An Effective and Comprehensive Approach for Traffic Grooming and Wavelength Assignment in SONET/WDM Rings[1]

Xijun Zhang† Chunming Qiao‡

† Department of Electrical Engineering

‡ Department of Computer Science and Engineering

University at Buffalo (SUNY)

Buffalo, NY 14260

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 used 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 uni-directional and bidirectional rings having an arbitrary number of nodes under both uniform and non-uniform (i.e. arbitrary) traffic with an arbitrary grooming factor. Several 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 these lower bounds can be closely approached in most cases or even achieved in some cases using the proposed algorithms. 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.

Keywords: SONET, WDM rings, ADMs, traffic grooming, wavelength assignment

[1] This research is supported in part by NSF under contract number MIP-9409864 and ANIR-9801778. A preliminary version appeared in SPIE Proceedings, All Optical Networks, Vol. 3531, pages 221-231, 1998.

1. 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.

In a SONET ring with point-to-point WDM links, each having W wavelengths, every node has W S-ADMs, one for each wavelength, and 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 we can groom the traffic destined to the node 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, Figure 1 shows a node with two different configurations, one at left using 3 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, then the saving percentage on the number of S-ADMs can be defined as $S = \frac{N \cdot W - D}{N \cdot W}$.

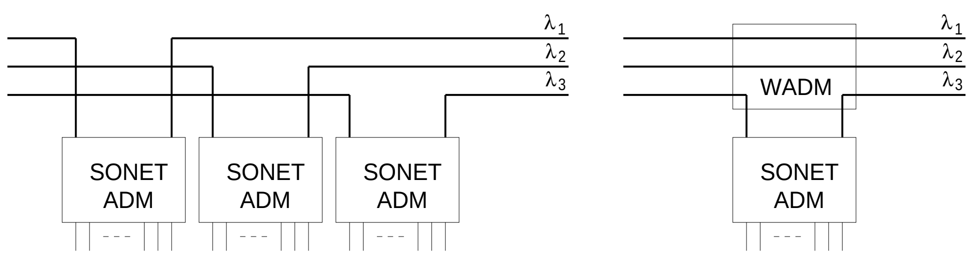


Figure 1. Reducing the number of S-ADMs.

In this paper, we consider cost-effective designs of SONET/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 per link (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 (in addition to assigning wavelengths appropriately). Note that, one may not always be able to minimize both W and D at the same time. Examples in which they cannot be minimized simultaneously are provided in [3] and [5].

The basic idea of our approach is as follows. Let the traffic from one node to another be expressed as a number of connections each requiring a base bandwidth (e.g. OC-3) which is $\frac{1}{M}$ (e.g. $M = 16$) of the bandwidth of a wavelength (e.g. OC-48). Heuristic algorithms based on the scheduling algorithms proposed in [6, 8, 9] are first used to construct as many *circles* as needed to include all the connections, where each circle consists of non-overlapping connections. As to be described later, a circle may be full or partial, and in a full circle, the number of S-ADMs needed is equal to the number of *end nodes* involved, i.e. nodes that are either sources or destinations of the connections forming the circle. After the circles (say C in total) are constructed, another heuristic algorithm is used to groom up to M circles onto each of $W = \lceil \frac{C}{M} \rceil$ wavelengths while trying to overlap as many end nodes belonging to different circles as possible so as to result in a small D.

Note that wavelength assignment is normally a part of the traffic grooming problem. Using our approach, however, wavelength assignment has largely been accomplished in the circle construction phase. In other words, our approach can effectively separate wavelength assignment from traffic grooming, and thus help simplify both problems and obtain efficient solutions. For instance, once the circles are constructed, it has been determined that the connections in each circle will be assigned the same wavelength; the number of wavelengths to be used, W, has also been determined (and possibly minimized). If D needs not to be minimized, one may simply groom arbitrary M circles onto an available wavelength; Otherwise, these circles can be groomed in a more judicious way.

Another major difference between this work and the 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, [3] 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. [5] 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 [7] without specifying the algorithm(s) or heuristic(s) employed. In [4], analytic results (such as D and W required) were presented for several specific optical WDM ring designs under uniform traffic (although a framework allowing non-uniform was also discussed). The SONET/WDM rings considered in this paper differ from all the designs considered in [4]. To our best knowledge, this is also the first paper to report quantitative results for non-uniform traffic grooming.

The rest of the paper is organized as follows. Section 2 deals with uniform traffic, in which optimal circle construction algorithms (i.e. those resulting in a minimum number of circles) are presented for both unidirectional and bidirectional SONET/WDM rings. In addition, for the cases with and without traffic grooming, respectively, lower bounds on the number of S-ADMs required are determined and a circle grooming algorithm is proposed. The latter is also applicable to non-uniform traffic, which is treated in Section 3. Numerical results are presented and discussed in Section 4. Finally, Section 5 concludes the paper.

2. UNIFORM TRAFFIC

In the rest of the paper, we will 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 $M \geq 1$. 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 (say j) that is s hops away (along a shortest path from i to $j = (i + s) \ mod \ N$). Hereafter, such a connection will be said to have a *stride* (or hop count) of s.

In addition, let $R_{i,s}$ denote the total bandwidth required by the traffic from node i to a node s hops away, where $R_{i,s} = H_{i,s} \cdot R_b$ for some integer $H_{i,s} \geq 0$. Since each connection has a base bandwidth of R_b, the number of connections to be established from i to j is $H_{i,s}$. Note that, if $H_{i,s} > M$, these connections have to be groomed onto different wavelengths. Even if $H_{i,s} < M$, these connections may still be groomed onto different wavelengths in order to minimize W and/or D.

In uniform traffic, $R_{i,s}$ is the same for every i and s, and thus we may let $R = R_{i,s}$ and $H = H_{i,s}$. If $R = B$ (or $H = M$), it is natural to *bundle* all H connections from one node to another into a *super-connection* which is then assigned one wavelength. Doing so eliminates the need to groom traffic between different pairs of nodes while fully utilizing the bandwidth of each wavelength. It also effectively increases the base bandwidth by H ($= M$) times. That is, we may define the *effective* base bandwidth (of a super-connection) to be r_b, where $r_b = H \cdot R_b$. Accordingly, we can express B and R in terms of r_b as $B = m \cdot r_b$ and $R = h \cdot r_b$, where $m = h = 1$. In short, having $M = H$ can be treated as equivalent to having $m = h = 1$.

In general, one may bundle $GCD(M, H)$ (where GCD stands for the *greatest common divider*) connections from one node to another into a super-connection, and then groom $m = \frac{M}{GCD(M,H)}$ super-connections onto one wavelength. Since the effective base bandwidth of a super-connection is $r_b = GCD(M, H) \times R_b$, we may express R as $h \cdot r_b$, where $h = \frac{H}{GCD(M,H)}$. For example, let $B = 2.488$ Gbps (OC-48) and $R_b = 155.52$ Mbps (OC-3). If $R = 7 \times 155.52$ Mbps, we have $M = 16$ and $H = 7$, and no bundling can be performed since $GCD(16, 7) = 1$ (and hence, we may set $m = M = 16$, $h = H = 7$ and $r_b = R_b$). However, if $R = 4 \times 155.52$ Mbps (i.e. OC-12), $GCD(16, 4) = 4$ connections can be bundled into one super-connection, effectively making $m = 4$ and $h = 1$ (since $r_b = 4R_b$). If $R = 10 \times 155.52$ Mbps, $GCD(16, 10) = 2$ connections can be bundled, effectively making $m = 8$ and $h = 5$ (since $r_b = 2R_b$). Finally, if $R = 64 \times 155.52$ Mbps (i.e, OC-192), $GCD(16, 64) = 16$ connections can be bundled, effectively making $m = 1$ and $h = 4$ (since $r_b = 16R_b = B$).

In what follows, we assume that bundling is performed whenever possible, and use the term "connection" to refer to "super-connection" as well if bundling does occur. Accordingly, we will focus only on the values of m and h, and define m to be the *grooming factor*. Note that m and h are integers (and $GCD(m, h) = 1$) even though neither $\frac{R}{B}$ nor $\frac{B}{R}$ needs to be an integer. In addition, when $m = 1$, no traffic grooming is needed, and hence each connection will be established as a lightpath. On the other hand, traffic grooming is needed when $m > 1$, which is only possible if $R < B$ (that is, $H < M$), or $R > B$ but R is not a multiple of B (that is, $H > M$ but $GCD(H, M) < M$). We will study how to construct circles using connections and groom circles in unidirectional rings first.

2.1. Unidirectional rings

2.1.1. No traffic grooming ($m = 1$)

In this subsection, we start with the simplest case where the traffic from one node to another requires the full bandwidth of a wavelength (i.e. $R = B$). As a result, $m = h = 1$. In such a case, each node needs to establish one connection to every other node for a total of $N(N - 1)$ connections from all N nodes.

Note that the minimum number of wavelengths needed to establish all these connections, i.e. the lower bound on W, is $W_{LB} = \frac{N(N-1)}{2}$ [8]. Since there are at least 2 S-ADMs on each wavelength carrying (at least) one connection, one for the source and the other for the destination of the connection, the minimum number of S-ADMs, i.e. the lower bound on D, is $D_{LB} = 2 \cdot W_{LB} = N(N-1)$.

These two lower bounds can be achieved by using the following **Algorithm I** (note that the algorithm merely constructs circles since no circle grooming will be needed).

```
Algorithm I: Construct circles in unidirectional rings for uniform traffic

for i = 0, 1, ···, N − 2
    for s = 1, 2, ···, N − 1 − i
        combine (i, s) and (i + s, N − s) in one circle
```

Specifically, the algorithm combines two connections which have common end nodes (e.g. one from i to j and the other from j to i) and thus complementary strides, s and $N-s$, to form a *full circle*. The total number of circles formed is $C = \sum_{i=0}^{N-2}(N-1-i) = \frac{N(N-1)}{2} = W_{LB}$. Once the circles are formed, wavelengths can be assigned arbitrarily (one wavelength to each circle). Given that each circle needs 2 S-ADMs, one for each end node involved in the circle, the total number of S-ADMs is equal to D_{LB}. In general, whenever there is no traffic grooming, the number of S-ADMs needed on each full circle is equal to the number of connections in the circle, and thus is minimized.

Note that the case where $m = 1$ but $h > 1$ 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 sets of C circles will be constructed, and both W and D (as well as W_{LB} and D_{LB}) will be increased by h times. In short, both W and D will be minimized as long as $m = 1$.

2.1.2. With traffic grooming ($m > 1$)

We now consider unidirectional rings under uniform traffic where either $R < B$, or $R > B$ but R is not a multiple of B. In either case, $m > 1$. In order to increase bandwidth utilization (and minimize W), we need to groom multiple (up to m if necessary) circles onto each wavelength. We will consider two cases where $h = 1$ and $h > 1$, respectively. If $h = 1$, only one set of circles which can be constructed using Algorithm I as described earlier needs to be groomed. If $h > 1$, h such sets of C circles need to be groomed. Since Algorithm I results in the minimum number of circles, if we groom as many circles as possible (up to m circles) onto each wavelength, the number of wavelengths needed, W, will be minimized. By carefully selecting the m circles to be groomed onto the same wavelength, the number of S-ADMs can also be reduced.

In the following, we will first consider the case where $h = 1$ and describe an algorithm to determine a lower bound on the number of S-ADMs required, D_{LB}. We then propose an efficient algorithm to groom circles for either $h = 1$ or $h > 1$.

A lower bound on D (D_{LB}) when $h = 1$

Let the number of circles constructed using Algorithm I (or Algorithms IV and V to be described later) be denoted by C. It is clear that $W_{LB} = \lceil \frac{C}{m} \rceil$. Assume that the number of circles groomed onto wavelength λ_w is m_w (where m_w may be different for different λ_w and $1 \leq w \leq W$), we have $\sum_{w=1}^{W} m_w = C$. 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). Since the maximum number of circles that can be constructed using Algorithm I among n end nodes is $\frac{n(n-1)}{2}$ or $\binom{n}{2}$, in order to have m_w circles on λ_w, we need to have $\binom{d(m_w)}{2} \geq m_w$. On the other hand, in order for $d(m_w)$ to be the minimum number of S-ADMs on λ_w, 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 6 S-ADMs are needed in order to groom 11 circles onto a wavelength. Similarly, $d(1) = 2$, $d(2) = 3$, $\cdots$, $d(16) = 7$, and so on. Accordingly, given the values of all m_w's, where $w = 1, 2, \cdots, W$, the minimum number of S-ADMs required is at least $tempD_{LB} = \sum_{w=1}^{W} d(m_w)$. Hereafter, we will call the set of values of m_w's a *solution* and denote it by $\{m_w\}$. In addition, we will call the solution which results in the minimum $tempD_{LB}$ (among all possible solutions) the *theoretically optimal* solution, and set D_{LB} to be equal to the corresponding minimum $tempD_{LB}$.

We now propose the following **Algorithm II** to find the theoretically optimal solution (and D_{LB}) without searching for all possible solutions. The basic idea is as follows. Without loss of generality, we may assume that $1 \leq m_1 \leq m_2 \leq \cdots \leq m_W \leq m$ since the wavelength assignment can be arbitrary. In Algorithm II, function $FindM()$ is called recursively to determine m_w in a wavelength index descending order. That is, it finds m_W first, m_{W-1} second and so on. Accordingly, for a given wavelength index w, at the time m_w is to be determined, all m_k's with $w+1 \leq k \leq W$ 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, $m_w \leq m_{w+1}$, and hence, we have $m_w \leq min\{m_{w+1}, C - tempC - w + 1\}$. On the other hand, since the $C - tempC$ circles which have not groomed so far will be allocated onto w wavelengths among which λ_w will be allocated the largest number of circles, we have $m_w \geq \lceil \frac{C - tempC}{w} \rceil$. 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 ($tempD_{LB}$) is calculated and the best solution with the lowest $tempD_{LB}$ found so far is recorded. At the end of the algorithm, the theoretically optimal solution $\{m_w\}$ and D_{LB} can thus be obtained.

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

```
main() {
    D_LB = N · M; //may set it to any large value
    FindM(W); //start with λ_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 λ_w
    }
    else { // w < W
        tempC = Σ_{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 = ⌈(C−tempC)/w⌉, ⌈(C−tempC)/w⌉ + 1, ···, M_max
            FindM(w − 1);
    }
    else { // groom all remaining circles onto the last wavelength
        m_1 = C − tempC;
        tempD_LB = Σ_{k=1}^{W} d(m_k);
        if (D_LB > tempD_LB) {
            D_LB = tempD_LB;
            save the solution {m_k}
        }
    }
}
```

Circle grooming (and wavelength assignment)

After the circles are constructed using Algorithm I (or other algorithms to be described later), the following **Algorithm III** can be applied to groom multiple (up to m) circles onto each wavelength such that the resulting D can be as close to D_{LB} as possible.

Algorithm III: Groom circles onto W wavelengths

```
//determine the number of circles to be groomed onto each wavelength
if (using Method A)
    invokes Algorithm II 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, ···, 1 {
        w' = ⌈(C − C_g)/m⌉;
        m_w = ⌈(C − C_g)/w'⌉;
        C_g = C_g + m_w;
    }
}
//use a heuristic to groom a pre-determined number of cirlces onto each wavelength
find the number of S-ADMs in each circle;
D = 0;
for w = W, W − 1, ···, 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 λ_w;
    for k = 1, 2, ···, m_w − 1 { // groom other m_w − 1 circles onto λ_w
L:      find a circle which, if groomed onto λ_w, results in a minimum number of
            additional S-ADMs (or maximum overlapping among the end nodes);
        groom this circle onto λ_w;
    }
    D = D + number of S-ADMs on λ_w;
}
```

Note that for a given number of circles, C, Algorithm III will groom these circles onto $W = \lceil \frac{C}{m} \rceil$ wavelengths. Since we can construct a minimum number of circles under uniform traffic in either unidirectional or bidirectional rings (to be discussed later), this means that the algorithm will use a minimum number of wavelengths.

Basically, Algorithm III works as follows. It first determines a solution $\{m_w\}$ using either of the following two methods. Method A is to use Algorithm II which identifies the theoretically optimal solution. However, Algorithm II may be time consuming when C and W are large. In fact, among all the algorithms (to be) described in this paper, Algorithm II is the only one whose worse-case computational time complexity is exponential to N. 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 $w' = \lceil \frac{X}{m} \rceil$ additional wavelengths (including λ_w) will be needed, we may groom $m_w = \lceil \frac{X}{w'} \rceil$ circles onto λ_w. Note that using Method B, we also have $1 \leq m_1 \leq m_2 \leq \cdots \leq m_W \leq m$. After the solution $\{m_w\}$ is determined, the rest of Algorithm III uses a heuristic to decide *which* m_w circles are groomed onto wavelength λ_w. Note that, given the heuristic nature of the algorithm, even if we use Method A and groom the same number of circles as that specified by the theoretically optimal solution onto each wavelength, the total number of S-ADMs used by Algorithm III may still be larger than D_{LB} determined by Algorithm II.

It is clear that Algorithm III also applies to the case where $h > 1$. More specifically, when $h > 1$, all Algorithm III needs to do *differently* is to groom h times more circles (since each circle has h identical copies). As long as additional circles need to be groomed onto a wavelength, a circle identical to the one just groomed will be chosen by the algorithm since in this way, no additional S-ADMS will be needed (see the pseudo code starting at the line labeled with L). Of course, the values of D and W will be different when compared to the case where $h = 1$. In particular, unlike the case for $m = 1$, D will not simply increase by h times due to traffic grooming although we will still have $W = W_{LB} = \lceil \frac{h \cdot C}{m} \rceil$.

Since Algorithm III 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 non-uniform traffic. We will present algorithms to construct circles for bidirectional rings and for non-uniform traffic next.

2.2. Bidirectional rings

In this section, we consider uniform traffic in bidirectional rings (where shortest path routing is assumed). Based on our previous discussion on unidirectional rings, we will only consider the case where $h = 1$ since the case where $h > 1$ is similar.

2.2.1. No traffic grooming ($m = 1$)

As for the case of unidirectional rings, we start by assuming that the traffic from one node to another requires the full bandwidth of a wavelength, and accordingly, $m = h = 1$, and the total number of connections to be established is $N(N-1)$. In [2] (which discussed the case where N is odd only) and [1,6,9], it is given that the minimum number of wavelengths required in a bidirectional ring is:

$$W_{LB} = \begin{cases} \frac{N^2-1}{8} & \text{for odd } N \\ \lceil \frac{N^2}{8} \rceil & \text{for even } N \end{cases} \quad (1)$$

In order to establish these connections using a minimum number of wavelengths, the basic idea of the scheduling algorithms proposed in [6] can be used. Specifically, for an even N, the following

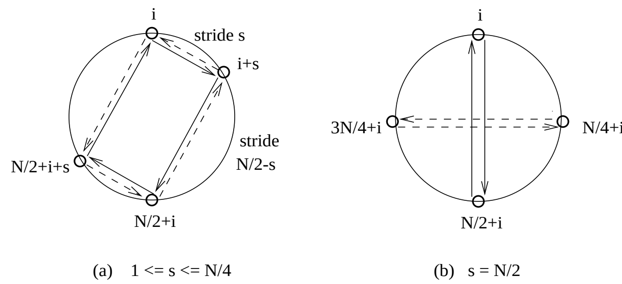


Figure 2. Construct circles in bidirectional rings when N is even.

Algorithm IV combines 4 clockwise connections having two complementary strides s and $\frac{N}{2} - s$ to form a full clockwise circle as shown by the solid lines in Figure 2 (a), and assigns one wavelength to the circle (Figure 2 (b) illustrates the special case where $s = \frac{N}{2}$). The total number of clockwise circles constructed by Algorithm IV is $2 \times (\lfloor \frac{N-2}{4} \rfloor + 1) + \frac{N}{2} \lfloor \frac{N-2}{4} \rfloor$ (if $N = 4n$) or $(\lfloor \frac{N-2}{4} \rfloor + 1) + \frac{N}{2} \lfloor \frac{N-2}{4} \rfloor$ (if $N = 4n+2$), which is equal to $\lceil \frac{N^2}{8} \rceil$ (the same as the minimum number of wavelengths needed given by Eq. 1).

Note that, counter-clockwise connections can be established in the same way as shown by the dashed lines in Figure 2. In general, since uniform traffic can be distributed among the two directions (i.e. clockwise and counterwise) in a symmetrical way based on shortest-path routing, we will only focus on those connections (and circles) that are clockwise in the following discussion.

Algorithm IV: Construct clockwise circles in bidirectional rings for uniform traffic (even N)

for $i = 0, 1, \cdots, \lfloor \frac{N-2}{4} \rfloor$ // $s = \frac{N}{2}$
  combine $(i, \frac{N}{2})$ and $(\frac{N}{2} + i, \frac{N}{2})$ in one circle

for $i = 0, 1, \cdots, \lfloor \frac{N-2}{4} \rfloor$ // $s = \frac{N}{4}$ (this is a special case when $N = 4n$)
  combine $(i, \frac{N}{4})$, $(\frac{N}{4} + i, \frac{N}{4})$, $(\frac{N}{2} + i, \frac{N}{4})$ and $(\frac{3N}{4} + i, \frac{N}{4})$ in one circle

for $i = 0, 1, \cdots, \frac{N}{2} - 1$
  for $s = 1, 2, \cdots, \lfloor \frac{N-2}{4} \rfloor$
    combine (i, s), $(i + s, \frac{N}{2} - s)$, $(\frac{N}{2} + i, s)$ and $(\frac{N}{2} + i + s, \frac{N}{2} - s)$ in one circle

Similarly, when N is odd, the following **Algorithm V** combines either 3 or 4 connections to form a full circle as shown in Figure 3, in which the case where $s = i$ is shown in solid lines and the case where $s = i - 1$ is shown in dashed lines. The total number of circles constructed in Algorithm V is $\sum_{i=0}^{n_1-1}(i+1) = \frac{N^2-1}{8}$ (see [9]), which is also equal to the minimum number of wavelengths needed. In addition, in both Algorithms IV and V, $\frac{N(N-1)}{2}$ connections are established on each direction in *full* circles, and thus the number of S-ADMs used (D) is minimized to $\frac{N(N-1)}{2}$ given that there is no need for traffic grooming.

Algorithm V: Construct clockwise circles in bidirectional rings for uniform traffic (odd N)

$n_1 = \frac{N-1}{2}$;
$n_2 = n_1 + 1 = \frac{N+1}{2}$;
for $i = 0, 1, 2, \cdots, n_1 - 1$ { // the starting node of each circle
  for $s = i, i-1, \cdots, 1$ { // the value of one stride
    construct a circle involving nodes $i, i + n_1 - s, n_2 + i$ and $i - s$, i.e.
    containing connections $(i, n_1 - s), (n_1 + i - s, s + 1), (n_2 + i, n_1 - s)$ and $(i - s, s)$
  }
  // the following is for a special case where $s = 0$
  construct a circle involving nodes $i, i + n_1$ and $n_2 + i$, i.e.
  containing connections $(i, n_1), (n_1 + i, 1)$ and $(n_2 + i, n_1)$

}

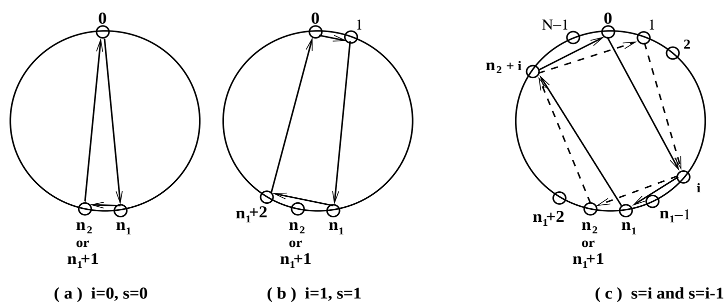


Figure 3. Constructing circles to establish all clockwise connections when N is odd.

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 [4]. 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 [4]. However, in the case to be discussed next, where $m > 1$ 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 [4]).

2.2.2. With traffic grooming ($m > 1$)

Similar to the case for unidirectional rings, after the circles are constructed using Algorithm IV or V in bidirectional rings, Algorithm III can be used to groom up to m circles onto every wavelength (according to the solution $\{m_w\}$ obtained using either Method A or Method B as described in Section 2.1.2).

However, since Algorithm IV or V is now used to construct circles instead of Algorithm I, there are usually more end nodes involved in a circle. Accordingly, $d(m_w)$ needs to be calculated differently. Specifically, the maximum number of circles involving n nodes is now given by:

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

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

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

Note that, a $d(m_w)$ determined here can be less than the actual number of end nodes (S-ADMs) required on each wavelength. This is because, for example, when N is even, the number of end nodes (or connections) involved in each circle constructed using Algorithm IV is either 2 or 4. If two circles are groomed on λ_w (i.e. $m_w = 2$), we have $d(m_w = 2) = 4$ according to the above equation, which is applicable if each of the two circles contains two nodes. However, when each of the two circles contains four nodes, the minimal number of S-ADMs needed could be 8. Accordingly, the lower bound on the required number of S-ADMs determined by Algorithm II based on $d(m_w)$'s will be loose for bidirectional rings.

3. NON-UNIFORM TRAFFIC

In non-uniform traffic, the bandwidth required by the traffic from node i to another node s hops away may vary with i and s. Let H' be the greatest common divider (GCD) of all *non-zero* $H_{i,s}$. As in the case for uniform traffic, we may bundle $GCD(H', M)$ connections into one super-connection, and thus treat the case as having $m = \frac{M}{GCD(H',M)}$ and $h_{i,s} = \frac{H_{i,s}}{GCD(H',M)}$ is the number of connections to be established from node i to another node s hops away. In this section, we propose a heuristic algorithm to construct circles for a given traffic matrix $\{h_{i,s}\}$, which can then be groomed by using Algorithm III described earlier.

Recall from the algorithms described so far for uniform traffic that the following rules seem to be useful in minimizing the number of wavelengths and the number of S-ADMs: in unidirectional rings, if two connections with complementary strides exist, they will be combined in one circle; Similarly, in bidirectional rings, up to four connections with complementary strides will be combined in one circle. For non-uniform traffic, it is natural to first follow these rules when constructing circles. Denote by C_1 the maximum number of circles constructed in this way. Then, we can apply the following heuristic algorithm, **Algorithm VI**, to continue the construction of circles until all connections have been included. This algorithm can be applied to both unidirectional and bidirectional rings. Note that the circles constructed using Algorithm VI may not be full. In a partial circle, there are one or more "gaps" which can not be fit in by any remaining connection to be established, resulting in some bandwidth waste.

In Algorithm VI, we construct 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. Intuitively, a fewer gaps help reduce not only the number of wavelengths (due to a better bandwidth utilization), but also the number of S-ADMs. Accordingly, the algorithm attempts to fit each connection into existing circles without generating an additional gap. If the connection being considered share 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 has been either included in some existing circles, or put into the GapMaker list, we start to process the GapMaker list by using its connections to construct (additional) circles. Note that, it is possible that some connections from the GapMaker list will now fit into some existing circles without creating an additional gap. For example, as shown in Figure 4, connections a and b are "gap makers", and will initially be put into the GapMaker list. If later on, connection x is added on circle 1 because it is not a "gap maker", connection a can be added to circle 1 as well and removed from the GapMaker list.

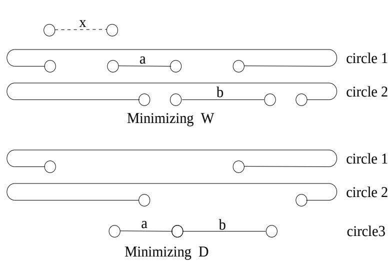


Figure 4. An illustration of the two options in Algorithm IV.

Algorithm VI: Construct circles for non-uniform traffic

```
C_2 = 1; // C_2 is the number of circles constructed by this algorithm so far
s_0 = N − 1 for unidirectional rings or s_0 = N/2 for bidirectional rings; //the maximum stride
for s = s_0, s_0 − 1, ···, 1 { // consider connections with longer strides first
    for i = 0, 1, ···, N − 1 { // one node at a time
        while (h_{i,s} > 0){
            for c = 1, 2, ···, 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 wavelengths) {
    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 S-ADMs) {
    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;
    }
}
```

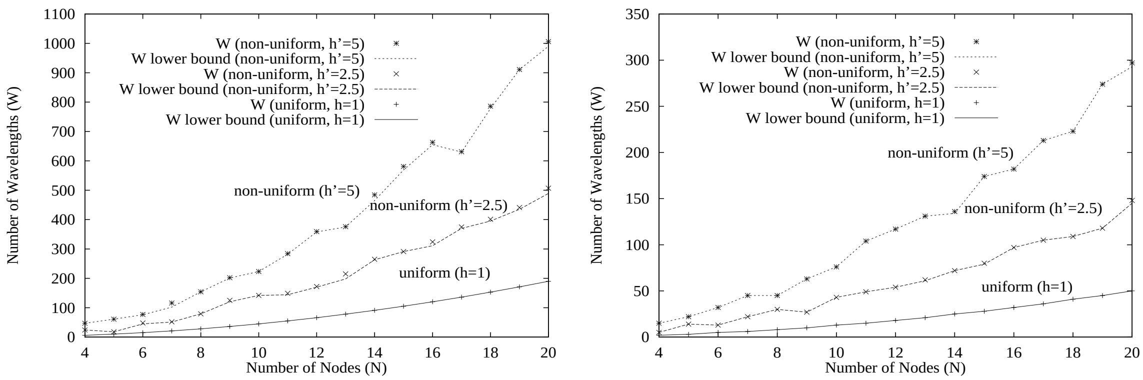
If there are still some connections left in the GapMaker list that cannot be fit into any existing circle without creating an additional gap, we have two options: one is to minimize the number of wavelengths (W) used, and the other is to minimize the number of S-ADMs (D) used. If W is to be minimized, 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 D 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. Figure 4 illustrates the difference between the two options, assuming that connection x does not exist. As can be seen, the first option results in one fewer wavelengths (2) and one more S-ADMs (8) than the second option (when there is no grooming).

The total number of circles constructed is $C = C_1 + C_2$. As mentioned earlier, Algorithm III can then be used to groom these circles on to $W = \lceil \frac{C}{m} \rceil$ wavelengths. To see how effective our approach is in minimizing W, we may, for a given traffic matrix $\{h_{i,s}\}$, determine the traffic load on each and every link. Let the maximum traffic load over all links be R_{max}, we can then use $W_{LB} = \frac{R_{max}}{B}$ as a lower bound on the number of wavelengths required. Note that, for unidirectional rings, one may calculate another lower bound on W as $\frac{\sum_{i=0}^{N-1} \sum_{s=1}^{N-1} h_{i,s} \cdot s}{m \cdot N}$ (for bidirectional rings, a similar formula may be used). However, this lower bound is not as tight as the former and hence will not be used. In addition, with non-uniform traffic, we will not compare the number of S-ADMs used by Algorithm III with any lower bound on D because even a lower bound such as $2 \cdot W_{LB}$ would be too loose to be meaningful, especially when $m > 1$. To obtain a tight lower bound on D in such a case would require exhaustive search for all possible combinations.

4. NUMERICAL RESULTS

In this section, we present numerical results on W, D, and their corresponding lower bounds (whenever applicable). For non-uniform traffic requiring the use of Algorithm VI to construct circles, the results reported here are obtained with the objective to minimize W unless otherwise specified.

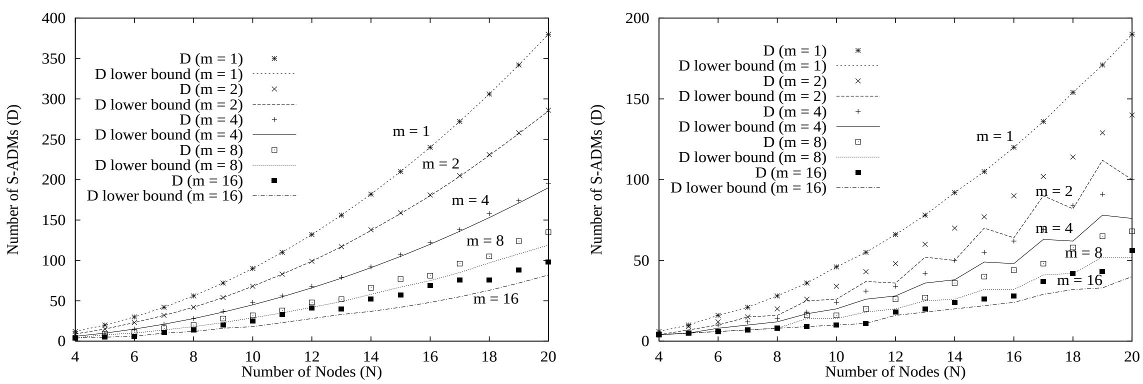
By default, we assume $4 \leq N \leq 20$. Figure 5 shows the number of wavelengths required and the corresponding lower bound for both uniform traffic and non-uniform traffic when there is no traffic grooming ($m = 1$). For uniform traffic, only the case where $h = 1$ is shown since if $h > 1$, one may simplify multiply both W and W_{LB} by h, as discussed earlier. For non-uniform traffic, we assume that $h_{i,s}$ is evenly distributed between 0 and $h_{max} = max\{h_{i,s}\}$ with an average of $h' = \frac{h_{max}}{2}$. Two cases, in which $h' = 2.5$ and 5, respectively, are shown in Figure 5. As can be seen, W_{LB} is achieved for uniform traffic, and closely approached for non-uniform traffic. Note that without traffic grooming, $W = C$. Hence, even with traffic grooming (i.e. $m > 1$), both W and W_{LB} will be $\frac{1}{m}$ of their values shown in the figure according to previous discussion on Algorithm III, implying that they will be identical or at least very close to each other.



(a) Unidirectional Ring

(b) Bidirectional Ring

Figure 5. Wavelength requirement in SONET/WDM rings with no traffic grooming.



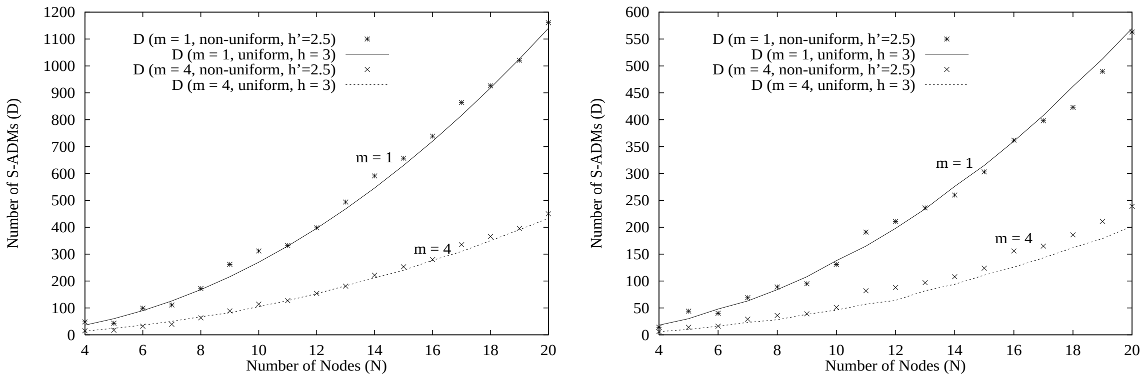
(a) Unidirectional Ring

(b) Bidirectional Ring

Figure 6. The number of S-ADMs needed in SONET/WDM rings for uniform traffic when $h = 1$.

Figure 6 compares the number of S-ADMs used by Algorithm III with the D_{LB} obtained by Algorithm II 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 circle, 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, $D = D_{LB}$ when $m = 1$, and D is close to D_{LB} when $m \leq 8$ in unidirectional rings. The main reason for $D > D_{LB}$

when $m > 1$ is that D_{LB} obtained using Algorithm II may not be tight (i.e. achievable) in some cases, especially in bidirectional rings with even N, as discussed in Section 2.2.2.

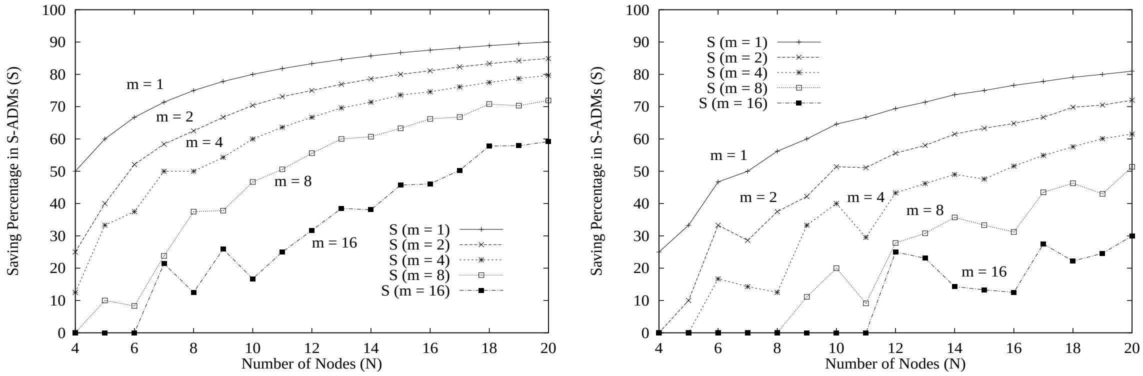


(a) Unidirectional Ring (b) Bidirectional Ring

Figure 7. Number of S-ADMs needed in SONET/WDM rings for uniform traffic ($h = 3$) and for non-uniform traffic ($h' = 2.5$).

Figure 7 shows the number of S-ADMs needed when $h = 3$ for uniform traffic and when $h' = 2.5$ for non-uniform 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 $h = 1$ in Figure 6) for unidirectional as well as bidirectional rings. In addition, when $m = 1$, the number of 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 3 copies for each connection, the traffic can be groomed more efficiently. In addition, one can see from the results that, for non-uniform traffic with $h' = 2.5$, the number of S-ADMs required is close to that for uniform traffic when $h = 3$ (this is because non-uniform traffic usually cannot be groomed as efficiently as uniform traffic). Note that if we compare the results in Figure 7 (b) with those in Figure 10 (b) where $h' = 5$ (to be discussed later), we may conclude that for non-uniform traffic, D also increases linearly with h' when $m = 1$, but sub-linearly when $m > 1$.

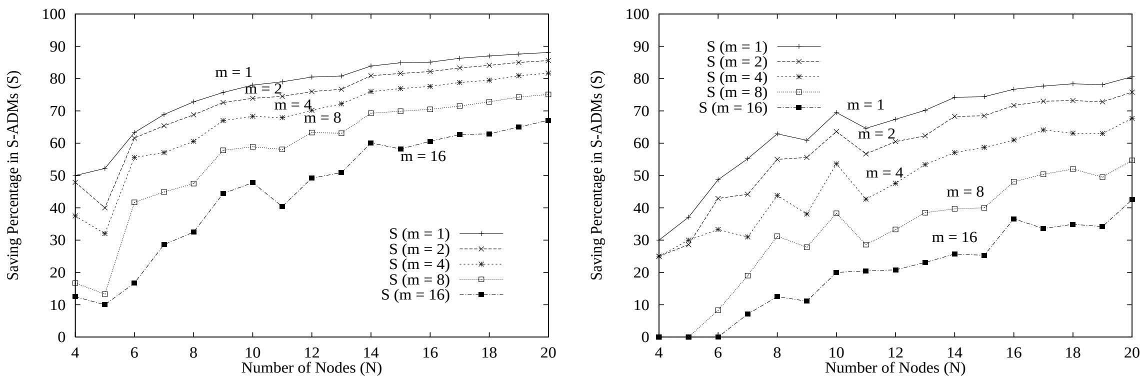
Figures 8 and 9 show the saving percentage (S) on the number of S-ADMs due to the proposed traffic grooming algorithms, which is calculated as $S = \frac{N \cdot W_{LB} - D}{N \cdot W_{LB}}$, for uniform (where $h = 1$) and non-uniform traffic (where $h' = 2.5$), respectively. Note that when m increases, 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 non-uniform traffic in unidirectional rings. The respective percentages in bidirectional rings are 30% and 40%).



(a) Unidirectional Ring

(b) Bidirectional Ring

Figure 8. Saving percentage in S-ADMs for uniform traffic ($h = 1$).

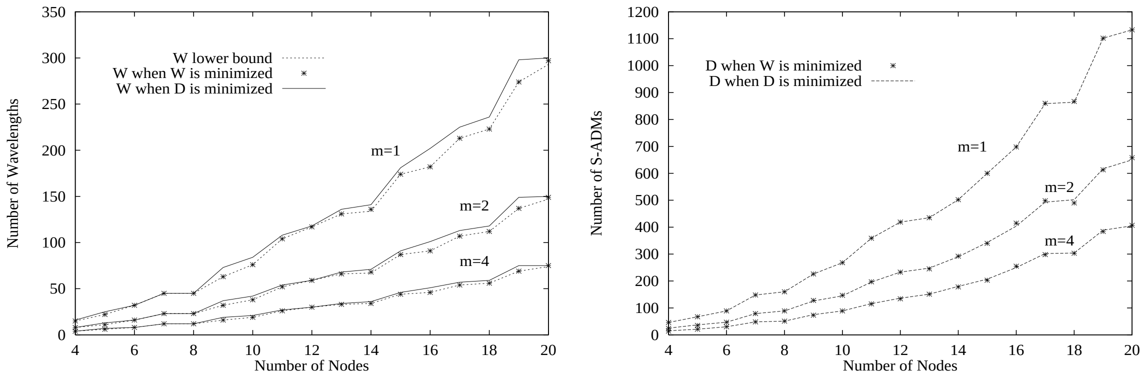


(a) Unidirectional Ring

(b) Bidirectional Ring

Figure 9. Saving percentage in S-ADMs for non-uniform traffic ($h' = 2.5$).

Recall that when using Algorithm VI to construct circles for non-uniform traffic, we can minimize either W or D. The values of W and D obtained by using these two options, respectively, are shown in Figure 10 for bidirectional rings (the case for unidirectional rings is similar). As can be seen, when the first option (minimizing W) is adopted, the resulting W is nearly the same as W_{LB}, and when the second option (minimizing D) is adopted, a few more wavelengths than W_{LB} are usually required (see Figure 10 (a)). On the other hand, the two options result in almost the same D (see Figure 10 (b)). This is because when the objective is to minimize W, the number of circles constructed, C, will be near minimum, which in turn results in a near minimum $W = \lceil \frac{C}{m} \rceil$



(a) Number of Wavelengths

(b) Number of S-ADMs

Figure 10. The effect of minimizing W and D for non-uniform traffic in bidirectional rings ($h' = 2.5$).

and helps reduce D used by Algorithm III as well. However, when the objective is to minimize D, Algorithm VI only *tries* to minimize the total number of S-ADMs (or end nodes) involved in all the circles, which does not necessarily guarantee that D used by Algorithm III will be minimized.

5. CONCLUSION

In this paper, we have proposed a suite of six algorithms that are useful for traffic grooming and wavelength assignment under uniform and non-uniform traffic in both unidirectional and bidirectional SONET/WDM rings. Algorithms I, IV and V are used to construct a minimal number of circles for uniform traffic in unidirectional rings, bidirectional rings with even N, and bidirectional rings with odd N, respectively. Algorithm VI is used to construct a near minimum number of circles for non-uniform traffic. After the circles are constructed, Algorithm III uses a heuristic to groom up to m circles onto each wavelength, where m is the grooming factor. The number of wavelengths needed is minimum if the traffic is uniform and is near minimum otherwise. All the algorithms proposed, except Algorithm II which is used to determine a lower bound on the number of S-ADMs needed for uniform traffic, have a worse-case computational time complexity that is polynomial of N. 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. of ICALP*, pages 574–585, 1996.

2. A. Elrefaie. Multiwavelength survivable ring network architectures. In *Proc. Int'l Conference on Communication*, pages 1245–1251, 1993.
3. O. Gerstel, P. Lin, and G. Sasaki. Wavelength assignment in a WDM ring to minimize cost of embedded SONET rings. In *INFOCOM'98*, pages 94–101, March 1998.
4. O. Gerstel, R. Ramaswami, and G. Sasaki. Cost effective traffic grooming in WDM rings. In *INFOCOM'98*, pages 69–77, March 1998.
5. E. H. Modiano and A. L. Chiu. Traffic grooming algorithms for minimizing electronic multiplexing costs in unidirectional SONET/WDM ring networks. In *CISS'98*, March 1998.
6. C. Qiao and X. Zhang. Optimal design of WDM ring networks via resource-balance. In *IEEE/LEOS Broadband Optical Networks*, ThB5, pages 49–50, August 1996.
7. J. M. Simmons, E. L. Goldstein, and A. A.M. Saleh. On the value of wavelength-add/drop in WDM rings with uniform traffic. In *OFC'98 ThU3*, pages 361–362, March 1998.
8. 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*, pages 208–219, 1997.
9. X. Zhang and C. Qiao. On scheduling all-to-all personalized connections and cost-effective designs in WDM rings. *IEEE/ACM Transactions on Networking*, (accepted for publication). A preliminary version appeared in *SPIE Proc. Vol. 2919, All-optical Communication Systems: Architecture, Control and Network Issues II*, pages 218–229, 1996.