The granularity of the traffic g is defined as

$$g = \frac{\text{channel capacity}}{\text{base bandwidth rate}}$$

e.g., for an OC-48 channel carrying several OC-3 streams, the granularity $g = (OC{-}48)/(OC{-}3) = 16$. It is the number of circles[1] C each wavelength can carry.

4) Original traffic matrix $T[n \times n]$.

The traffic amount from node i to node j $(i, j = 1, \ldots, N-1, i \neq j)$ on the ring is denoted by t_{ij} (entry of matrix T at row i and column j) and is always a multiple of the base bandwidth rate.

[1] The circle here refers to the circle built by the algorithm in [6].

5) $T'[n \times n]$ is the new traffic matrix.
6) V_{ij}^{cw} represents the number of virtual connections for node pair (i,j) on wavelength w in circle c according to the original matrix. It is known in the reconfiguration problem.
7) $V_{ij}'^{cw}$ represents the number of new virtual connections from i to j on wavelength w of circle c according to the new traffic matrix.
8) V_{ij}^{cw+} represents the number of new virtual connections from i to j on new wavelength w^{+} of circle c according to the new traffic matrix (applies to full fit case only).
9) ADM_i^w represents the number of ADMs on node i on wavelength w for the original matrix.
10) $\mathrm{ADM}_i'^{w}$ represents the number of additional ADMs on node i on wavelength w (applies to full fit case only).
11) ADM_i^{w+} represents the number of ADMs on node i on new wavelength w^{+} (applies to full fit case only).

We assume that for the original traffic matrix, a solution has been found (e.g. using the algorithm in [5] and [6]). So we know the number of wavelengths W and the virtual connections for each pair (i,j). In addition, we also know the the number of SADMs and their positions.

Without special indication, unidirectional rings are assumed in the following problem descriptions. For bidirectional rings, constraints for both directions should be satisfied. Here are two cases to consider in the problem.

A. Best Fit Case

In best fit case, we try to place as much new traffic as possible without increasing the number of SADMs

$$\text{Maximize: } \sum_i \sum_j \sum_c \sum_w V_{ij}'^{cw} (\text{objective function}).$$

The objective is to maximize the traffic amount according to the new traffic demand. The following constraints are assumed.

1) Traffic constraint

$$\sum_w \sum_c V_{ij}'^{cw} + \sum_w \sum_c V_{ij}^{cw} \le t_{ij}'.$$

The traffic constraint indicates that the total number of virtual connections from i to j should be less than the new demand traffic matrix of entry t_{ij}'.

2) Circle capacity constraint

$$\sum_{(i,j)e\in(i,j)} (V_{ij}'^{cw} + V_{ij}^{cw}) \le 1.$$

The circle capacity constraint requires that no two connections can share a link on a circle.

3) Transmitter constraint

$$\sum_c \sum_j V_{ij}'^{cw} + \sum_c \sum_j V_{ij}^{cw} \le g \cdot \mathrm{ADM}_i^w.$$

The transmitter constraint requires that the total number of virtual connections starting at node i should be less than the transmitter capacity of SADM at this node on wavelength w.

4) Receiver constraint

$$\sum_c \sum_i V_{ij}'^{cw} + \sum_c \sum_i V_{ij}^{cw} \le g \cdot \mathrm{ADM}_w^j.$$

The receiver constraint requires that the total number of virtual connections terminating at node j should be less than the receiver capacity of SADMs at this node on wavelength w.

$V_{ij}^{cw'} \in \{0,1,-1\}$. $ADM_j^w \in \{0,1\}$. $V_{ij}^{cw'} = -1$ only if $V_{ij}^{cw} = 1$ and there is no connection between i and j on wavelength w of circle c for the new configuration any more.

B. Full Fit Case

In full fit case, we add the minimum number of SADM to satisfy all the new traffic

Minimize:

$$\sum_i \sum_w \mathrm{ADM}_i'^{w} + \delta \sum_i \sum_{w+} \mathrm{ADM}_i^{w+} (\text{objective function}).$$

The objective is to fit all the traffic with the minimum number of SADMs added. δ in the objective function is the weight parameter representing the cost of adding more wavelengths. Because usually adding new wavelengths will cost more than adding more SADMs on existing wavelengths, δ is supposed to be no less than one. The following constraints are assumed.

1) Traffic constraint

$$\sum_w \sum_c (V_{ij}^{cw} + V_{ij}'^{cw}) + \sum_{w+} \sum_c V_{ij}^{cw+} = t_{ij}'.$$

The traffic constraint indicates that the total number of virtual connections from node i to node j should equal the traffic demand from node i to j.

2) Circle capacity constraint

$$\sum_{(i,j)e\in(i,j)} (V_{ij}^{cw} + V_{ij}'^{cw}) \le 1$$
$$\sum_{(i,j)e\in(i,j)} V_{ij}^{cw+} \le 1.$$

The circle capacity constraint requires that no two connections can share a single link on a circle.

3) Transmitter constraint

$$\sum_c \sum_j \left(V_{ij}^{cw} + V_{ij}'^{cw}\right) \le g \cdot \left(\mathrm{ADM}_i^w + \mathrm{ADM}_j'^{w}\right)$$
$$\sum_c \sum_j V_{ij}^{cw+} \le g \cdot \mathrm{ADM}_i^{w+}.$$

The transmitter constraint requires that the total number of virtual connections should be less than the transmission capacity of the equipment at this node.

4) Receiver constraint

$$\sum_c \sum_i \left(V_{ij}^{cw} + V_{ij}'^{cw}\right) \le g \cdot \left(\mathrm{ADM}_i^w + \mathrm{ADM}_j'^{w}\right)$$
$$\sum_c \sum_i V_{ij}^{cw+} \le g \cdot \mathrm{ADM}_j^{w+}.$$

The receiver constraint requires that the total number of virtual connections should be less than the receiving capacity of the equipment at this node

$$V_{ij}'^{cw} \in \{0,1,-1\}. V_{ij}^{cw+}, \mathrm{ADM}_j'^{w} \text{ and } \mathrm{ADM}_j^{w+} \in \{0,1\}.$$

As we know, the integer linear programming problem is NP-complete [4]. The reconfiguration problem is described based on integer linear programming models. We expect this problem also to be intractable. In the next section, we propose heuristic approaches to solve this problem.

III. Heuristic Algorithms

The heuristic algorithms for dynamic grooming were developed for both the best fit case and the full fit case.

A. Best Fit

The objective of the best fit case is to include as much new traffic as possible using available capacity of the current configuration of the ring networks without increasing the number of SADMs. Two heuristic algorithms are proposed for best fit case, greedy heuristic and tabu search heuristic (TS-1).

1) Greedy Heuristic: In our greedy algorithm, the value of each entry of both the old matrix from which the current configuration was obtained and the new matrix using which we will perform the reconfiguration is generated randomly in the range $[0, r]$. The value is uniformly distributed between 0 and r. We try to fit as much new traffic as possible without adding more SADMs. Here is a description of the algorithm.

1) Get grooming information for the original traffic matrix using an existing algorithm ([5], [6]).

   For each circle, which we create for the current configuration, we should know whether the entire capacity between two nodes is occupied and whether there is an SADM at this node. We should also know on which wavelength this circle is groomed.

2) Find the difference traffic matrix.

   Given the new traffic matrix, compute the difference between the old one and the new one e.g. $D[i,j] = T'[i,j] - T[i,j]$. This is the matrix we try to groom in our algorithm with existing SADMs and traffic capacity. For some entries $D[i,j] = -m < 0$ (there exist some connections built for the old matrix that are not needed in the new matrix any more), remove the connections between (i,j) from m circles over at most m wavelengths.

3) Merge connections.

   We want to keep the largest continuous gap between nodes. So if there are two continuous connections over two circles, we merge them into a bigger one on the same circle.

4) Groom new traffic.

   Start from the smallest hop.

```
From node 0 to node $N-1$
From $s=1$ to $s=N$/*s is the number of
  hops*
while $(D[i,i+s] \neq 0)$
  Try all the circles,
  for each node pair ($i$,$i+s$) do
  if capacity is available *and*
  if there is SADM on both $i$ and $i+s$
  {
  Groom one unit of $D[i,i+s]$ to this
    circle.
    set capacity occupied.
    $D[i.i+s] = D[i,i+s] - 1$.
    }
Endwhile
End
```

2) Tabu Search Heuristic (TS-1): Tabu search is a meta-heuristic approach to solve hard optimization problems. The optimizing function is $f(x)$ subject to $x \in X$. The set X summarizes constraints on the vector of decision variables x. If x is the initial solution, neighborhood $N(x)$ is a set obtained by going one step further from the solution x. Such a step is called a move. Each element in $N(x)$ is put into the candidate list. At the same time, a tabu list T is built to keep track of the solutions that have been visited before. Each element in the tabu list has a tabu tenure. After each move, the value of tabu tenure is decremented by one. Once the value becomes zero, the element is removed from the list. The tabu list prevents such solutions being revistied within cycles of length less than or equal to tabu tenure. A modified neighborhood $N^*(x)$ of current solution is $N(x)$ defined as $N(x) \setminus T$. Here, $N^*(x)$ is a subset of $N(x)$. This kind of tabu search is short term TS. The best solution of $N^*(x)$ which is in the candidate list but not tabu (not in the tabu list) is chosen for the initial solution of next iteration. If after certain number of iterations (we call it the tabu limit) which is specified by the user, there is no improvement, the program stops. Otherwise, we continue the iteration and build a new candidate list.

Here is the algorithm description of TS-1.

1) Initialization.

   Compute the initial reconfiguration solution by using the greedy algorithm. Set the initial tabu tenure value, which should be defined before the program runs.

2) Build candidate list.

   The neighborhood of the solution x, $N(x)$ is defined as

$$N(x) = \{x' | x' \text{ is a move by swapping part of the two circles of the solution } x \text{ and one part of the swap should have available capacity}\}.$$

   Search all the circles. Compute $N(x)$ and put each element in $N(x)$ into the candidate list. Compute the additional new traffic amount that could been placed after such swap. Keep the maximum value of the increased traffic amount and its corresponding solution x.

3) Choose the best solution and continue.

   For all the solutions in the candidate list, choose the move that is not tabu and can increase the traffic amount the most. The values of tabu tenure of all the elements in the tabu list are decremented by one. If some element's tenure becomes zero, remove it from the tabu list. Put the current move into the tabu list.

4) Terminate or continue with the current solution.

   If after a certain number of iterations of moves, which is specified by the user, there is no improvement, the program stops. Otherwise, go back to build the candidate list for the new movement.

   The tabu tenure for TS-1 is 48 and the tabu limit is 60.

B. Full Fit Case

A two-phase algorithm is developed for the full fit case. The old traffic matrix and its solution are known. Recall that each entry of the old matrix and new matrix is generated randomly ranging from $[0, r]$. The objective is to fit all new traffic requests. Adding more SADMs is allowed.

1) Algorithm Description: Here, we present a two-phase algorithm for the full fit case.

1) Use a best fit algorithm (greedy or TS-1) to groom as much traffic as is allowed with existing SADMs.
2) If the capacity on circle c is available for the connection (i, j), place SADMs at nodes i and j on the wavelength which circle c is groomed on and groom traffic onto this circle.
3) Use the tabu search algorithm (TS-2) to groom the remaining traffic onto the new wavelength. Place SADMs at the nodes whenever necessary.

2) Tabu Search Heuristic (TS-2): The problem that was solved by TS-2 is the static traffic grooming problem, on which a lot of work has been done so far [2], [5], [6]. That is, given the remaining traffic matrix, groom that traffic onto wavelengths so that the number of SADMs is minimized. We observe that the static traffic grooming problem is a special case of the dynamic problem described in Section II-B when the old traffic matrix is empty and the new matrix is the traffic matrix that will be groomed. The following is the description of TS-2.

1) Initialization.

   Use the algorithm in [6] to get an initial solution x. Set the initial value of the tabu tenure.
2) Build candidate list.

   The neighborhood of the solution x, $N(x)$ is defined as

$$N(x) = \{x' | x' \text{ is a move by swapping two circles from different wavelength of solution } x \text{ and the swapping will lead to no more or at most one more SADM on a wavelength}\}.$$

   Search all the circles. Compute $N(x)$ and place each element in $N(x)$ into the candidate list. Compute the number of SADMs that could be saved after this swap. Keep the maximum number of saved SADMs.
3) Choose the best solution and continue.

   For all the solutions in the candidate list, choose the move that is not tabu and saves the most SADMs. The values of tabu tenure of all the elements in the tabu list are decremented by one. If some element's tenure becomes zero, remove it from the tabu list. Place the current move into the tabu list.
4) Terminate or continue with the current solution.

   If after a certain number of iterations of moves, which is specified by the user, there is no improvement, the program stops. Otherwise, go back to build the candidate list for the new movement.

   The tabu limit is 170 and tabu tenure is in $15, 20, 24$.

IV. Simulation Results and Analysis

In this section, we present the simulation results for both the best fit case and the full fit case.

A. Results for the Best Fit Case

First, we define an upper bound on the number of new connections that can be groomed to the current configuration to evaluate the performance of the best fit case algorithms. Then we give the result of both the greedy and the tabu search (TS-1) algorithms according to this upper bound. The running time comparison of both algorithms are also given.

1) Evaluation Factors: The results of reconfiguration algorithms depend not only on the algorithms themselves but also on the input matrix. The best fit strategy strives to place as much traffic as it can, but there is no guarantee to fit all of the traffic in. Here, we first develop an upper bound U on the number of connections that could possibly be groomed

$$U = \sum_i \sum_j \sum_c \{V_{ij} | \text{ there is enough capacity between } i \text{ and } j \text{ for new traffic } T'[i][j] \text{ and there is an SADM at both nodes } i \text{ and } j \text{ on circle } c\}.$$

The upper bound is computed by searching all the circles that are already built according to the old matrix to find available capacity to groom new connections. A connection can be established if there is capacity available between the terminal nodes and there are SADMs at the two nodes. The capacity of a connection is equal to the base bandwidth of one wavelength.

Although this upper bound is loose, because not all the connections available in the upper bound can be established at the same time, we can prove that no more connections can be built beyond this upper bound. We define the load factor α by using this upper bound

$$\alpha = \frac{\text{actual new connections established}}{\text{upper bound}} \times 100\%.$$

This factor shows how much percentage of the new traffic could be groomed into the current configuration according to the upper bound.

2) Results of Greedy Heuristic: We randomly generate 18 matrix pairs for the old traffic matrices and the new matrices (the matrix pair is numbered as $1, 2, \ldots, 18$ in Table III) for a five-node unidirectional ring with a granularity (g) of 3. The traffic amount generated for each node pair is evenly distributed between the range of 0 to $12 (r = 12)$. Table III provides the grooming results by using the evaluation factor α. From this table we observe that, when the number of nodes is not large and the granularity of the ring is small, the greedy algorithm can assign most of the new traffic that is possible to be groomed, according to the upper bound.

3) Results of TS-1: Table IV shows the results for the greedy algorithm and TS-1 under different numbers of nodes and different granularities for unidirectional rings $(r = 12)$. For each entry, 20 matrix pairs (old matrix and new matrix) are randomly generated. The average value of α is computed. We observe that tabu search yields better results than the greedy algorithm for most cases, especially with a large number of nodes and large

TABLE III
RESULTS FOR THE BEST FIT GREEDY ALGORITHMS FOR A FIVE-NODE RING

Matrix pair No.	1	2	3
Upper bound	15	24	24
Greedy connections	15	24	24
load factor α	100%	100%	100%
Matrix pair No.	4	5	6
Upper bound	23	20	13
Greedy connections	22	19	10
load factor α	96%	95%	77%
Matrix pair No.	7	8	9
Upper bound	10	19	8
Greedy connections	9	17	7
load factor α	90%	89%	88%
Matrix pair No.	10	11	12
Upper bound	23	12	18
Greedy connections	23	12	17
load factor α	100%	100%	94%
Matrix pair No.	13	14	15
Upper bound	22	18	33
Greedy connections	17	16	29
load factor α	82%	88%	88%
Matrix pair No.	16	17	18
Upper bound	18	16	32
Greedy connections	15	16	25
load factor α	83%	100%	78%

TABLE IV
RESULTS FOR THE BEST FIT ALGORITHMS. N IS THE NUMBER OF NODES IN A RING. α IS THE LOAD FACTOR. g IS THE GRANULARITY

Node number	method	g=3	g=4	g=12
N=5	greedy	92	89	79
(α *in* %)	tabu	92	90	80
N=6	greedy	90	89	78
(α *in* %)	tabu	91	90	80
N=8	greedy	90	84	76
(α *in* %)	tabu	91	85	77
N=12	greedy	86	83	70
(α *in* %)	tabu	87	84	72
N=16	greedy	85	82	67
(α *in* %)	tabu	86	84	68
N=20	greedy	84	80	63
(α *in* %)	tabu	86	82	69

TABLE V
ABSOLUTE CONNECTIONS BUILT BY THE GREEDY AND THE TABU ALGORITHMS

Node Number	method	g=3	g=4	g=12
N=5	greedy	16	17	22
	tabu	17	17	23
N=6	greedy	22	24	35
	tabu	22	25	36
N=8	greedy	39	40	54
	tabu	39	40	54
N=12	greedy	78	87	123
	tabu	79	88	126
N=16	greedy	128	142	193
	tabu	130	146	198
N=20	greedy	204	224	280
	tabu	209	229	290

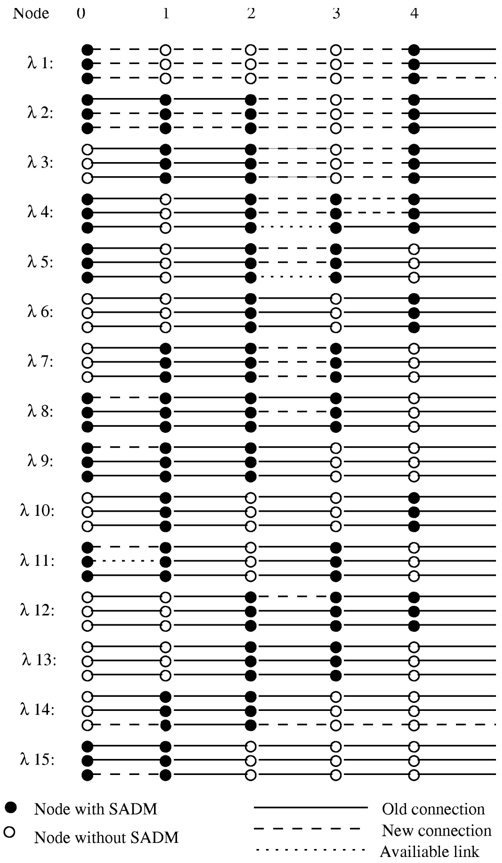

Fig. 4. Example of a reconfiguration problem using the greedy algorithm proposed in this work.

granularity. Table V shows the average number of new connections that can be groomed by the greedy algorithm and the tabu search algorithm using the same set of input data as in Table IV. The TS-1 gains 2% more connections than the greedy algorithm on the average.

Here, we give a specific example to indicate that the TS-1 can find a better solution than the greedy algorithm. In this example, there are five nodes in a unidirectional SONET ring. The granularity of the ring is three. The old traffic matrix and the new matrix are generated as follows:

$$T = \begin{bmatrix} 0 & 2 & 6 & 1 & 0 \\ 7 & 0 & 6 & 1 & 3 \\ 2 & 2 & 0 & 3 & 3 \\ 5 & 2 & 7 & 0 & 2 \\ 4 & 5 & 7 & 2 & 0 \end{bmatrix} \quad T' = \begin{bmatrix} 0 & 4 & 6 & 2 & 5 \\ 7 & 0 & 8 & 1 & 4 \\ 3 & 3 & 0 & 10 & 11 \\ 5 & 4 & 7 & 0 & 2 \\ 6 & 5 & 7 & 3 & 0 \end{bmatrix}.$$

Figs. 4 and 5 show the grooming results for the greedy algorithm and the tabu search algorithm respectively. The following two matrices (R_g, R_t) are the remaining traffic matrices that cannot be groomed after employing those two algorithms. We observe from Fig. 5 that two connections of $t_{2,3}$ are groomed to wavelength 5 instead of wavelength 4 in the tabu search algorithm. Then, two more connections of $t_{2,4}$ could be groomed to wavelength 4 which results in a gain of two more connections for the tabu search than the greedy heuristic

$$R_g = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 2 \\ 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \quad R_t = \begin{bmatrix} 0 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

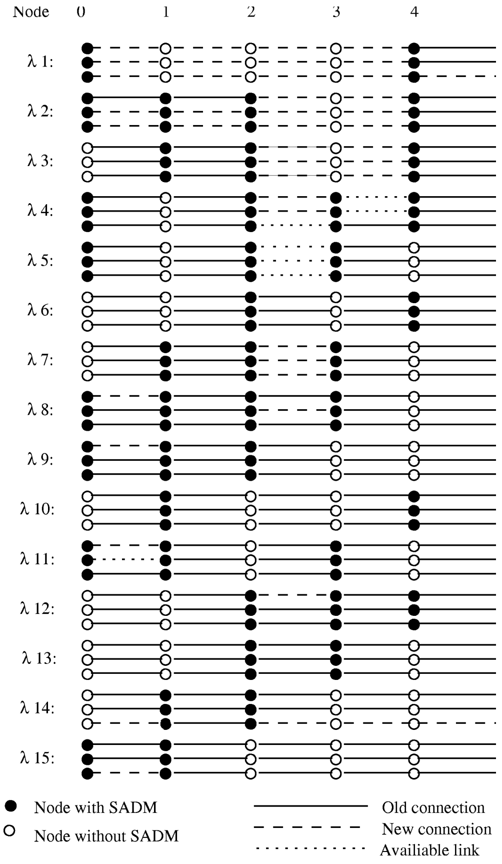

Fig. 5. Example of a reconfiguration problem using the TS-1 proposed in this work.

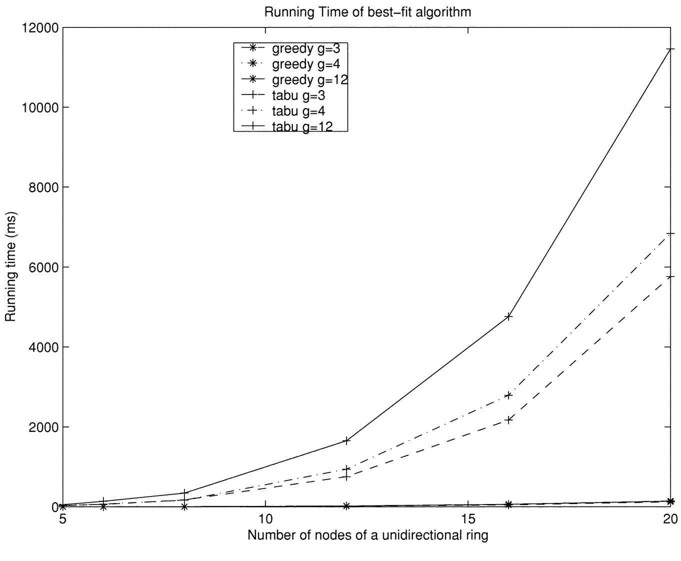

Fig. 6. Running time of the best fit algorithm.

4) Revenue Analysis and Running Time: While tabu search yields better results in most cases, it also takes more time to obtain the solution with the tabu heuristic than with the greedy algorithm. Fig. 6 shows the running time of the algorithms for the unidirectional ring under different values of granularity and different numbers of nodes. It uses the same set of data that generated the result in Table IV. As we note from the figure, the running time increases greatly when the number of nodes in the ring increases. We compute the revenue [9] of the reconfiguration problem according to the number of connections that are built in total. Revenue is the income the network service provider earns by running the network. The following is the revenue formula for the reconfiguration problem.

C: the cost of using a single link (a single link is any node pair in $[i, i+1]$, where $i \in 0, \ldots, N-1$)

$$\text{Revenue} = \sum_{w} \sum_{c} \sum_{i} \sum_{j} \left(V_{ij} + V'_{ij}\right) \times (j - i + N) \bmod N \times C \qquad (1)$$

From the formula, we observe that even one more connection can achieve quite a large percentage gain in the revenue if the cost of a single link is high. So although the TS-1 usually takes longer than the greedy algorithm to run, it is still worthwhile to employ tabu search when the single link cost is relatively high.

TABLE VI
RESULTS FOR STATIC TRAFFIC GROOMING

g	method	N=4	N=8	N=16	N=20
g=3	greedy	7	31	79	146
	SA	7	31	69	124
	TS-2	7	31	69	124
g=4	greedy	7	29	71	123
	SA	7	28	66	120
	TS-2	7	28	66	120
g=16	greedy	4	14	36	65
	SA	4	14	33	57
	TS-2	4	14	33	58
g=48	greedy	4	8	21	34
	SA	4	8	19	37
	TS-2	4	8	21	32
g=64	greedy	4	8	15	28
	SA	4	8	15	28
	TS-2	4	8	15	28

B. Results for the Full Fit Case

We mentioned in Section III-B2 that the static traffic grooming problem is a special case of the dynamic problem (for the full fit case). We run the algorithm (TS-2) for the static traffic grooming problem for a unidirectional ring under uniform traffic. The traffic amount of each entry in the matrix is one unit (equals base bandwidth). We compare the results with both the greedy algorithm [6] and the simulated annealing algorithm [5] in Table VI. For ILP results see [5]. We observe that the tabu search algorithm obtains the same results as simulated annealing (SA) for most cases. One entry for the tabu search method is better than SA. For some cases, the number of SADMs used in tabu heuristic is a little bit more than in SA. The running time for each entry is less than 27 seconds on a 450-MHz UltraSPARC II processor based SUN Ultra-60 Workstation. In [5], the SA algorithm was run for 30 trials and best result was chosen. TS-2 does not depend on a statistical result and need not be run multiple times. We observe that it is never worse than the greedy algorithm. TS-2 is relatively stable, which is preferred in dynamic models. Because the traffic pattern changes from time to time, it is usually not feasible to run the program many times to obtain the best solution.

V. Conclusion

In this work, we introduced the dynamic traffic grooming model for reconfiguration problems. Two cases (best fit and full fit) are presented in an integer linear programming description. Since integer linear programming problems are NP-hard, we expect that dynamic traffic grooming problem is also intractable for both cases. For the best fit case, two heuristic algorithms greedy and tabu search (TS-1) are proposed. An upper bound of the number of new virtual connections is developed to evaluate the performance of the algorithms. The results show that the tabu search algorithm (TS-1) proposed in this study will yield better solutions but takes more running time than the greedy algorithm for the best fit case. For the full fit case, a two-phase algorithm is developed. We observe that the static traffic grooming problem is a special case of the dynamic traffic grooming problem. The algorithms we proposed here (particularly the full fit case, TS-2) can also solve the static grooming problem. Our algorithm is more stable than the simulated annealing algorithm proposed in previous work. Moreover, some of the solutions are better than those obtained in the earlier work on the static grooming problem.

References

[1] A. L. Chiu and E. H. Modiano, "Reducing electronic multiplexing costs in unidirectional SONET/WDM ring networks via efficient traffic grooming," in *Proc. IEEE/IEICE Global Telecommunications Conf.*, vol. 1, Sydney, Australia, Nov. 1998, pp. 332–327.

[2] O. Gerstel and R. Ramaswami, "Cost effective traffic grooming in WDM rings," in *Proc. IEEE INFOCOM '98*, vol. 1, 1998, pp. 69–77.

[3] E. Modiano and A. Narula-Tam, "Mechanisms for providing optical bypass in WDM-based networks," *SPIE Optical Networks Mag.*, vol. 1, no. 1, pp. 9–16, Jan. 2000.

[4] A. Schrijver, *Theory of Linear and Integer Programming*. New York: Wiley, 1986.

[5] J. Wang, V. R. Vemuri, W. Cho, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM ring networks: nonuniform traffic and bidirectional ring," in *Proc. ICC 2000*, vol. 3, 2000, pp. 1295–1299.

[6] X. Zhang and C. Qiao, "An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings," in *Proc. SPIE-Int. Society for Optical Engineering*, vol. 3531, 1998, pp. 221–232.

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

[8] P. Wan, L. Liu, and O. Frieder, "Grooming of arbitrary traffic in SONET/WDM rings," in *Proc. IEEE/IEICE Global Telecommunication Conf.*, vol. 1B, 1999, pp. 1012–1016.

[9] M. Sridharan and A. K. Somani, "Revenue maximization in survivable WDM networks," in *Proc. Optical Networking and Communications, SPIE*, vol. 4233, 2000, pp. 291–302.

Shu Zhang received the B.E. degree in computer science and engineering from Northern Jiaotong University, Beijing, China, in 1997 and the M.S. degree in computer science from the University of Nebraska, Lincoln, in 2001.

She has been with Microsoft Corporation, Redmond, WA, since June, 2001

Byrav Ramamurthy (S'98–A'98) received the B.Tech. degree in computer science and engineering from Indian Institute of Technology, Madras, India, in 1993 and the M.S. and Ph.D. degrees in computer science from University of California (UC), Davis, in 1995 and 1998, respectively.

Since August 1998, he has been an Assistant Professor in the Department of Computer Science and Engineering, University of Nebraska, Lincoln (UNL). He is the Founding Co-Director of the Advanced Networking and Distributed Experimental Systems (ANDES) Laboratory at UNL. He has served as a Member of the technical program committees for several conferences and symposia in the areas of networking and optical communications. His research areas include optical and wireless networks, distributed systems, computer security, and telecommunications.