An Effective and Comprehensive Approach for Traffic Grooming and Wavelength Assignment in SONET/WDM Rings

Xijun Zhang and Chunming Qiao, *Member, IEEE*

Abstract—**In high-speed SONET rings with point-to-point WDM links, the cost of SONET add–drop multiplexers (S-ADMs) can be dominantly high. However, by grooming traffic (i.e., multiplexing lower-rate streams) appropriately and using wavelength ADMs (WADMs), the number of S-ADMs can be dramatically reduced. In this paper, we propose optimal or near-optimal algorithms for traffic grooming and wavelength assignment to reduce both the number of wavelengths and the number of S-ADMs. The algorithms proposed are generic in that they can be applied to both unidirectional and bidirectional rings having an arbitrary number of nodes under both uniform and nonuniform (i.e., arbitrary) traffic with an arbitrary grooming factor. Some lower bounds on the number of wavelengths and S-ADMs required for a given traffic pattern are derived, and used to determine the optimality of the proposed algorithms. Our study shows that using the proposed algorithms, these lower bounds can be closely approached in most cases or even achieved in some cases. In addition, even when using a minimum number of wavelengths, the savings in S-ADMs due to traffic grooming (and the use of WADMs) are significant, especially for large networks.**

Index Terms—**ADMs, SONET, traffic grooming, wavelength assignment, WDM rings.**

I. Introduction

SYNCHRONOUS optical network (SONET) rings are widely used in today's network infrastructures. Each SONET ring is constructed by using fibers to connect SONET add-drop multiplexers (hereafter called S-ADM for simplicity). Typically, for each working fiber, there is a protection fiber and hence, two and four fibers are usually used to construct unidirectional and bidirectional rings, respectively. One of the critical operations of the S-ADMs is traffic grooming. Specifically, each S-ADM can multiplex multiple lower-rate streams to form a higher-rate stream, or demultiplex a higher-rate stream to several lower-rate ones. For example, four OC-12 streams can form one OC-48 stream, in which case the grooming factor is 4.

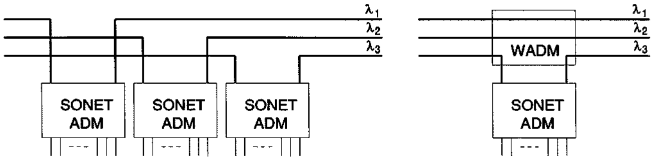


Fig. 1. Reducing the number of S-ADMs.

In a SONET ring with point-to-point WDM links, let W be the number of wavelengths needed to support a given traffic pattern. If one S-ADM is used on every wavelength at every node, the total number of S-ADMs is $N \cdot W$, where N is the number of nodes. When the number of wavelengths is large (e.g., $W \geq 32$) and each wavelength operates at OC-48 (or higher), the dominant system cost is no longer the cost of the fibers but that of S-ADMs. Fortunately, a node may not need to add/drop streams on every wavelength, especially if the traffic destined to the node can be groomed onto only one or a few wavelengths (instead of spreading it over all wavelengths). By employing wavelength routing at each node, that is, using a wavelength ADM (WADM) capable of dropping (and adding) only the wavelengths carrying traffic destined to (and originated from) a node, the number of S-ADMs needed can be dramatically reduced. For example, Fig. 1 shows a node with two different configurations, the one at left using three S-ADMs with point-to-point WDM links, and the other at right using only one S-ADM plus a WADM (assuming that only λ_3 carries streams that need to be added/dropped at this node). Let D be the number of S-ADMs required when using WADMs and traffic grooming to support the given traffic pattern, then the saving percentage on the number of S-ADMs can be defined as $S = (N \cdot W - D)/(N \cdot W)$.

In this paper, we consider cost-effective designs of SONET over WDM rings for a given (static) traffic pattern, where the traffic from one node to another may require a fraction of the bandwidth provided by one wavelength. We assume that at each node, a WADM and as many S-ADMs as necessary may be used. Our objective is to minimize the number of wavelengths (W) and the total number of S-ADMs (D) required to support the given traffic pattern by grooming (or multiplexing) traffic between different source–destination node pairs at each node whenever needed (and by assigning wavelengths appropriately). Note that, one may not always be able to minimize both W and D at the same time. An example in which they cannot be minimized simultaneously is given in Section IV.

Manuscript received January 8, 1999; revised September 13, 1999; approved by IEEE/ACM Transactions on Networking Editor B. Mukherjee. This work was supported in part by the National Science Foundation under Contract MIP-9409864 and Contract ANIR-9801778. An earlier version was published in *SPIE Proc., All Optical Networks*, vol. 3531, pp. 221–231, 1998.

X. Zhang was with Lucent Technologies InterNetworking Systems, Westford, MA 01886 USA. He is now with Quantum Bridge Communications, Andover, MA 01810 USA (e-mail: xzhang@quantumbridge.com).

C. Qiao is with the Department of Computer Science and Engineering, State University of New York at Buffalo, Amherst, NY 14260 USA (e-mail: qiao@computer.org).

Publisher Item Identifier S 1063-6692(00)09115-9.

A major difference between this work and other work done previously is the generality of our approach in that the proposed traffic grooming and wavelength assignment algorithms can be applied to either unidirectional or bidirectional SONET/WDM rings with an arbitrary network size N and an arbitrary grooming factor. More specifically, [4] considered wavelength assignment for a given set of lightpaths in SONET/WDM rings to reduce D and/or W but did not consider traffic grooming. [6], [2] proposed heuristics for grooming uniform traffic in unidirectional SONET/WDM rings. Traffic grooming for uniform traffic in bidirectional SONET/WDM rings having an odd number of nodes is discussed in [9] without specifying the algorithm(s) or heuristic(s) employed, though a specific type of nonuniform traffic, namely, distance-dependent traffic, was recently studied in [8]. In [5], analytic results (such as D and W required) were presented for several specific optical WDM ring designs under uniform traffic (although a framework allowing nonuniform traffic was also discussed). The SONET/WDM rings considered in this paper differ from all the designs considered in [5]. To our best knowledge, this is also the first paper to report quantitative results for arbitrary traffic grooming. This work also differs from others in that it effectively separates wavelength assignment from traffic grooming, and thus helps simplify both problems and obtain efficient solutions (see Section II for more discussion).

The rest of the paper is organized as follows. Section II describes the problem of traffic grooming and the proposed generic approach which has two major phases, namely, circle construction and circle grooming. Sections III and IV propose circle construction algorithms for uniform and nonuniform traffic, respectively. For uniform traffic, optimal circle construction algorithms (i.e., those resulting in a minimum number of circles) exist for both unidirectional and bidirectional SONET/WDM rings. For nonuniform traffic, a heuristic is proposed which uses the rules developed for uniform traffic as a first step of circle construction and a greedy algorithm as a second step. In Section V, lower bounds on the number of S-ADMs and the number of wavelengths required are determined for the cases with and without traffic grooming with uniform or nonuniform traffic. A generic circle grooming algorithm, which is applicable to both unidirectional and bidirectional rings, as well as to both uniform traffic and nonuniform traffic, is proposed in Section VI. Numerical results are presented and discussed in Section VII. Finally, Section VIII concludes the paper.

II. General Methodology

To facilitate our presentation, we number the N nodes in a ring from 0 to $N-1$, and use (i, s) to denote a connection from node i to another node j that is s hops away along a shortest path. Hereafter, such a connection will be said to have a *stride* (or hop count) of s. Let B be the bandwidth of one wavelength (e.g., OC-48) and r_b the base bandwidth of a connection (e.g., OC-3), where $B = m \cdot r_b$ for some integer $m \geq 1$. In addition, let $R_{i,s} = h_{i,s} \cdot r_b$ (where $h_{i,s} \geq 0$) denote the total bandwidth required by the traffic from node i to node j which is s hops away. Note that $h_{i,s}$ can be considered as the number of connections to be established from i to j. If $h_{i,s} > m$, these connections have to be groomed onto different wavelengths. However, even if $h_{i,s} < m$, these connections may still be groomed onto different wavelengths in order to minimize W and/or D, which is the objective of traffic grooming.

For arbitrary traffic, $R_{i,s}$ (and $h_{i,s}$) may vary with i and s. Let h be the *greatest common divider* (GCD) of all *nonzero* $h_{i,s}$, i.e., $h = \text{GCD}(\{h_{i,s} \neq 0\})$. Without loss of generality, we may assume that $\alpha = \text{GCD}(m, h)$ is 1. This is because if $\alpha > 1$, α connections from one node to another can be bundled into a *super*-connection, which effectively increases the base bandwidth r_b by α times, and reduces both m and h as well as all (nonzero) $h_{i,s}$'s by α times. In what follows, we define m to be the *grooming factor.* When $m = 1$, each connection will be established as a lightpath, and hence no traffic grooming is needed. On the other hand, traffic grooming is needed when $m > 1$.

The basic idea behind the proposed approach to traffic grooming and wavelength assignment is as follows. First, heuristic algorithms based on the scheduling algorithms proposed in [7], [10], [11], are used to construct as few *circles* as possible to include all requested connections so as to minimize W, where each circle consists of multiple nonoverlapping (i.e., link disjoint) connections. Second, after the circles are constructed, another heuristic algorithm is used to groom up to m circles onto a wavelength (or λ) ring while trying to overlap as many end nodes belonging to different circles as possible so as to result in a small D. Let C be the total number of circles constructed to support the given traffic pattern. There will be $\lceil C/m \rceil$ λ-rings. That is, $W = \lceil C/m \rceil$ wavelengths are needed, one for each of these λ-rings.

Note that wavelength assignment is normally a part of the traffic grooming problem. Our approach, however, can effectively separate wavelength assignment from traffic grooming, and thus help simplify both problems and obtain efficient solutions. Specifically, if D does not need to be minimized, one may arbitrarily groom m circles onto each λ-ring. Otherwise, these circles can be groomed in a more judicious way using a grooming algorithm. In any case, once the λ-rings are constructed, an arbitrary available wavelength can be assigned to each λ-ring. Note that, to some extent, wavelength assignment has largely been accomplished in the circle construction phase. In other words, once the circles are constructed, it has been determined that the connections in each circle will be assigned the same wavelength, and in addition, the number of wavelengths to be used, W, has also been determined (and possibly minimized). In what follows, we first study circle construction algorithms for uniform and nonuniform traffic, respectively. Then, we propose a generic circle grooming algorithm.

III. Circle Construction for Uniform Traffic

In uniform traffic, as a special case of arbitrary traffic, $R_{i,s}$ is the same for every i and s, and thus we may let $R = R_{i,s}$ and have $h = h_{i,s}$. If $R = r_b$ (and $h = 1$), each node needs to establish one connection to every other node for a total of $N(N-1)$ connections from all N nodes.

An algorithm was proposed in [10] to construct $C = (N(N-1))/2$ circles in unidirectional rings. This algorithm, hereafter

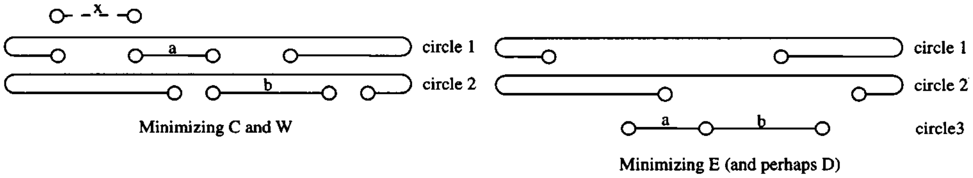


Fig. 2. An illustration of the two options in Algorithm IV.

referred to as **Algorithm I,** combines two connections having common end nodes (e.g., one from i to j and the other from j to i) to form a *full circle* (which spans every hop of the ring). Note that, as to be discussed in more detail in Section V, having full circles will minimize W and when $m = 1$, minimize D as well (in addition, it will reduce D when $m > 1$).

Two algorithms called "Complementary Assembling with Dual Strides" (hereafter **Algorithm II**) and "Complementary Assembling with Triadic Strides" (hereafter **Algorithm III**) were proposed in [11] to construct full circles in bidirectional rings with even and odd N, respectively. Specifically, for an even N, Algorithm II combines either two or four connections, while for an odd N, Algorithm III combines either three or four connections to form a full circle (both clockwise and counter-clockwise). The total number of circles (in either clockwise or counter-clockwise direction[1]) constructed by the two algorithms are $\lceil N^2/8 \rceil$ and $(N^2 - 1)/8$, respectively.

Note that the case where $h > 1$ is the same except that each node will need h connections to every other node, and the total number of connections will be $h \cdot N(N-1)$. Accordingly, h copies for each circle constructed in the case of $h = 1$ will be constructed.

IV. Circle Construction for Nonuniform Traffic

In this section, we propose a heuristic algorithm to construct circles for a given arbitrary traffic matrix $\{h_{i,s}\}$.

Based on the previous discussion on uniform traffic, in order to use a small W and D, it is natural to first construct as many full circles as possible by combining two connections with common end nodes in unidirectional rings, and up to four connections with overlapping end nodes in bidirectional rings. Assume that C_1 full circles can be constructed this way. After these C_1 full circles are constructed, and the traffic matrix $\{h_{i,s}\}$ is updated by removing the connections in those C_1 circles, there may still be some nonzero $h_{i,s}$ corresponding to connections that remain to be included in some circles. In other words, it is possible that not all the requested connections can be included in the way described above. Hence, we propose the following heuristic algorithm, **Algorithm IV** (which is applicable to both unidirectional and bidirectional rings), shown on the next page, to construct additional (say C_2) circles for a total of $C = C_1 + C_2$ circles. Note that each of the C_2 additional circles could be full or partial due to the nonuniform nature of the traffic demand. In a partial circle, there are one or more "gaps" which cannot be fit in by any remaining connection to be established, resulting in some bandwidth being wasted. The left-hand side of Fig. 2 shows two partial circles (assuming connection x has not been considered yet).

Intuitively, fewer gaps help reduce not only W (due to better bandwidth utilization and thus fewer circles), but also D (more formal discussion will be given in Section V-C). In order to minimize the number of gaps, the proposed heuristic attempts to fit each connection into existing circles without generating an additional gap. More specifically, Algorithm IV (in pseudo code) works as follows. It constructs circles using the connections having the longest stride in the traffic matrix $\{h_{i,s}\}$ first (this is because connections with shorter strides are more likely to be able to fit into the gaps generated by the connections with longer strides). If the connection being considered shares at least one end node (source or destination) with other connections already contained in the circle, it will be added into the circle (unless there is no room, or in other words, the new connection will overlap with an existing one). If fitting a connection into any existing circle will generate an additional gap, we will call this connection a "gap maker," and put it into a *GapMaker* list which is initially empty.

After all the connections in the traffic matrix have been either included in some existing circles, or put into the GapMaker list, we start to process the GapMaker list (that is, to include its connections in existing or additional circles). Note that, it is possible that a connection from the GapMaker list will now fit into an existing circle without creating an additional gap. For example, as shown in Fig. 2 (left), connection a was put into the GapMaker list because connection x has not been included in circle 1 at the time. However, after connection x is added to circle 1, connection a can be removed from the GapMaker list and added to circle 1 as well without creating an additional gap.

If there are still some connections left in the GapMaker list that cannot be fit into any existing circle without creating an additional gap, we have two options: one is to minimize C_2, which in turn minimizes C (and W), and the other is to minimize the number of end nodes (say, E) involved in all the C circles (which helps reduce D), where $E \leq D$ ($E = D$ if no grooming is needed). If C (or W) is to be minimized, each connection will be fit into an existing circle as long as there is enough bandwidth, even though an additional gap may be created. In other words, a new circle is created for a connection *only* if there is no room for the connection in any existing circle. On the other hand, if E is to be minimized, a new circle will be generated for a connection that cannot be fit into any existing circle without creating an additional gap. Generating a new circle for this "gap maker" gives a chance for all "gap makers" to share end nodes. Fig. 2 illustrates the difference between these two options assuming that connection x does not exist at all. As can be seen, the first option (shown at left) results in one fewer circles (for a total of 2) but one more end node (for a total of 8)

[1] We are only interested in the number of circles in one direction as it will be used to derive W and D.

Algorithm IV: Construct circles for non-uniform traffic

```
$C_2 = 1$; // $C_2$ is the number of circles being constructed by this algorithm so far
$s_0 = N - 1$ for unidirectional rings or $s_0 = \frac{N}{2}$ for bidirectional rings; //the maximum stride
for $s = s_0, s_0 - 1, \cdots, 1$ { // consider connections with longer strides first
    for $i = 0, 1, \cdots, N - 1$ { // one node at a time
        while $(h_{i,s} > 0)${
            for $c = 1, 2, \cdots, C_2$ { // try all circles if necessary
                try to fit connection $(i, s)$ into circle $c$ without creating an additional gap;
                if (succeed) {
                    $h_{i,s} = h_{i,s} - 1$;
                    break from the inner "for" loop; // no need to try the remaining circles
                } // else *fail* and then try the next circle
            }
            if (fail because $(i, s)$ would overlap with some connections in any existing circle) {
                do {
                    generate a new circle for $(i, s)$;   $h_{i,s} = h_{i,s} - 1$;   $C_2 = C_2 + 1$;
                } while $(h_{i,s} > 0)$;
            }
            if (fail because fitting $(i, s)$ in any existing circle would generate an additional gap) {
                do {
                    put $(i, s)$ into the GapMaker list (to be processed later);   $h_{i,s} = h_{i,s} - 1$;
                } while $(h_{i,s} > 0)$;
            }
        }
    }
} // all $h_{i,s}$ have been included either in the circles or in the GapMaker list
if (the objective is to minimize the number of circles) {
    for each connection in the GapMaker list {
        try to fit into existing circles;
        if (fail) { // because it overlaps with existing connections
            generate a new circle for the connection;   $C_2 = C_2 + 1$;
        }
        remove the connection from the GapMaker list;
    }
}
if (the objective is to minimize the number of end nodes) {
    for each connection in the GapMaker list {
        try to fit into existing circles without creating an additional gap;
        if (fail) { // because it either overlaps with existing connections
                    // or it creates an additional gap
            generate a new circle for the connection;   $C_2 = C_2 + 1$;
        }
        remove the connection from the GapMaker list;
    }
}
```

than the second option (shown at right). In the second option, a new circle for connection a gives a chance for connection b to share one end node with a. Other "gap makers" (if any) may also share one end node with either a or b in this new circle.

V. Lower Bounds

In this section, we present some lower bounds on the number of circles, wavelengths, and S-ADMs, denoted by C_{LB}, W_{LB} and D_{LB}, respectively.

A. Uniform Traffic with $m = 1$

As given in [10], the minimum number of circles that need to be constructed to support all the connections of uniform traffic in a unidirectional ring is $C_{\text{LB}} = (N(N-1))/2$. When $m = 1$ (i.e., no grooming is needed), the lower bound on the number of wavelengths is $W_{\text{LB}} = C_{\text{LB}}$. Finally, since there are at least two S-ADMs on each wavelength carrying one connection (one for the source and the other for the destination of the connection), a lower bound on D is $D_{\text{LB}} = 2 \cdot W_{\text{LB}} = N(N-1)$. Recall that Algorithm I constructs $C = (N(N-1))/2$ full circles, and each full circle needs two S-ADMs, one for each of the two common nodes in the circle. This means that all the three lower bounds can be achieved.

For bidirectional rings, it is first given in [3] (which discussed the case where N is odd only) and then in [7], [1], [11] that the minimum number of wavelengths (and circles) required in a bidirectional ring is

$$W_{\text{LB}} = C_{\text{LB}} = \begin{cases} \dfrac{N^2-1}{8} & \text{for odd } N \\ \left\lceil \dfrac{N^2}{8} \right\rceil & \text{for even } N. \end{cases} \qquad (1)$$

Recall that Algorithms II and III construct the same number of full circles, i.e., $C = C_{\text{LB}}$, (using up to four connections for each circle) as given in the above equation, which implies that these two lower bounds are achieved.

Though as in unidirectional rings, one may use $2 \cdot W_{\text{LB}}$ as a lower bound on D, such a lower bound is too loose for bidirectional rings (because each wavelength carries two to four connections). Since at least one S-ADM is needed to establish a connection (assuming the other end of this connection is always shared with another connection), the total number of connections to be established can be used as another lower bound on

D. In the case of unidirectional rings under uniform traffic, this lower bound is the same as $2 \cdot W_{\rm LB}$ (both equal to $N(N-1)$). In the case of bidirectional rings, we have $D_{\rm LB} = (N(N-1))/2$ (since only these many connections are established clockwise or counter-clockwise), which is tighter than $2 \cdot W_{\rm LB}$. In fact, since the number of S-ADMs needed for each full circle is equal to the number of connections in the circle, Algorithms II and III which construct only full circles to include all the requested connections guarantee that such a lower bound on D is achieved (the same reasoning can also be applied to Algorithm I in the case of unidirectional rings).

Note that, when $m = 1$, a bidirectional SONET/WDM ring employing our proposed traffic grooming and wavelength assignment algorithms behaves as a *fully optical ring*, which is considered in [5]. In other words, when there is no traffic grooming, our results on W and D agree with those obtained for the fully optical ring in [5]. However, our results in Section VII indicate that when $m > 1$ and $h < m$ (i.e., when the traffic from one node to another requires a fraction of the bandwidth of one wavelength), the bidirectional SONET/WDM ring will require fewer D and W than the fully optical ring (and other designs considered in [5]).

B. Uniform Traffic with $m > 1$

In this section, we examine the lower bounds $W_{\rm LB}$ and $D_{\rm LB}$ for uniform traffic when $m > 1$. When m circles can be groomed onto each wavelength, it is clear that $W_{\rm LB} = \lceil C_{\rm LB}/m \rceil$, and $W = \lceil C/m \rceil$. Hence, any circle grooming algorithm (such as Algorithm VI to be described later) that grooms up to m circles onto each wavelength will use the minimum number of wavelengths (provided that the number of circles constructed is also minimum, i.e., $C = C_{\rm LB}$, which is the case if Algorithm I, II, or III is used). However, to minimize D, different number of circles may need to be groomed on different wavelengths.

To determine a reasonably tight lower bound on D, assume that the actual number of circles groomed onto wavelength λ_w is m_w, where $1 \le w \le W$ and $\sum_{w=1}^{W} m_w = C$. Let the minimum number of S-ADMs needed on λ_w be denoted by $d(m_w)$ (note that $d(m_w)$ is also the number of end nodes involved on λ_w). If for a given set of m_w, where $w = 1, 2, \cdots, W$, which we call a *solution* and denote by $\{m_w\}$, we can find a unique $d(m_w)$ for each λ_w, the total minimum number of S-ADMs required can then be calculated as $TempD_{\rm LB} = \sum_{w=1}^{W} d(m_w)$. In addition, if a solution that has the minimum $TempD_{\rm LB}$ among all possible solutions (called *optimal* solution hereafter) can be obtained, $D_{\rm LB}$ can be set to be equal to the corresponding minimum $TempD_{\rm LB}$. In other words, we propose to use two steps to derive a reasonable $D_{\rm LB}$. First, for any given value of m_w, determine a reasonable $d(m_w)$. Second, for given C, W and m, find an optimal solution $\{m_w\}$ that gives the minimum $TempD_{\rm LB}$.

Note that in unidirectional rings under uniform traffic, the maximum number of (full) circles that can be constructed among n end nodes is $(n(n-1))/2$ or $\binom{n}{2}$ as long as every connection is unique (i.e., $h = 1$). This is independent of how circles are constructed (i.e., not specific to Algorithm I). Consequently, in order to have m_w circles on λ_w, we need to have $\binom{d(m_w)}{2} \ge m_w$. On the other hand, in order for $d(m_w)$ to be the minimum number of S-ADMs on λ_w as intended, we also need to have $\binom{d(m_w)-1}{2} < m_w$. Based on this observation, we can obtain a unique value of $d(m_w)$ for any given m_w. For example, let $m_w = 11$. Since $\binom{6}{2} = 15 > 11$ but $\binom{5}{2} = 10 < 11$, we have $d(m_w) = 6$, meaning that *at least* six S-ADMs are needed in order to have eleven circles groomed on the same λ. Similarly, $d(1) = 2$, $d(2) = 3, \ldots, d(16) = 7$, and so on.

For bidirectional rings, $d(m_w)$ needs to be calculated differently. Specifically, the maximum number of circles involving n nodes is now given by

$$C(n) = \begin{cases} \dfrac{n^2 - 1}{8} & \text{for odd } n \\ \left\lceil \dfrac{n^2}{8} \right\rceil & \text{for even } n \end{cases} \tag{2}$$

and thus, the following condition will be used to determine $d(m_w)$ for a given m_w:

$$C[d(m_w) - 1] < m_w < C[d(m_w)] \tag{3}$$

Algorithm V, shown on the next page, is proposed to find the optimal solution $\{m_w\}$ (and determine a reasonable $D_{\rm LB}$). Note that a trivial approach would require all the possible values of m_w for each w be examined. However, Algorithm V reduces the search space as follows. Without loss of generality, we may assume that $1 \le m_1 \le m_2 \le \cdots \le m_W \le m$ as the wavelength assignment can be arbitrary. Since $\sum_{w=1}^{W} m_w = C$, we have $\lceil C/W \rceil \le m_W \le m$. In Algorithm V, m_w is determined in a wavelength index descending order, i.e., m_W first, m_{W-1} second and so on, by calling a function $FindM()$ recursively. At the time m_w is to be determined, all m_k's with $w+1 \le k \le W$ have been determined, and the number of circles groomed so far is $TempC = \sum_{k=w+1}^{W} m_k$. In addition, since at least one circle needs to be groomed onto wavelengths having index from 1 to $(w-1)$, the number of circles that could possibly be groomed onto wavelength λ_w is at most $C - TempC - (w-1)$. Furthermore, according to the assumption described earlier, $m_w \le m_{w+1}$, and hence, we have $m_w \le \min\{m_{w+1}, C - TempC - w + 1\} = M_{\max}$. On the other hand, since the $C - TempC$ circles which have not been groomed so far will be allocated onto w wavelengths among which λ_w will be allocated the largest number of circles, we have $m_w \ge \lceil (C - TempC)/w \rceil = M_{\min}$. For every value of m_w in the range of $[M_{\min}, M_{\max}]$, $FindM(w-1)$ is called to determine possible values of m_{w-1}. In this way, we have limited the possible values of each m_w and in turn, the number of solutions to be examined by the algorithm.

For each solution $\{m_w\}$ examined by the algorithm, the corresponding number of S-ADMs required ($TempD_{\rm LB}$) is calculated as described earlier and the best solution with the lowest $TempD_{\rm LB}$ found so far is recorded. At the end of the algorithm, the optimal solution $\{m_w\}$ and $D_{\rm LB}$ can thus be obtained.

C. Non-Uniform Traffic

For a given nonuniform traffic matrix $\{h_{i,s}\}$, the traffic load on each and every link can be determined by counting all the connections on the link. Let the maximum traffic load over all

Algorithm V: Determine a lower bound on the number of S-ADMs for uniform traffic

```
main() { // $N$ and $m$ are given, $C$ and $W$ are pre-determined
        $D_{LB} = N \cdot W$; //may set it to any large value
        FindM($W$); //start with $\lambda_W$
        // $D_{LB}$ and the corresponding $\{m_w\}$ are obtained
}

FindM(integer $w$) {
        if ($w = W$) {
                $tempC = 0$; // $tempC$ is the number of circles groomed so far
                $U = m$; // $U$ is an upper bound on the number of circles on $\lambda_w$
        }
        else { // $w < W$
                $tempC = \sum_{k=w+1}^{W} m_k$;
                $U = m_{w+1}$;
        }

        if ($w > 1$) { // as long as this is not the last wavelength
                $M_{max} = min\{U, C - tempC - w + 1\}$;
                for $m_w = \lceil \frac{C-tempC}{w} \rceil, \lceil \frac{C-tempC}{w} \rceil + 1, \cdots, M_{max}$
                        FindM($w - 1$);
        }
        else { // groom all remaining circles onto the last wavelength
                $m_1 = C - tempC$;
                $tempD_{LB} = \sum_{k=1}^{W} d(m_k)$;
                if ($D_{LB} > tempD_{LB}$) {
                        $D_{LB} = tempD_{LB}$;
                        save the solution $\{m_k\}$
                }
        }
}
```

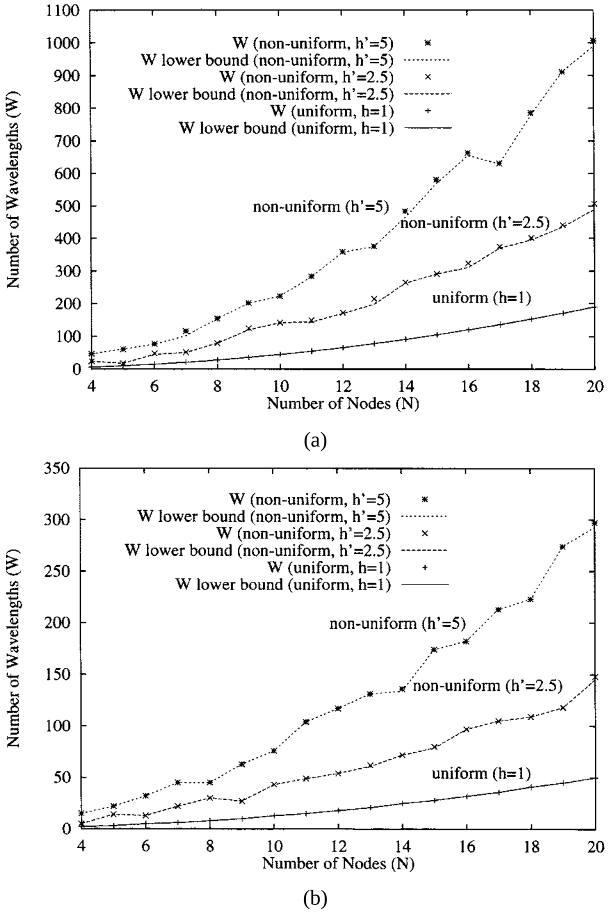
Algorithm VI: Groom circles onto W wavelengths

```
//determine the number of circles to be groomed onto each wavelength
if (using Method A) // Only applicable to uniform traffic
        invokes Algorithm V to find the theoretically optimal solution $\{m_w\}$;
if (using Method B) { //distribute circles as evenly as possible
        $C_g = 0$; // the number of circles groomed so far;
        for $w = W, W-1, \cdots, 1$ {
                $W' = \lceil \frac{C-C_g}{m} \rceil$;
                $m_w = \lceil \frac{C-C_g}{W'} \rceil$;
                $C_g = C_g + m_w$;
        }
}

//use a heuristic to groom a pre-determined number of circles onto each wavelength
find the number of S-ADMs in each circle;
$D = 0$;
for $w = W, W-1, \cdots, 1$ {
        find the circle which has the maximum number of S-ADMs (i.e. end nodes involved)
                over all existing circles, and groom it onto $\lambda_w$;
        for $k = 1, 2, \cdots, m_w - 1$ { // groom other $m_w - 1$ circles onto $\lambda_w$
                find a circle which, if groomed onto $\lambda_w$, results in a minimum number of
                        additional S-ADMs (or maximum overlapping among the end nodes);
                groom this circle onto $\lambda_w$;
        }
        $D = D +$ number of S-ADMs on $\lambda_w$;
}
```

links be $R_{\max}$, we have $C_{\text{LB}} = \lceil R_{\max}/r_b \rceil$. In addition, we can use $W_{\text{LB}} = \lceil R_{\max}/B \rceil$ as a lower bound on the number of wavelengths whether $m = 1$ or $m > 1$. Note that for unidirectional rings, one may calculate another lower bound on W as $(\sum_{i=0}^{N-1} \sum_{s=1}^{N-1} h_{i,s} \cdot s/m \cdot N)$ (a similar formula may be used for bidirectional rings). However, this lower bound is not as tight as the first one and hence will not be used.

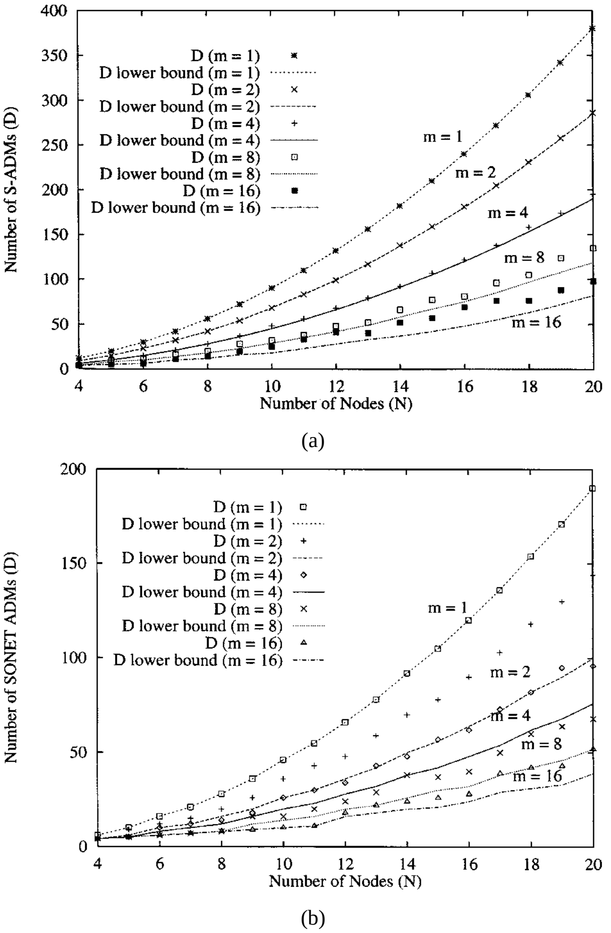
We now determine a reasonable lower bound on D when $m = 1$. For nonuniform traffic, full and/or partial circles need to be constructed using either Algorithm IV or any other possible heuristics. As mentioned in Section V-A, the number of S-ADMs needed in a full circle is equal to the number of end nodes (also the number of connections) involved in this circle. On the other hand, the number of S-ADMs in a partial circle is equal to the number of connections plus the number of gaps in this circle. Therefore, when no grooming is needed, the total number of S-ADMs is equal to the total number of connections (A) plus the total number of gaps (G), i.e., $D = A + G$. For a



(a)

(b)

Fig. 3. Wavelength requirement in SONET/WDM rings with no traffic grooming. (a) Unidirectional ring. (b) Bidirectional ring.



(a)

(b)

Fig. 4. The number of S-ADMs needed for uniform traffic when $h = 1$. (a) Unidirectional ring. (b) Bidirectional ring.

given traffic pattern, A is fixed. Hence, minimizing G is equivalent to minimizing D (this is the rationale behind Algorithm IV), and a lower bound on D can be derived from a lower bound on G (denoted by $G_{\rm LB}$).

A simple way to determine $G_{\rm LB}$ is as follows. Let s_i denote the number of connections using node i as the source, and d_i denote the number of connections using node i as the destination. The number of gaps involving node i at one of its ends is at least $|s_i - d_i|$. Therefore, $G_{\rm LB}$ can be given as $(1/2)\sum_{i=0}^{N-1} |s_i - d_i|$ (the fraction $1/2$ is needed because every gap is counted twice, once at each of its two end nodes). For any given traffic pattern, $D_{\rm LB}$ is calculated as $A + G_{\rm LB}$. This lower bound was first given in [4]. Since it is possible that two connections originating and terminating at node i cannot be in the same circle, thus resulting in additional gaps, $A + G_{\rm LB}$ may not be achievable in some cases. In fact, it may be even smaller than $2 \cdot W_{\rm LB}$ occasionally. Therefore, we will use $D_{\rm LB} = \max\{A + G_{\rm LB},\, 2 \cdot W_{\rm LB}\}$.

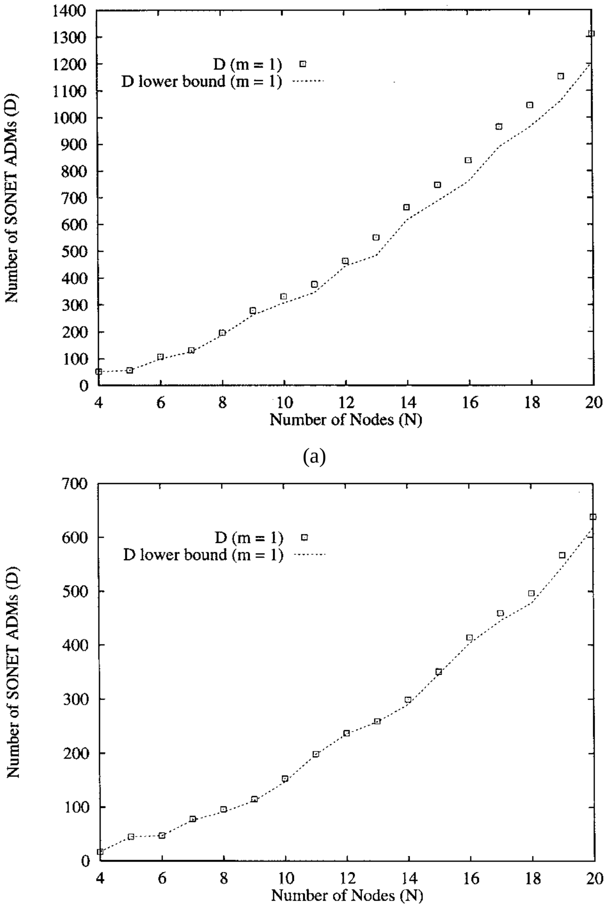
For nonuniform traffic with $m > 1$, we will not compare the number of S-ADMs used with any lower bound on D because a lower bound such as $2 \cdot W_{\rm LB}$ would be too loose to be meaningful.

VI. Circle Grooming

After the circles are constructed using Algorithms I–IV, **Algorithm VI**, shown on the previous page, can be applied to groom multiple (up to m) circles onto each of $W = \lceil C/m \rceil$ wavelengths so as to result in a small D. Since Algorithm VI merely grooms the circles, and does not depend on the traffic pattern or the way the circles are constructed, it is applicable to both unidirectional and bidirectional rings, as well as to both uniform traffic and nonuniform traffic.

Before grooming circles, Algorithm VI first determines the number of circles to be groomed onto each wavelength, i.e., a solution $\{m_w\}$, using either of the following two methods. Method A is to use Algorithm V which identifies the theoretically optimal solution. However, Algorithm V may be time consuming when C and W are large, and it is only applicable to uniform traffic. An alternative is to use Method B, which tends to distribute all the circles as uniformly as possible among W wavelengths. More specifically, when determining the value of m_w, assume that there are X circles left to be groomed. Since at least $W' = \lceil X/m \rceil$ additional wavelengths (including λ_w) will be needed, we may groom $m_w = \lceil X/W' \rceil$ circles onto λ_w. As to be shown, this method is as effective as Method A.

After the solution $\{m_w\}$ is determined, the rest of Algorithm VI uses a heuristic to decide *which* m_w circles are groomed onto wavelength λ_w. The idea is to overlap as many end nodes as possible when grooming circles onto a wavelength (the algorithm pseudo code is self-explanatory). Note that, given the heuristic nature of the algorithm, even if we use Method A for uniform traffic and groom the same number of



(b)

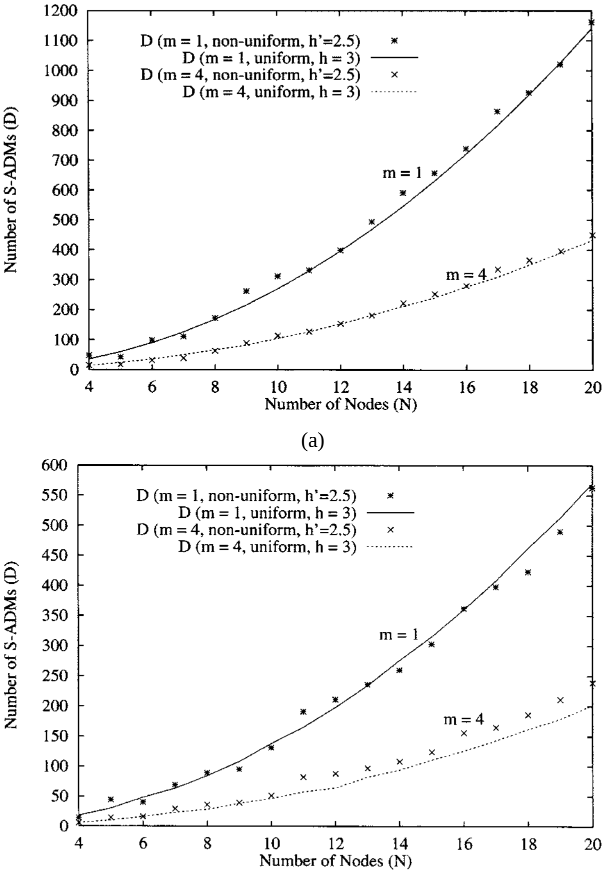
Fig. 5. The number of S-ADMs needed for nonuniform traffic when $h' = 2.5$ and $m = 1$. (a) Unidirectional ring. (b) Bidirectional ring.

circles as that specified by the theoretically optimal solution onto each wavelength, the total number of S-ADMs used by Algorithm VI may still be larger than D_{LB} determined by Algorithm V.

VII. Numerical Results

In this section, we present numerical results on W, D, and their corresponding lower bounds (whenever applicable). The results reported for nonuniform traffic requiring the use of Algorithm IV to construct circles are obtained with the objective to minimize C and W unless otherwise specified.

By default, we assume $4 \leq N \leq 20$. Fig. 3 shows the number of wavelengths required and the corresponding lower bound for both uniform traffic and nonuniform traffic when there is no traffic grooming ($m = 1$). For uniform traffic, only the case where $h = 1$ is shown since if $h > 1$, one may simply multiply both W and W_{LB} by h, as discussed earlier. For nonuniform traffic, we assume that $h_{i,s}$ is evenly distributed between 0 and some maximum value $h_{\max} = \max\{h_{i,s}\}$ with an average of $h' = h_{\max}/2$. Two cases, in which $h' = 2.5$ and 5, respectively, are shown in Fig. 3. As can be seen, W_{LB} is achieved for uniform traffic, and closely approached for nonuniform traffic. Note that in the case of no traffic grooming, we have $W = C$ and $W_{\text{LB}} = C_{\text{LB}}$, hence even with traffic grooming (i.e., $m > 1$), both W and W_{LB} will be $1/m$ of their values shown in the figure according to previous discussion, implying that they will be identical or at least very close to each other.



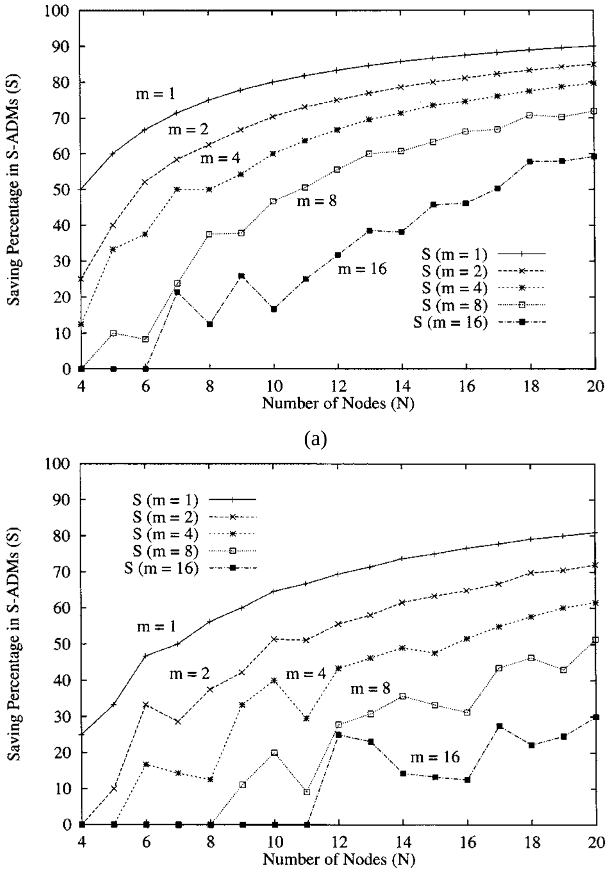
(b)

Fig. 6. Number of S-ADMs needed for uniform traffic ($h = 3$) and for nonuniform traffic ($h' = 2.5$). (a) Unidirectional ring. (b) Bidirectional ring.

Fig. 4 compares the number of S-ADMs used by Algorithm VI with the D_{LB} obtained by Algorithm V for uniform traffic with and without traffic grooming (i.e., $m = 1, 2, 4, 8,$ and 16) and $h = 1$. Recall that we have two different methods to determine the number of circles to be groomed onto each wavelength, but since our results show that these two methods give almost the same performance, we will not distinguish them in this section. As shown in the figure, $D = D_{\text{LB}}$ when $m = 1$, and D is close to D_{LB} when $m \leq 8$ in unidirectional rings. The main reason for $D > D_{\text{LB}}$ when $m > 1$ is that D_{LB} obtained using Algorithm V may not be tight (i.e., achievable) in some cases, especially in bidirectional rings.

Fig. 5 compares the number of S-ADMs used by Algorithm IV with the D_{LB} obtained in Section V-C for nonuniform traffic without traffic grooming ($m = 1$) and $h = 1$. As shown in the figure, D is very close to D_{LB} and even reaches D_{LB} at some points. This (as well as the results shown in Fig. 3) implies that the circle construction algorithm (i.e., Algorithm IV) for nonuniform traffic is very efficient.

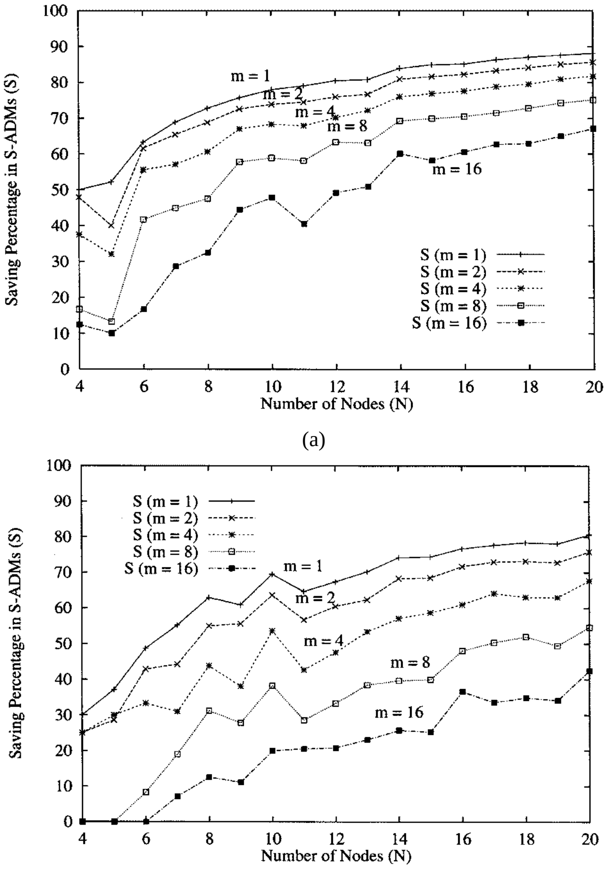
Fig. 6 shows the number of S-ADMs needed when $h = 3$ for uniform traffic and when $h' = 2.5$ for nonuniform traffic. As can be seen, for uniform traffic, as m increases from 1 to 4, D is reduced by about 60% when $h = 3$ (compared to about 50% when $h = 1$ as seen from Fig. 4) for unidirectional as well as bidirectional rings. In addition, when $m = 1$, the number of



(a)

(b)

Fig. 7. Saving percentage in S-ADMs for uniform traffic ($h = 1$). (a) Unidirectional ring. (b) Bidirectional ring.



(a)

(b)

Fig. 8. Saving percentage in S-ADMs for nonuniform traffic ($h' = 2.5$). (a) Unidirectional ring. (b) Bidirectional ring.

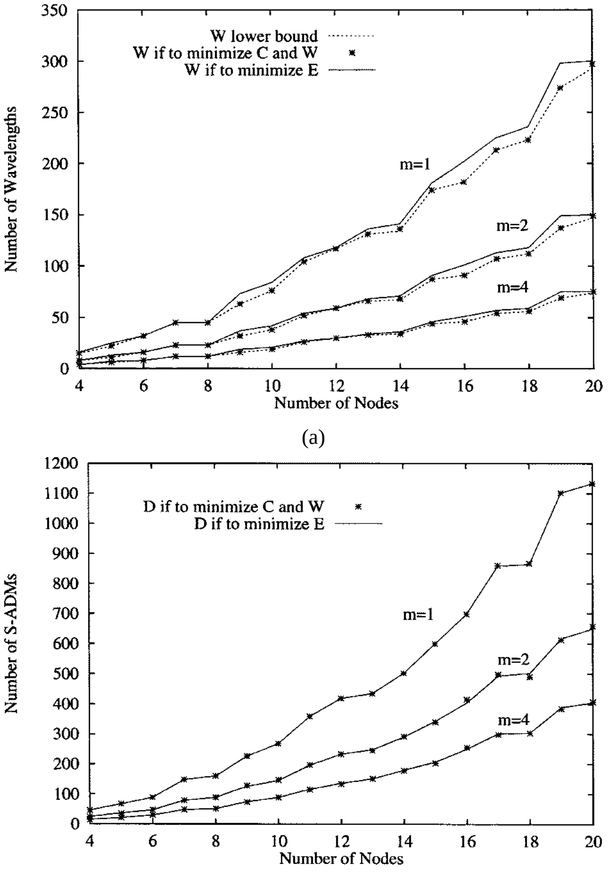
S-ADMs needed when $h = 3$ is exactly 3 times of that needed when $h = 1$. However, if $m = 4$, the number of S-ADMs needed when $h = 3$ is only about 2 times of that needed when $h = 1$. This is because with three copies of each connection, the traffic can be groomed more efficiently. Similarly, if we compare the results in Fig. 6(b) with those in Fig. 9(b) where $h' = 5$ (to be discussed later), we may conclude that for nonuniform traffic, D also increases linearly with h' when $m = 1$, but sublinearly when $m > 1$. From Fig. 6, one can also see that, for nonuniform traffic with $h' = 2.5$, the number of S-ADMs required is close to that for uniform traffic when $h = 3$ (this is because nonuniform traffic usually cannot be groomed as efficiently as uniform traffic).

Figs. 7 and 8 show the saving percentage (S) on the number of S-ADMs due to the proposed traffic grooming algorithms, which is calculated as $S = (N \cdot W - D)/(N \cdot W)$, for uniform (where $h = 1$) and nonuniform traffic (where $h' = 2.5$), respectively. Note that when m increases, the saving percentage decreases for a fixed N because when more circles need to be groomed onto each wavelength, more S-ADMs are involved. As N increases, the saving percentage increases and then saturates gradually. The saving percentage can be as high as 90% in unidirectional rings and 81% in bidirectional rings when $m = 1$ and $N = 20$. Even when $m = 16$, the saving percentage is still significant when $N = 20$ (e.g., about 60% for uniform traffic and 67% for nonuniform traffic in unidirectional rings. The respective percentages in bidirectional rings are 30% and 40%).

Recall that when using Algorithm IV to construct circles for nonuniform traffic, we can minimize either C (and W) or E (and perhaps D). The values of W and D obtained by using these two options, respectively, are shown in Fig. 9 for bidirectional rings (the case for unidirectional rings is similar). As can be seen, when the first option (minimizing C and W) is adopted, the resulting W is nearly the same as W_{LB}, and when the second option (minimizing E) is adopted, a few more wavelengths than W_{LB} are usually required [see Fig. 9(a)]. On the other hand, the two options result in almost the same D [see Fig. 9(b)]. This is because when the objective is to minimize C, a near-minimum C results in a near-minimum $W = \lceil C/m \rceil$ and helps reduce D used by Algorithm VI as well. However, when the objective is to minimize E, which is the total number of end nodes involved in all the circles, it does not necessarily *guarantee* that D used by Algorithm VI will be minimum.

VIII. Conclusion

In this paper, we have proposed a suite of six algorithms that are useful for traffic grooming and wavelength assignment under uniform and nonuniform traffic in both unidirectional and bidirectional SONET/WDM rings. Algorithms I–III are used to construct a minimal number of circles for uniform traffic in unidirectional rings, bidirectional rings with even N, and bidirectional rings (with either odd or even N), respectively. Algorithm IV is used to construct a near-minimum number of circles for



(a)

(b)

Fig. 9. The effect of minimizing C and E for nonuniform traffic in bidirectional rings ($h' = 5$). (a) Number of wavelengths. (b) Number of S-ADMs.

nonuniform traffic. After the circles are constructed, Algorithm VI uses a heuristic to groom up to m circles onto each wavelength, where m is the grooming factor. In addition, Algorithm V is used to determine a lower bound on the number of S-ADMs needed for uniform traffic. The results obtained show that the proposed algorithms perform very well in reducing the number of S-ADMs as well as minimizing the number of wavelengths.

References

[1] J.-C. Bermond *et al.*, "Efficient collective communication in optical networks," in *Proc. ICALP*, 1996, pp. 574–585.

[2] A. Chiu and E. Modiano, "Reducing electronic multiplexing costs in unidirectional SONET/WDM ring networks via traffic grooming," in *Proc. GLOBECOM'98*, Nov. 1998, pp. 322–327.

[3] A. Elrefaie, "Multiwavelength survivable ring network architectures," in *Proc. Int. Conf. Communication*, 1993, pp. 1245–1251.

[4] O. Gerstel, P. Lin, and G. Sasaki, "Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings," in *Proc. INFOCOM'98*, Mar. 1998, pp. 94–101.

[5] O. Gerstel, R. Ramaswami, and G. Sasaki, "Cost effective traffic grooming in WDM rings," in *Proc. INFOCOM'98*, Mar. 1998, pp. 69–77.

[6] E. Modiano and A. Chiu, "Traffic grooming algorithms for minimizing electronic multiplexing costs in unidirectional SONET/WDM ring networks," in *Proc. CISS'98*, Mar. 1998, pp. 653–658.

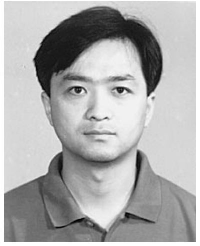
[7] C. Qiao and X. Zhang, "Optimal design of WDM ring networks via resource-balance," *IEEE/LEOS Broadband Optical Networks*, vol. ThB5, pp. 49–50, Aug. 1996.

[8] J. Simmons, E. 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, pp. 48–57, Jan. 1999.

[9] J. Simmons, E. Goldstein, and A. Saleh, "On the value of wavelength-add/drop in WDM rings with uniform traffic," in *OFC'98 ThU3*, Mar. 1998, pp. 361–362.

[10] X. Zhang and C. Qiao, "Scheduling in unidirectional WDM rings and its extensions," in *SPIE Proc. Vol. 3230, All Optical Communication Systems: Architecture, Control and Network Issues III*, 1997, pp. 208–219.

[11] ——, "On scheduling all-to-all personalized connections and cost-effective designs in WDM rings," *IEEE/ACM Trans. Networking*, vol. 7, pp. 435–445, June 1999.



Xijun Zhang received the Ph.D. degree in electrical engineering from the State University of New York at Buffalo (SUNY), Amherst, NY.

His current research interests include optical network architecture, IP/WDM interworking, cost-effective network design, and network control and management. He is a Guest Editor of *Optical Network Magazine Special Issue on IP over WDM*.

Dr. Zhang has served as a Session Chair for the International Conference on Computer Communications and Networks 1999.

Chunming Qiao (S'89–M'92) received the B.S. degree in computer engineering in 1985 from the University of Science and Technology of China (USTC), P. R.O.C., and the Ph.D. degree from the University of Pittsburgh, Pittsburgh, PA, in 1993.

He joined the State University of New York at Buffalo and is currently an Associate Professor in the Computer and Science Engineering Department, as well as an Adjunct Associate Professor in the Electrical Engineering Department. He is also the director of the Lab for Advanced Network Design, Evaluation and Research (LANDER), which conducts cutting-edge research related to optical networks, wireless/mobile networks, and the Internet. He pioneered research on the next-generation Optical Internet, and in particular, optical burst switching (OBS). He has published more than two dozen papers in leading technical journals, in addition to about forty-eight conference papers, and has given several invited lectures and keynote speeches. His research has been sponsored by the National Science Foundation, Alcatel Corporate Research Center, Nokia Research Center, and Telcordia (formerly Bellcore), and the results have been featured in Optical Engineering (OE) Reports, published by SPIE (International Society for Optical Engineering), as well as in the Communication Weeks, an event sponsored by Alcatel. He has served as a Co-Chair for the annual All-Optical Networking Conference since 1997, a Program Vice Co-Chair for the 1998 International Conference on Computer Communications and Networks (IC3N), expert panelist, technical program committee member, and session organizer/chair in several other conferences and workshops. He is also the Chair of the recently established Technical Group on Optical Networks (TGON) sponsored by SPIE, an Editor of the *Journal on High-Speed Networks* (JHSN) and the *Optical Networks Magazine*. He is the Editor-at-Large for the IEEE Communication Society.

He received the Andrew-Mellon Distinguished Doctoral Fellowship Award from the University of Pittsburgh. He is a member of IEEE Computer Society and IEEE Communications Society.