Grooming Multicast Traffic in Unidirectional SONET/WDM Rings

Anuj Rawat, Richard La, Steven Marcus, and Mark Shayman

***Abstract*— In this paper we study the problem of efficient grooming of given non-uniform multicast traffic demands on a unidirectional SONET/WDM ring. The goal is to try to minimize the network cost as given by (*i*) the number of wavelengths required per fiber and (*ii*) the number of electronic Add-Drop Multiplexers (ADMs) required on the ring. The problem is NP hard for both the cost functions. We observe that the problem with cost function (*i*) can be reduced to a corresponding traffic grooming problem for unicast traffic which can then be modelled as a standard circular-arc graph coloring problem. For cost function (*ii*), we construct a graph based heuristic and compare it against the multicast extension of the best known unicast traffic grooming heuristic [1]. We observe that our heuristic requires fewer ADMs than required by the multicast extension of the unicast heuristic given in [1]. We also develop a lower bound and compare it against some upper bounds to study the maximum penalty of not employing intelligent wavelength assignment and/or traffic grooming under the unidirectional SONET/WDM ring scenario.**

***Index Terms*— Graph theory, SONET ring, traffic grooming, wavelength division multiplexing (WDM).**

I. INTRODUCTION

WAVELENGTH Division Multiplexing (WDM) significantly increases the available network bandwidth capacity by delivering data over multiple wavelengths (channels) simultaneously. With each channel operating at a high rate (currently ~ 10 Gb/s) and multiple channels deployed per fiber (currently ~ 320 wavelengths per fiber), very high transmission capacities (currently ~ 3.2 Tb/s) can be achieved. An important issue with such a high capacity network is that it places enormous burden on the electronic switches. Hence, it is hardly surprising that the dominant cost in WDM based networks is the cost of the electronic switching equipment required. Fortunately it is not necessary to electronically process all the incoming traffic at each node since most of the incoming traffic is neither sourced at that node nor destined to it. So to reduce the cost of electronic components at each node, we can selectively drop the wavelengths carrying traffic that requires electronic processing at that node and allow the remaining wavelengths to optically bypass the node.

Typically in WDM based optical networks, the bandwidth available per wavelength is much larger than the bandwidth required per session, and with the advancement of optical technology, it seems likely that this mismatch will continue to grow in the near future. Hence for efficient bandwidth usage, it is prudent to combine several low rate traffic sessions onto a single wavelength. The problem of *effectively* packing lower rate traffic streams onto the available wavelengths in order to achieve some desired goal is called *traffic grooming*. If the traffic demands are known in advance, then the problem is called *static*, otherwise the problem is called *dynamic*. In static traffic grooming, usually the aim is to minimize the overall network cost. Here the network cost includes the cost of electronics (this is the dominant cost) as well as the cost of optics (wavelengths per fiber). In dynamic traffic grooming, the aim is to groom the incoming traffic demands such that the blocking probability is minimum. In this work we are interested in the static traffic grooming problem.

The inherent reliability and high bandwidth capacity of a WDM based Synchronous Optical Network (SONET/WDM) ring has made it the architecture of choice in the current network infrastructure. Typically, in a SONET/WDM ring each wavelength operates at a line rate of OC-N[1] and can carry several low rate OC-M ($M \leq N$) traffic channels using Time Division Multiplexing (TDM). The timeslots on a wavelength are referred to as the *subwavelength channels*.

Electronic Add-Drop Multiplexers (ADMs) are required to add (drop) the subwavelength traffic at the source (destination) node. On receiving a wavelength channel, the ADM, corresponding to that particular wavelength, can add/drop timeslots on the wavelength channel without disrupting the onward transmission of other timeslots on the wavelength. So if a node (say n) does not act as a source or a destination for any traffic on some wavelength (say λ), i.e., if no add/drop of any timeslot on λ is required at n, then there is no need for an ADM corresponding to wavelength λ at node n. Since the cost of the ADMs (electronics) form the bulk of the network cost [2], we can see that intelligent grooming of low-rate traffic onto wavelengths can result in ADM savings, which results in a lower network cost.

Grooming static unicast subwavelength traffic to minimize either the number of ADMs or the number of wavelengths required per fiber in WDM ring networks is a well studied problem [1][2][3]. Different traffic scenarios such as uniform all-to-all traffic [3][4], distance dependent traffic [2] and non-uniform traffic [1][5] have been studied. Work has also been done on other cost functions such as the overall network cost [6], which includes the cost of transceivers, wavelengths and the number of required hops. Recently there has been a lot of work on grooming both static [7] as well as dynamic [8][9][10]

Manuscript received March 7, 2006.
The authors are with the Department of Electrical and Computer Engineering, University of Maryland, College Park, MD (email: {anuj, hyongla, marcus, shayman}@glue.umd.edu).
Digital Object Identifier 10.1109/JSAC-OCN.2007.025806.

[1] OC-N (Optical Carrier-N) is a SONET standard designating a fiber optic circuit operating at N times 51.84 megabits per second, i.e., N times the operating rate of STS-1 (Synchronous Transport Signal-1).

traffic in mesh networks. References [11] and [12] provide an excellent review on the problem of grooming unicast traffic in WDM networks.

While most of the earlier studies of the traffic grooming problem have dealt exclusively with the unicast traffic, it is expected that in the future a sizable portion of the traffic will be multicast in nature. This is mainly because of the increasing demand of multicast services such as multimedia conferencing, video distribution, collaborative processing, etc. Grooming multicast traffic is still an area of active research and although a lot of literature is available, not many results are known. Most of the work in the multicast case has focused on heuristics for grooming multicast traffic in WDM mesh networks under non-uniform static [13] as well as dynamic traffic [14][15][16][17][18][19] scenarios.

Although multicast traffic grooming in mesh WDM networks is a general case of the same problem in WDM rings, the ideas that are applied for mesh networks in [13][14][15][16][17][18][19] are not very attractive for unidirectional rings. The difference between the mesh and unidirectional ring cases is that, in mesh networks there are many possible routings for each traffic demand whereas in unidirectional rings the routing is fixed and we have control over wavelength assignment only. All of the heuristics for grooming multicast traffic in mesh networks take advantage of the multiple routings possible and the wavelength assignment is usually trivial (first fit). This is clearly not desired for grooming in unidirectional rings, since for unidirectional rings the routing is already fixed and the only way to effectively groom traffic is by using intelligent wavelength assignment. Although most of the work on multicast traffic grooming looks at mesh WDM networks, there has been some work in the case of WDM rings also. More specifically, in [20] the authors look at the problem of grooming given multicast traffic demands in a bidirectional WDM ring. They present a heuristic algorithm inspired by the algorithm to groom unicast traffic demands on WDM rings given in [1]. We shall compare this to our work in more detail once we state our exact problem.

In this paper we look at the problem of static grooming of non-uniform multicast traffic on a unidirectional SONET/WDM ring. In general, the SONET ring nodes may or may not have SONET digital-cross connects (DXCs) and wavelength converters. SONET DXCs and wavelength converters are expensive components so in this work we assume that the network nodes do not have wavelength converters and SONET DXCs. Since the ADMs do not have wavelength conversion or timeslot changing functionality, the absence of wavelength converters and SONET DXCs implies wavelength-continuity and timeslot-continuity constraints in the network. This sort of network setup for grooming unicast traffic has been categorized as a *singlehop ring* [21]. In another type of network setup some nodes of the network use SONET DXC to consolidate or segregate subchannels (timeslots on a wavelength). This setup is referred to as a *multihop ring* [21]. Hence in this work we are concerned with the singlehop ring case for grooming multicast traffic.

We consider two different cost functions (*i*) the number of wavelengths required per fiber and (*ii*) the total number of ADMs. We observe that for cost function (*i*), the problem can be modeled as a circular-arc graph coloring problem. Thus, the standard coloring techniques apply. We then suggest a graph based heuristic for cost function (*ii*). We extend the traffic grooming heuristic for non-uniform unicast traffic given in [1] to the multicast case and compare this multicast extension to our heuristic. We also develop a lower bound on the number of ADMs and compare it against some of the upper bounds to get interesting insights into the problem.

The problem that we study here is quite different from the problem studied in [20]. The main difference, other than the fact that we study unidirectional rings while [20] looks at bidirectional rings, is that the cost function used is different. We consider the number of ADMs and the number of wavelengths required per fiber as our cost, whereas in [20], the total number of ports of *e-DAC* nodes in the network is considered as the cost. In [20], the authors define two different types of nodes, *o-DAC* and *e-DAC* nodes. When all the traffic on all the incoming wavelengths needs to be forwarded, o-DAC nodes are used since the splitting can be done in the optical domain. If this is not the case then e-DAC nodes are used. Note that the cost functions are not the same since we require ADMs at all the nodes where some traffic needs to be dropped whereas in [20], even the nodes where there is some drop traffic can be treated as o-DAC nodes. Another important difference is that in [20], the authors consider a multihop ring whereas we look at a singlehop ring.

The rest of the paper is organized as follows. In Section II, we state our assumptions on the network, the traffic and the node architecture. Here we also state the precise problem. In Section III we model the problem (for both the cost functions) using graphs. We present our heuristics in Section IV. In Section V, we develop and study some lower and upper bounds. In Section VI, we present the complexity analysis for the grooming schemes introduced in this work. Section VII presents the simulation results. Finally Section VIII concludes the paper. For quick reference, Table I lists some of the improtant symbols and notations used in the paper along with brief explainations.

II. Problem Statement

A. Physical Network

The physical network is assumed to be a clockwise unidirectional WDM ring with N nodes numbered $0, 1, \ldots, N-1$ distributed on the ring in the clockwise direction as shown in Figure 1(a). We assume that there is a single fiber between adjacent nodes, which can support W wavelengths given by $\lambda_0, \lambda_1, \ldots, \lambda_{W-1}$ and the capacity of each wavelength is assumed to be C units.

Also, as noted in Section I, we assume that the network nodes do not have wavelength converters and SONET DXCs. This implies timeslot-continuity and wavelength-continuity constraints in the network.

B. Assumptions On Traffic

We assume that there are M given multicast traffic requests denoted by $R_0, R_1, \ldots, R_{M-1}$. Every multicast request specifies a source node and a set of destination nodes. We assume that each multicast request is for r units of traffic. Also, the

TABLE I
LIST OF IMPORTANT SYMBOLS

Symbol	Stands for
N	number of nodes in the SONET ring
M	number of multicast sessions
g	grooming ratio
N'	number of nodes which act as a source or a destination for at least one multicast request
W_{min}	minimum number of wavelengths required per fiber
$G = (V, E)$	contention graph with vertex set V representing the multicast requests
χ	chromatic number of graph G
$G[C_i]$	subgraph induced by vertex set $C_i \subseteq V$ (representing the multicast requests groomed on wavelength λ_i) on the contention graph G
$\chi(G[C_i])$	chromatic number of graph $G[C_i]$
S_v	set consisting of source and destinations for the multicast traffic request corresponding to vertex $v \in V$
$\sigma = \{C_0^\sigma, C_1^\sigma, \dots\}$	partition of vertex set V into nonoverlapping clusters such that $\bigcup_i C_i^\sigma = V$, $C_i^\sigma \bigcap C_j^\sigma = \emptyset$ and $\chi(G[C_i^\sigma]) \leq g$
Σ	set of all valid partitioning
$G_i \equiv G[V_i]$	subgraph induced by vertex set $V_i \subseteq V$ (representing the multicast requests which contain network node i as source or some destination) on the contention graph G
χ_i	chromatic number of graph G_i
k_i	number of multicast sessions which contain network node i as source or some destination
α_i	number of multicast sessions which contain network node i as an intermediate destination
β_i	number of multicast sessions which contain network node i as the source
γ_i	number of multicast sessions which contain network node i as the final destination
$\hat{G}_i \equiv G[\hat{V}_i]$	subgraph induced by vertex set $\hat{V}_i \subseteq V$ (representing the multicast requests which contain network node i as source or final destination) on the contention graph G
w_i	maximum width of interval graph $\hat{G}_i$
z_i	size of a $i-th$ multicast session R_i
z_{min}	minimum possible size of multicast sessions
z_{max}	maximum possible size of multicast sessions
z_{avg}	average size of multicast sessions
$\mathcal{F}$	c.d.f. according to which sizes of multicast sessions are distributed
$\mu_{\mathcal{F}}$	mean of c.d.f. $\mathcal{F}$
U_{wc}	worst case upper bound on the number of ADMs
U_{Algo-A}	upper bound on the number of ADMs required by Algorithm A
U_{Algo-B}	upper bound on the number of ADMs required by Algorithm B
L	lower bound on the number of ADMs
L_2, L_3	other lower bounds on the number of ADMs

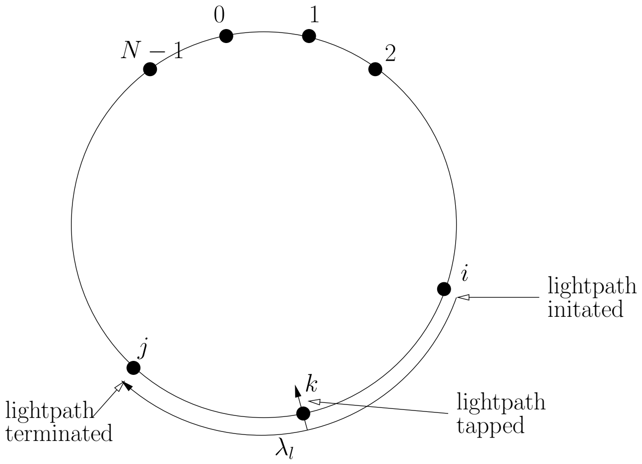
wavelength capacity C is assumed to be an integral multiple of the required traffic rate r, i.e., $C = g \times r$. We refer to g, the number of subwavelength multicast demands that can be groomed on a single wavelength channel, as the *grooming ratio*. Assuming an integral grooming ratio g is justified in case of SONET rings, since in SONET networks the capacity of a single wavelength (OC-192, OC-48, etc.) is usually an exact multiple of the bandwidth requirement for an individual traffic request (OC-3, OC-12, etc.). Note that for SONET rings, we can assume without loss of generality that r is equal to the capacity of individual subwavelength channels (timeslots). Hence, the grooming ratio g is equal to the number of subwavelength channels available on each wavelength.

With the above traffic model we can consider multicast requests of different bandwidth requirements also. The important requirement is that each request should be splittable into individual multicast requests of granularity r.

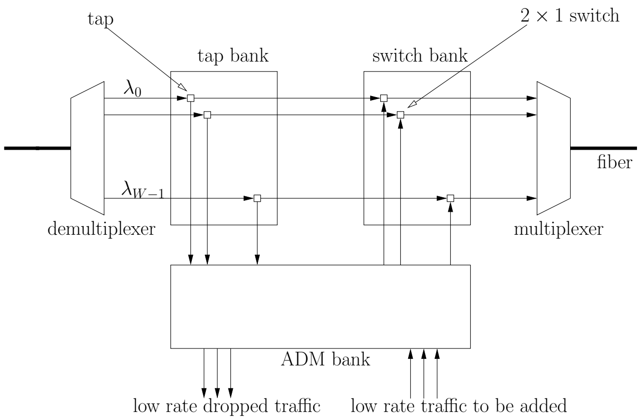
C. Node Architecture

Most of the current work on multicast traffic in optical networks uses multicast capable nodes called Splitter-and-Delivery nodes and multicast incapable nodes with drop-and-continue capability. In this work, since we are looking at ring networks, we do not require the nodes to split the incoming light over multiple outgoing links to form light-trees. This is because here the light-trees are just arcs on the ring as shown in Figure 1(a), and the nodes with drop-and-continue capability suffice. The architecture for the drop-and-continue capable nodes used in the ring network is similar to the Tap-and-Continue node architecture first given in [22]. In fact the Tap-and-Continue nodes used in this paper are much simpler than the nodes in [22], since here we only have a single incoming and outgoing fiber per node and therefore no optical switching is required.

As shown in Figure 1(a), if a lightpath is set-up between nodes i and j on wavelength λ_l, and traffic from i to an



(a) Unidirectional SONET ring with tap-and-continue nodes.



(b) Tap-and-continue node architecture.

Fig. 1. Network and node model.

intermediate node k is also groomed on λ_l or i has to send the same traffic to k (this is the case when i is the source and j and k are the destinations of a multicast traffic request), then instead of terminating the lightpath at k, we can drop a small amount of light of wavelength λ_l at k to extract the required data packets and let the rest of the light flow through, i.e., we can *tap* the lightpath at any intermediate node. It should be clear that if we want to add (groom) some subwavelength traffic on wavelength λ_l at node k, then we have to tear down the lightpath at k, carry out the grooming and then set-up a new lightpath on wavelength λ_l at node k. Note that in case we are sending different traffic to nodes j and k from node i on a single lightpath (by tapping the lightpath at intermediate node k), then using the above scheme, after passing node k there would be some unnecessary traffic on the lightpath (traffic sent from i to k). Clearly there is no such bandwidth wastage when we are sending the same traffic to both the nodes j and k.

The Tap-and-Continue node architecture that we consider in this paper is shown in Figure 1(b). First the incoming light is split into individual wavelength channels using a demultiplexer. Then each wavelength channel passes through a tap bank. Here we have an option of tapping a small amount of power from the wavelength and sending it to the ADM bank to separate it into its constituent lower rate components. The switch bank consists of 2×1 switches for each wavelength. If no new traffic is added on a wavelength, it is allowed to pass through the switch but if a new lightpath is being initiated at some wavelength then the signal coming from the ADM banks consisting of the groomed traffic is switched forward. Finally the wavelengths are combined using a multiplexer and sent over the outgoing fiber.

Note that at any node, in the ADM bank, we require the ADMs only for wavelengths which are being processed at that particular node, i.e., we require ADMs for all the wavelengths which correspond to the lightpaths that are being dropped or terminated or initiated at that node.

D. Objective

The objective is to minimize the network cost. As noted in Section I, the cost of the network is equivalent to the cost of the network components and the dominating cost among all the components is the cost of ADMs. We also noted that another cost function that is usually considered is the number of wavelengths required per fiber. In this work we study the problem of traffic grooming under the following two objectives.

(*i*) Minimize the number of wavelengths required per fiber.

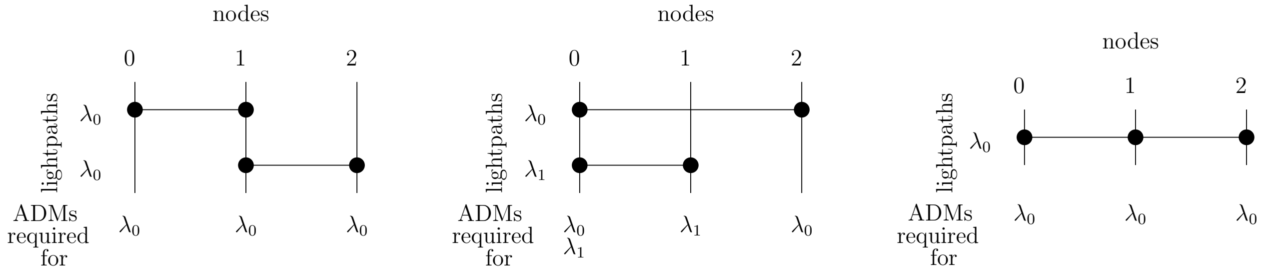
(*ii*) Minimize the total number of ADMs required in the network.

If we count the number of logical hops required by (possibly multihop) paths between all the source-destination pairs, then this gives us an estimate of the *size* of the logical network. Since O-E-O conversions introduce delay[2], the size of the network provides us with a measure of the overall delay in the system. Usually we also want to reduce the delay and therefore the size of the network. An important advantage of using Tap-and-Continue nodes instead of regular nodes is that we can achieve reduction in the network size.To show this we consider the following example. Let us assume that we have a multicast request R having network node 0 as its source and network nodes 1 and 2 as its destinations. If we use regular nodes in the ring then we can configure the lightpaths in either of the two ways listed below.

(*i*) Set up a light path between nodes 0 and 1 and another lightpath between nodes 1 and 2. To save ADMs and conserve the wavelengths used we can use the same wavelength λ_0 for both the lightpaths. Figure 2(a) depicts this configuration. Note that this requires a total of 3 ADMs and 1 wavelength. The network size achieved by this configuration is equal to 3 logical hops (1 for s-d pair $0-1$ and 2 for s-d pair $0-2$).

(*ii*) Set up a light path between nodes 0 and 2 on wavelength λ_0 and set up another lightpath on a different

[2]O-E-O conversions introduce the dominant delay in the network.



(a) Regular nodes (configuration I): requires 3 ADMs, 1 wavelength, 3 logical hops.

(b) Regular nodes (configuration II): requires 4 ADMs, 2 wavelengths, 2 logical hops.

(c) TaC nodes: requires 3 ADMs, 1 wavelength, 2 logical hops.

Fig. 2. Advantage Of using TaC nodes.

wavelength λ_1 between nodes 0 and 1. This configuration is shown in Figure 2(b). It is clear that since both the lightpaths are carrying the same traffic, we are wasting bandwidth in this scenario. This configuration requires a total of 4 ADMs and 2 wavelengths. But now the network size is equal to 2 logical hops (1 hop each for both the s-d pairs).

In case we use Tap-and-Continue nodes in the ring, we can simply setup a single lightpath on wavelength λ_0 between nodes 0 and 2, and tap this lightpath at node 1. Figure 2(c) depicts this configuration. Here we need 3 ADMs, 1 wavelength and the network size is 2 logical hops (1 hop each for both the s-d pairs[3]). Thus we can simultaneously achieve reduction in the number of ADMs, number of wavelengths and the network size. Therefore it makes sense to employ Tap-and-Continue nodes rather than regular nodes on the ring.

III. Modeling

A. Minimizing Wavelengths

First we look at the case where the objective is to minimize the number of wavelengths required per fiber irrespective of the number of ADMs used in the network.

Since we assume the network to be a clockwise unidirectional ring, each traffic request can be treated as an arc on the ring starting from the source and going clockwise through the intermediate destinations (drop points) up to the *final* destination (termination point). Now the wavelength and timeslot continuity constraint implies that each arc (traffic request) should be assigned one subwavelength channel. So if any two multicast requests share some fiber, i.e., the corresponding arcs overlap, then they cannot be groomed on the same subwavelength channel (although they can still share the same wavelength channel). We use this observation to model the problem of minimizing the number of wavelengths per fiber as a graph coloring problem. Consider a graph $G = (V, E)$ where $V = \{v_0, v_1, \ldots, v_{M-1}\}$ is the set of vertices[4] with each vertex v_i representing a multicast request R_i and there is an edge $v_i v_j \in E$ if and only if the multicast requests R_i and R_j share some fiber, i.e., the arcs corresponding to requests R_i and R_j overlap. The graph G is refered to as the *contention graph* because the adjacent vertices in G represent the traffic requests which cannot be groomed on the same subwavelength channel. Now the problem of assigning subwavelength channels to the multicast requests such that we need the minimum number of wavelengths per fiber is equivalent to coloring the vertices of the contention graph G with the minimum number of colors such that no two adjacent vertices share the same color. This is the standard *minimum graph coloring* problem. Note that here each color signifies a subwavelength channel and not a complete wavelength.

We denote the minimum number of colors required for coloring contention graph G, also known as the *chromatic number* of the graph, by χ. Since the minimum number of subwavelength channels required to groom the given traffic requests is equal to χ and since each wavelength contains g subwavelength channels, the minimum number of wavelengths required per fiber is given by

$$W_{min} = \left\lceil \frac{\chi}{g} \right\rceil \tag{1}$$

An interesting observation is that minimizing the number of wavelengths required per fiber is independent of the fact that we are looking at multicast traffic. This is because we are modelling the traffic requests as arcs on a circle and this model only preserves the information about the source and the final sink of the traffic requests. So if we consider the given traffic requests to be unicast with source and sink nodes the same as the source and final sink nodes of the multicast requests $R_0, R_1, \ldots, R_{M-1}$, then we obtain the same contention graph G and therefore the same solution for minimizing the number of wavelengths required per fiber.

Another observation is that the graph G belongs to the family of *circular arc graphs* [23]. Since minimum coloring of circular arc graphs is NP complete [24] and any instance of minimum arc graph coloring can be reduced to the traffic grooming problem under study, grooming multicast (or unicast) traffic on a unidirectional ring to minimize the number of wavelengths required per fiber is NP complete.

[3]We only require 1 logical hop for s-d pair $0-2$ because the traffic reaching from node 0 to node 2 remains in optical domain at the intermediate node 1, i.e., it does not undergo O-E-O conversion anywhere on the path from the source to the destination.

[4]Note that in this work, we refer to the nodes on the SONET ring as *nodes* and the nodes of any graph used in the problem modelling (such as G, where each node represents a traffic request) as *vertices*.

B. Minimizing ADMs

Now we consider the case when the objective is to minimize the ADMs required in the network under the timeslot and wavelength continuity constraint.

For modeling this problem we again represent multicast requests as arcs on the ring and construct the contention graph $G = (V, E)$ as described in Section III-A. Also to each vertex $v_i \in V$ (corresponding to multicast request R_i), we assign a set S_{v_i} consisting of the source and the destinations for request R_i.

Consider the vertex set $C_i \subseteq V$ representing all the multicast traffic requests groomed on wavelength λ_i. Note that the contention graph corresponding to the traffic requests represented by C_i is exactly equal to $G[C_i]$, the subgraph induced by vertex set C_i on the contention graph G. Now as described in Section III-A, the minimum number of sub-wavelength channels required to groom the traffic requests represented by C_i is given by $\chi(G[C_i])$, the chromatic number of the contention graph corresponding to the particular set of traffic requests. So the traffic requests represented by C_i can be groomed on a single wavelength only if $\chi(G[C_i]) \leq g$, i.e., the subgraph $G[C_i]$ induced on the contention graph G by the vertex set C_i is *g-colorable*. Also in this case the number of ADMs corresponding to wavelength λ_i required in the network is equal to $|\bigcup_{v \in C_i} S_v|$.

So given a set of multicast traffic requests modeled by the contention graph $G = (V, E)$, any valid traffic grooming can be modeled as a partitioning $\sigma = \{C_0^\sigma, C_1^\sigma, \ldots\}$ of the vertex set V into non-overlapping clusters $C_0^\sigma, C_1^\sigma, \ldots$ such that $\bigcup_i C_i^\sigma = V$, $C_i^\sigma \bigcap C_j^\sigma = \emptyset$ for all $i \neq j$ and $G[C_i^\sigma]$ (subgraph induced by vertex set C_i^σ on contention graph G) is g-colorable for all i. Also, the cost (number of ADMs required in the network) corresponding to partitioning σ is given by

$$\sum_i |\bigcup_{v \in C_i^\sigma} S_v| \tag{2}$$

Now let Σ denote the set of all such partitionings. Then our problem reduces to finding the partitioning $\sigma \in \Sigma$ which minimizes the required number of ADMs as given in (2).

Note that since the problem of grooming unicast traffic on unidirectional rings in order to minimize the number of ADMs required is NP complete [2], and the multicast traffic grooming is at least as hard as the unicast case, grooming multicast traffic on unidirectional rings to minimize the ADMs is NP hard.

IV. Heuristics

A. Minimizing Wavelengths Per Fiber

As described in Section III-A, minimizing the number of wavelengths required per fiber can be modeled as a circular arc graph coloring problem. The contention graph to be colored is obtained as described in Section III-A and the minimum number of wavelengths required is given by (1). Although circular arc coloring is NP complete [24], there are several approximation algorithms [25][26] available in the literature. Kumar et. al. [25] give a randomized algorithm with approximation ratio $(1 + 1/e + o(1))$ for instances of the problem needing at least $\omega(\ln(n))$ colors, where n is the number of arcs to be colored. In [26], Karapetian et. al. present a $3/2$ approximation algorithm for circular arc coloring. Either of these two algorithms can be used to color the contention graph in our problem. Since these algorithms suffice for minimizing the number of wavelengths required per fiber and they have already been well studied in the literature, we will not discuss this cost function any further in this paper. Now we go on to the more interesting problem of minimizing the total number of ADMs required in the network.

B. Minimizing ADMs

We consider a *graph based* heuristic approach to minimize the number of ADMs required in the SONET ring. The basic idea of the heuristic is to start off by assigning different wavelengths to each of the multicast sessions. Now at every step of the algorithm we update the wavelength assignment by selecting two wavelengths and assigning a single wavelength to all the multicast requests previously assigned to these two wavelengths. Obviously we can *combine* a wavelength pair in this manner only if all the corresponding multicast sessions can indeed be groomed on a single wavelength. The order in which the wavelength pairs are considered for combination is based on the fact that if the multicast sessions assigned to two wavelengths share several source/destination nodes, then we can save a lot of ADMs by using a single wavelength for all these sessions.

In more detail, we first construct the graph $G = (V, E)$ and determine the set S_v corresponding to each node $v \in V$, as discussed in Section III-B. Now let $H(n) = (\Lambda(n), L(n))$ be the weighted graph representing the wavelength assignment after n steps of the heuristic. Here the vertex set $\Lambda(n)$ represents the wavelengths and corresponding to each wavelength $\lambda_i \in \Lambda(n)$ we have a set of multicast requests $C_i \subseteq V$ to which this wavelength is assigned. With slight abuse of notation let $S_{\lambda_i} = \bigcup_{v \in C_i} S_v$ denote the set of nodes which act as source or destination for any multicast session being groomed on wavelength λ_i. Now there is edge $\lambda_i\lambda_j \in L(n)$ if the multicast requests corresponding to the two wavelengths have some common source/destination nodes and the weight of the edge is given by $c(\lambda_i\lambda_j) = |S_{\lambda_i} \bigcap S_{\lambda_j}|$. Note that we can combine any two wavelengths λ_i and λ_j if the subgraph induced by the node set $C_i \bigcup C_j$ on the contention graph G is g-colorable. If this is so then the two wavelengths are said to be *reducible*. Also note that if we combine wavelengths λ_i and λ_j into a new wavelength λ_k, then the number of ADMs required by wavelength λ_k is

$$|S_{\lambda_k}| = |S_{\lambda_i} \bigcup S_{\lambda_j}| = |S_{\lambda_i}| + |S_{\lambda_j}| - c(\lambda_i\lambda_j) \tag{3}$$

But now after combining wavelengths λ_i and λ_j, we no longer need any ADMs on these two wavelengths, therefore we end up saving $|S_{\lambda_i}| + |S_{\lambda_j}| - |S_{\lambda_k}| = c(\lambda_i\lambda_j)$ ADMs. So at the $(n+1)$th step we determine the reducible wavelength pair $\lambda_\alpha, \lambda_\beta \in \Lambda(n)$ such that for any reducible wavelength pair $\lambda_i, \lambda_j \in \Lambda(n)$, $c(\lambda_\alpha\lambda_\beta) \geq c(\lambda_i\lambda_j)$. If there is more than one such wavelength pair, let the set of all such wavelength pairs be $\hat{\Lambda}$. Now we pick the wavelength pair $\lambda_\alpha, \lambda_\beta \in \hat{\Lambda}$ such that for any wavelength pair $\lambda_i, \lambda_j \in \hat{\Lambda}$, $|S_{\lambda_\alpha} \bigcup S_{\lambda_\beta}| \leq |S_{\lambda_i} \bigcup S_{\lambda_j}|$. This is motivated by the fact that if $|S_{\lambda_i}|$ is large for wavelength λ_i then there is a high chance of having a larger cost edge incident on this vertex at some later iteration in the algorithm, so we may not want to use wavelength λ_i in this step for a smaller ADMs saving. If there are still some ties left, then we select any wavelength pair from the possible choices. Now we update the graph $H(n)$ to graph $H(n+1)$ by replacing vertices $\lambda_\alpha, \lambda_\beta$ with a single vertex $\lambda_{\alpha\beta}$ and recomputing all the edges and weights. By this we mean that we contract the edge between vertices $\lambda_\alpha, \lambda_\beta$ into the new vertex $\lambda_{\alpha\beta}$ with $S_{\lambda_{\alpha\beta}} = S_{\lambda_\alpha} \bigcup S_{\lambda_\beta}$. Clearly the only edges affected are the edges that were incident on either λ_α or λ_β.

We continue until there is no reducible wavelength pair. It is clear that the maximum number of iterations is bounded by the number of multicast requests, since initially the number of wavelengths is equal to the number of multicast requests, and each iteration reduces the number of wavelengths by one.

Algorithm 1 Minimizing ADMs

Require: Graph $G = (V, E)$ and for every $v \in V$, set S_v as described in Section III-B.

Ensure: $\min_{\sigma\in\Sigma} \sum_i |\bigcup_{v \in C_i^\sigma} S_v|$
where $\sigma = \{C_0^\sigma, C_1^\sigma, \dots\}$ is a valid partition of vertex set V as described in Section III-B and Σ is the set of all such valid partitionings.

Construct graph $H(0) = (\Lambda(0), L(0))$ with $\Lambda(0) = V$ and evaluate the edge weights $c(\lambda_i\lambda_j)$ for every edge $\lambda_i\lambda_j \in L(0)$.
$condition \Leftarrow TRUE$
while $condition$ is $TRUE$ **do**
  Determine the reducible wavelength pair $\pi = \lambda_\alpha\lambda_\beta \in L(n)$ having the largest edge weight among all the reducible wavelength pairs. If there are several such pairs λ_i, λ_j, then select one with minimum value of $|S_{\lambda_i} \bigcup S_{\lambda_j}|$. If there are still ties then pick any of the possible choices randomly.
  if $\exists$ such π **then**
    Construct graph $H(n+1)$ from graph $H(n)$ as described in Section IV-B and update the edge weights.
  else
    $condition \Leftarrow FALSE$
  end if
end while

Note that the circular arc graphs form an intersection class of graphs [27] and are therefore closed under induced subgraph [27]. Also, as described in Section III-A, graph G belongs to the family of circular arc graphs, so any induced subgraph of graph G is also a circular arc graph. And since coloring circular arc graphs is NP hard, to check whether an induced graph of graph G is g-colorable or not (this is what we need to check in order to determine whether a given wavelength pair is reducible or not) is also NP hard. So instead of doing this we color the induced subgraph using Tucker's algorithm for coloring circular arcs [23] and see if we need more than g colors. Clearly this is sub-optimal but we use this since in general Tucker's algorithm gives a good bound on the chromatic number.

The complete heuristic is given as Algorithm 1.

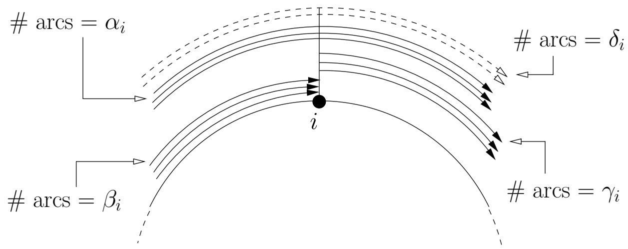


Fig. 3. Bounds: observe each network node separately.

V. Bounds

A. Lower Bound

To get a lower bound on the total number of ADMs required, we consider each node of the network separately. We look at all the multicast requests having network node $i \in \{0, \dots, N-1\}$ as either the source or one of the destinations and try to use as few ADMs as possible on the network node i to support these requests. To do this we construct graph $G_i = (V_i, E_i)$ where $V_i = \{v \in V : i \in S_v\}$ is the set of vertices corresponding to multicast requests having network node i as source or one of the destinations and there is an edge $v_j v_k \in E_i$ if and only if the corresponding multicast requests R_j and R_k share some fiber. Now G_i is exactly $G[V_i]$, the subgraph induced by vertex set $V_i \subseteq V$ on contention graph G described in Section III-A. Let χ_i be the chromatic number of graph G_i. Since all the requests represented by the vertex set V_i have network node i as either source or some destination, node i must have ADMs corresponding to all the wavelengths on which any of these requests are groomed. So in order to use the minimum number of ADMs at node i (irrespective of the number of ADMs required at other nodes), we need to groom the traffic requests represented by vertex set V_i on as few wavelengths as possible. Using the argument from Section III-A, the set of multicast requests that have network node i as either the source or a destination must be spread over at least $\lceil \frac{\chi_i}{g} \rceil$ wavelengths, and hence there must be at least this many ADMs at node i. Now applying a similar lower bound on the number of ADMs required at each node of the network, we see that the total number of ADMs required in the network is at least as large as L given by

$$L = \sum_{i=0}^{N-1} \left\lceil \frac{\chi_i}{g} \right\rceil \tag{4}$$

where N is the number of nodes in the SONET ring and g is the grooming ratio. Thus (4) gives a lower bound on the total number of ADMs required in the network.

As noted in Section IV-B, the class of circular arc graphs is closed under induced subgraph. Also, as described in Section III-A, graph G belongs to the class of circular arc graphs. So the subgraph G_i, induced on the graph G by the vertex set V_i is also a circular arc graph. And, although it is NP complete to determine the chromatic number of a general circular arc graph, as described next, the chromatic number χ_i of graph G_i can be easily calculated.

Figure 3 shows all the multicast traffic requests that pass through network node i. The solid arcs correspond to the requests that contain node i as source or one of the destinations and the dotted arcs represent the traffic requests that pass through node i but do not contain it as source or a destination. Of these, let α_i be the number of requests that contain i as an intermediate destination, β_i be the number of requests that contain i as the final destination and γ_i be the number of requests that contain i as the source. Clearly in the graph G_i, the traffic requests represented by the vertex set V_i correspond to the solid arcs and therefore $k_i = \alpha_i + \beta_i + \gamma_i$. Let us look at graph $\hat{G}_i = (\hat{V}_i, \hat{E}_i)$ instead of graph G_i, where $\hat{G}_i$ is the subgraph induced on graph G_i (or equivalently on contention graph G) by the vertex set $\hat{V}_i$ representing the requests that either have network node i as source or the final destination, i.e., we are removing the α_i nodes corresponding to the requests that contain network node i as an intermediate destination. Now graph $\hat{G}_i$ is an *interval graph* [28] and we know that for interval graphs the chromatic number is easily computable and is equal to the maximum *width* of the graph [29]. Here the width of an interval graph at some point is defined as the number of arcs overlapping at that point. Also it is clear that if vertex $u \in V_i$ and vertex $v \in V_i \setminus \hat{V}_i$, then u and v can not have the same color (since requests corresponding to vertices u and v share some fiber in the network). Therefore the chromatic number χ_i of graph G_i is given by

$$\chi_i = w_i + \alpha_i \tag{5}$$

where w_i is the maximum width of the interval graph $\hat{G}_i$ and $\alpha_i = |V_i \setminus \hat{V}_i|$ is the number of traffic requests which contain i as an intermediate destination.

During our simulations presented in Section VII, we observe an interesting property of the lower bound presented above. It seems that the lower bound given in (4) does not depend on the number of nodes in the ring. To explain this, we try to calculate the expected value of the lower bound for grooming M multicast traffic requests on a SONET ring having N nodes and grooming ratio g. Let z_i represent the *size* of the i-th multicast session R_i. Here by the size of a session, we mean the total number of source and destination nodes in that session. Therefore, the size of session R_i (represented by vertex $v_i \in V$) is given by $z_i = |S_{v_i}|$. For the purpose of our simulations (and hence for this analysis), we assume that the multicast session sizes $z_0, z_1, \ldots, z_{M-1}$ are independent and identically distributed according to some cumulative distribution function $\mathcal{F}$ having mean $\mu_{\mathcal{F}}$. We also assume that the nodes (acting as the source or any destination) in any multicast session, are selected randomly and uniformly from all the nodes of the ring, i.e., for every multicast request R_i (represented by vertex $v_i \in V$) having size z_i, the probability that node $j \in S_{v_i}$ is equal to $\frac{z_i}{N}$ for every $j \in \{0, \ldots, N-1\}$. Moreover, the selection of source and destination nodes of different multicast sessions is assumed to be independent of each other.

We first note that even though it is hard to estimate the expected value of the lower bound, we can estimate the expected value of the following function which approximates the lower bound.

$$\hat{L} = \sum_{i=0}^{N-1} \left\lceil \frac{k_i}{g} \right\rceil \tag{6}$$

Here k_i is the number of multicast sessions that have node i as either source or some destination. When graphs G_i are dense (which is the case in Section VII as well as the case in most of the interesting examples), $\chi_i \approx k_i$. Now it is easy to observe that the expected value of the approximate lower bound $\hat{L}$ is given by

$$\mathbb{E}(\hat{L}) = \mathbb{E}\left(\sum_{i=0}^{N-1} \left\lceil \frac{k_i}{g} \right\rceil\right) = \sum_{i=0}^{N-1} \mathbb{E}\left(\left\lceil \frac{k_i}{g} \right\rceil\right) = N \cdot \mathbb{E}\left(\left\lceil \frac{k}{g} \right\rceil\right) \tag{7}$$

Here for the third equality we are using the fact that in any multicast session, nodes are selected with equal probability, and hence k_i's are identically distributed. So we can drop the subscript i and assume that the number of multicast requests that have i as either source or some destination is distributed according to random variable k.

To get an estimate of $\mathbb{E}(\hat{L})$, we first observe that

$$\mathbb{E}\left(\frac{k}{g}\right) \le \mathbb{E}\left(\left\lceil \frac{k}{g} \right\rceil\right) \le \mathbb{E}\left(\frac{k}{g} + 1\right) \tag{8}$$

Also, the number of multicast sessions selecting a particular network node as source or one of the destinations can be written as

$$k = x_0 + x_1 + \cdots + x_{M-1} \tag{9}$$

where random variable x_i takes value 1 if the i-th multicast session R_i selects the node under consideration as source or one of the destinations and 0 otherwise. Now we can evaluate $\mathbb{E}(k)$ as

$$\begin{aligned} \mathbb{E}(k) &= \sum_{i=0}^{M-1} \mathbb{E}(x_i) = \sum_{i=0}^{M-1} \mathbb{E}(\mathbb{E}(x_i|z_i)) \\ &= \sum_{i=0}^{M-1} \mathbb{E}\left(\frac{z_i}{N}\right) = \frac{1}{N}\sum_{i=0}^{M-1} \mathbb{E}(z_i) = \frac{M}{N}\mu_{\mathcal{F}} \end{aligned} \tag{10}$$

Here the third equality follows from the fact that given z_i (the size of session R_i), the random variable x_i is distributed according to a Bernoulli trial with probability of success $p = \frac{z_i}{N}$. Now (10) gives us

$$\mathbb{E}\left(\frac{k}{g}\right) = \frac{M\mu_{\mathcal{F}}}{Ng} \tag{11}$$

and

$$\mathbb{E}\left(\frac{k}{g} + 1\right) = \frac{M\mu_{\mathcal{F}}}{Ng} + 1 \tag{12}$$

Using equations (7), (8), (11) and (12), we can easily bound the required expectation as

$$\frac{M\mu_{\mathcal{F}}}{g} \le \mathbb{E}(\hat{L}) \le \frac{M\mu_{\mathcal{F}}}{g} + N \tag{13}$$

Now if $M\mu_{\mathcal{F}}/g \gg N$ (which is the case in Section VII and is typically the case), then from (13), we note that the expected value of our lower bound can be approximated by $M\mu_{\mathcal{F}}/g$, which is independent of the number of nodes in the ring. It should be clear that this is mainly because our lower bound looks at each node of the ring in isolation. If we start considering pairs (or triplets, etc.) of nodes at a time then our bound will depend on the number of nodes in the ring. But it is not trivial to extend the given bound and for the purpose of our discussion the given bound suffices.

B. Upper Bounds

Now we investigate some upper bounds on the number of ADMs required in the network. We study the upper bounds for the worst case and two very simple algorithms.

1) Worst Case: The maximum number of ADMs is required when we use a different wavelength for each multicast request, i.e., we do no traffic grooming and wavelength reuse. In this case, the number of ADMs required U_{wc}, is given by

$$U_{wc} = \sum_{i=0}^{N-1} k_i \tag{14}$$

where N is the number of nodes in the SONET ring and k_i is the number of traffic requests having network node i as source or one of the destinations. Thus (14) gives an upper bound on the number of ADMs required.

Trivially in (5), the value of maximum width w_i is lower bounded by

$$w_i \geq \max\{\beta_i, \gamma_i\} \tag{15}$$

Now using (5) and (15) we get

$$\chi_i \geq \alpha_i + \max\{\beta_i, \gamma_i\} \tag{16}$$

We also know that

$$k_i = \alpha_i + \beta_i + \gamma_i \tag{17}$$

Using (16) and (17), we can easily show that

$$\begin{aligned} k_i = \alpha_i + \beta_i + \gamma_i &\leq \alpha_i + 2\max\{\beta_i, \gamma_i\} \\ &\leq 2(\alpha_i + \max\{\beta_i, \gamma_i\}) \leq 2\chi_i \end{aligned} \tag{18}$$

Now (18) and (14) give

$$U_{wc} = \sum_{i=0}^{N-1} k_i \leq \sum_{i=0}^{N-1} 2\chi_i \leq 2g \sum_{i=0}^{N-1} \left\lceil \frac{\chi_i}{g} \right\rceil = 2gL \tag{19}$$

This means that any sort of wavelength assignment (and traffic grooming) solution is an approximation algorithm with approximation ratio $2g$. An interesting observation is that in case of no grooming ($g = 1$), any wavelength assignment will be within twice the optimal as far as the number of ADMs required in the network is concerned.

2) Algorithm A: Another interesting bound that we consider is for the simple heuristic in which we randomly group the traffic requests into clusters of g requests each. We assume that requests in a particular cluster are routed on the same wavelength. This is clearly possible since we are providing a separate subwavelength channel for each traffic request. Then we assume that each network node that acts as a source or destination for some multicast request is provided with an ADM for all these wavelengths. Note that if network node i does not act as source or destination for any multicast request, i.e., if $i \notin S_v$ for every $v \in V$, then since no traffic is being added or dropped at i, there is no need to equip i with ADM on any wavelength. Let N' denote the number of nodes that act as source or destination for at least one multicast request. Clearly $N \geq N'$. Now the number of ADMs in the network is given by

$$U_{Algo-A} = N' \left\lceil \frac{M}{g} \right\rceil \tag{20}$$

Let z_{avg} be the average size of multicast sessions, i.e., let

$$\sum_{i=0}^{N-1} k_i = z_{avg} M \tag{21}$$

Now (18), (20) and (21) give us

$$\begin{aligned} U_{Algo-A} = N' \left\lceil \frac{\sum_{i=0}^{N-1} k_i}{g z_{avg}} \right\rceil &\leq N' \left\lceil \frac{2\sum_{i=0}^{N-1} \chi_i}{g z_{avg}} \right\rceil \\ \leq N' \sum_{i=0}^{N-1} \left\lceil \frac{\chi_i}{g} \right\rceil &= N'L \leq NL \end{aligned} \tag{22}$$

The second inequality holds because of the fact that $z_{avg} \geq 2$. This is true since every multicast session has at least one source and one destination.

Note that if we further assume a large enough average multicast session size, then we can show a better upper bound. More specifically, we can show the following

$$\begin{aligned} U_{Algo-A} &\leq N' \left\lceil \frac{2\sum_{i=0}^{N-1} \chi_i}{g z_{avg}} \right\rceil \leq \frac{N'}{z_{avg}} \left\lceil \frac{2\sum_{i=0}^{N-1} \chi_i}{g} + z_{avg} \right\rceil \\ &\leq \frac{2N'}{z_{avg}} \left\lceil \sum_{i=0}^{N-1} \frac{\chi_i}{g} \right\rceil + N' \leq \frac{2N'}{z_{avg}} \sum_{i=0}^{N-1} \left\lceil \frac{\chi_i}{g} \right\rceil + N' \\ &\leq \left(\frac{2N'}{z_{avg}} + 1 \right) \sum_{i=0}^{N-1} \left\lceil \frac{\chi_i}{g} \right\rceil = \left(\frac{2N'}{z_{avg}} + 1 \right) L \\ &\leq \left(\frac{2N}{z_{avg}} + 1 \right) L \end{aligned} \tag{23}$$

Here the last inequality is due to the fact that if node i acts as a source or a destination for at least one multicast request, then the graph G_i has at least one vertex and therefore $\chi_i \geq 1$. Now observing that there are N' such nodes, we get $\sum_{i=0}^{N-1} \left\lceil \frac{\chi_i}{g} \right\rceil \geq N'$.

So the simple heuristic of routing any g traffic requests on the same wavelength is an approximation algorithm with approximation ratio N. And if the average session size $z_{avg} \geq \frac{2N}{N-1}$, then, for the same heuristic, we can get a better approximation ratio equal to $1 + \frac{2N}{z_{avg}}$.

3) Algorithm B: Another simple heuristic is that we try to use the minimum number of subwavelength channels for all the traffic requests and then we randomly combine g subwavelength channels into one wavelength. So now we may have more than g requests in one wavelength. This is equivalent to coloring the graph G described in Section III-B using the minimum number of colors and then grouping g colors (subwavelength channels) together to form one wavelength. Let the chromatic number of graph G be χ. As mentioned in Section III-B, G is a circular-arc graph and therefore it can be colored by Karapetian's algorithm [26] using at most $\lfloor \frac{3}{2}\chi \rfloor$ colors. Again, as for $Algorithm\ A$, the maximum number of ADMs required by this technique is when each node that acts as a source or a destination for at least one multicast request, is provided with an ADM for all the wavelengths. The number of ADMs is given by

$$U_{Algo-B} = N' \left\lceil \frac{\lfloor \frac{3}{2}\chi \rfloor}{g} \right\rceil \tag{24}$$

Now if all the sessions are multicast with size at least z_{min}, then the minimum number of ADMs required for each wavelength is z_{min}. Hence a lower bound (other than our primary lower bound L given in (4)) on the number of ADMs required in the network is

$$L_2 = z_{min} \left\lceil \frac{\chi}{g} \right\rceil \tag{25}$$

Using equations (24) and (25), we see that

$$\begin{aligned} U_{Algo-B} &= N' \left\lceil \frac{\lfloor \frac{3}{2}\chi \rfloor}{g} \right\rceil \leq N' \left\lceil \frac{2\chi}{g} \right\rceil \leq 2N' \left\lceil \frac{\chi}{g} \right\rceil \\ &\leq \frac{2N'}{z_{min}} L_2 \leq \frac{2N}{z_{min}} L_2 \end{aligned} \tag{26}$$

So the approximation ratio of this simple algorithm is $\frac{2N}{z_{min}}$.

We can arrive at a different (better in some cases) approximation ratio by following a separate line of analysis. Let

$$\frac{\chi}{g} = 2n + \delta + \epsilon \tag{27}$$

where n is a non-negative integer, $\delta \in \{0, 1\}$ and $0 \leq \epsilon < 1$. From (25) and (27), we get

$$L_2 = z_{min} \lceil 2n + \delta + \epsilon \rceil = z_{min}(2n + \delta + \lceil \epsilon \rceil) \tag{28}$$

Again from (24) and (27), we get

$$\begin{aligned} U_{Algo-B} &= N' \left\lceil \frac{\lfloor \frac{3}{2}\chi \rfloor}{g} \right\rceil \leq N' \left\lceil \frac{3\chi}{2g} \right\rceil \\ &= N' \left\lceil \frac{3}{2}(2n + \delta + \epsilon) \right\rceil \\ &\leq N' \left(3n + 3 \left\lceil \frac{\delta + \epsilon}{2} \right\rceil \right) \end{aligned} \tag{29}$$

Now only the following two cases are possible.

- $\delta + \epsilon = 0 \Rightarrow \delta = \epsilon = 0$
  In this case, from (28), the lower bound becomes
  $$L_2 = 2nz_{min} \tag{30}$$
  And from (29) and (30), we get
  $$U_{Algo-B} \leq 3nN' = \frac{3N}{2z_{min}} L_2 \tag{31}$$
- $\delta + \epsilon > 0$
  In this case, from (28) we get
  $$L_2 = z_{min}(2n + \delta + \lceil \epsilon \rceil) \geq z_{min}(2n + 1) \tag{32}$$
  And from (29), we get
  $$U_{Algo-B} \leq N' \left(3n + 3 \left\lceil \frac{\delta + \epsilon}{2} \right\rceil \right) = N'(3n + 3) \tag{33}$$
  where the second equality is based on the fact that since $\delta \in \{1, 0\}, \epsilon \in [0, 1)$ and $\delta + \epsilon > 0$, $0 < \delta + \epsilon < 2$. Now using equations (32) and (33), we get
  $$\begin{aligned} U_{Algo-B} &\leq \frac{3N'}{2}(2n + 1) + \frac{3N'}{2} \\ &\leq \frac{3N'}{2z_{min}} L_2 + \frac{3N'}{2} \\ &\leq \frac{3}{2} \left(\frac{N'}{z_{min}} + 1 \right) L_3 \\ &\leq \frac{3}{2} \left(\frac{N}{z_{min}} + 1 \right) L_3 \end{aligned} \tag{34}$$
  where L_3 is another lower bound on the number of required ADMs given by
  $$L_3 = \max\{L_2, N'\} \tag{35}$$
  It should be clear that L_3 is a valid lower bound because L_2 is a lower bound and since we need at least one ADM on all N' nodes that act as a source or a destination for at least one multicast request, N' is also a valid lower bound on the number of ADMs required in the network.

From equations (31) and (34), we observe that the algorithm has an approximation ratio $\frac{3(N+z_{min})}{2z_{min}}$. Also note that this approximation ratio is better than the previously computed approximation ratio of $\frac{2N}{z_{min}}$ whenever $z_{min} < \frac{N}{3}$.

VI. Complexity Analysis

In this section we present the complexity analysis for our graph based traffic grooming heuristic presented in Section IV-B and the two simple schemes presented as upper bounds for the grooming problem in Section V-B.

A. Algorithm A

Algorithm A described in Section V-B.2 starts by randomly grouping the given traffic requests into clusters of size g each. This clustering requires $O(M)$ steps and we get $\left\lceil \frac{M}{g} \right\rceil$ clusters. Let the clusters be $C_0, \ldots, C_{\lceil \frac{M}{g} \rceil - 1}$. All the traffic requests in cluster C_i are routed on wavelength λ_i, therefore the set of network nodes which should be equipped with ADM on wavelength λ_i is given by

$$S_{\lambda_i} = \bigcup_{v \in C_i} S_v \tag{36}$$

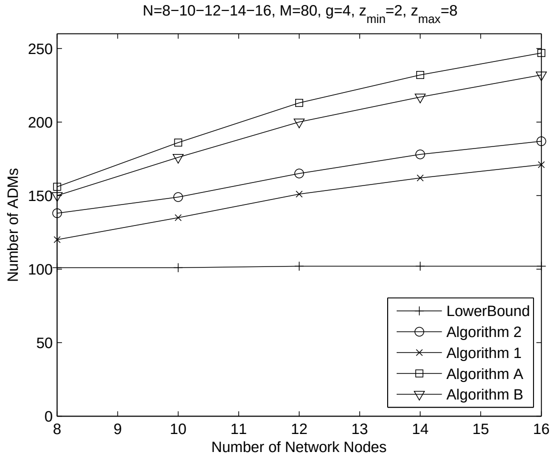
There are $\left\lfloor \frac{M}{g} \right\rfloor$ clusters containing g traffic requests and $\left\lceil \frac{M}{g} \right\rceil - \left\lfloor \frac{M}{g} \right\rfloor$ clusters containing $M \bmod g$ traffic requests. Since $S_v \subseteq \{0, \ldots, N-1\}$ for every $v \in V$, the number of steps required for evaluating S_{λ_i} according to (36) is Ng if cluster C_i contains g requests and $N(M \bmod g)$ if it contains $M \bmod g$ requests. So the total number of steps required for determining the placement of ADMs at each network node on all the wavelengths is

$$\begin{aligned} &\left\lfloor \frac{M}{g} \right\rfloor Ng + \left(\left\lceil \frac{M}{g} \right\rceil - \left\lfloor \frac{M}{g} \right\rfloor \right) N \left(M \bmod g \right) \\ &= \left\lfloor \frac{M}{g} \right\rfloor Ng + N \left(M \bmod g \right) = NM \end{aligned} \tag{37}$$

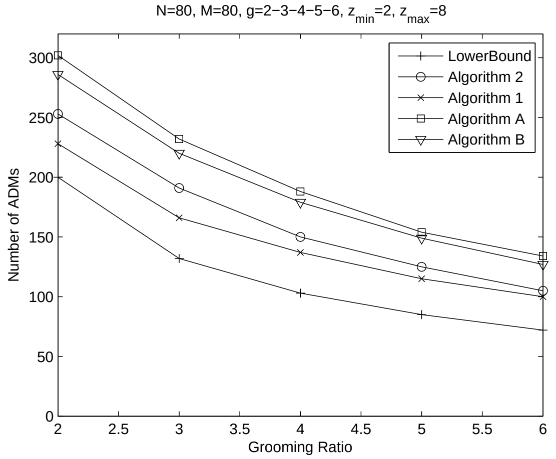
Therefore the overall complexity of Algorithm A is $O(NM)$.

B. Algorithm B

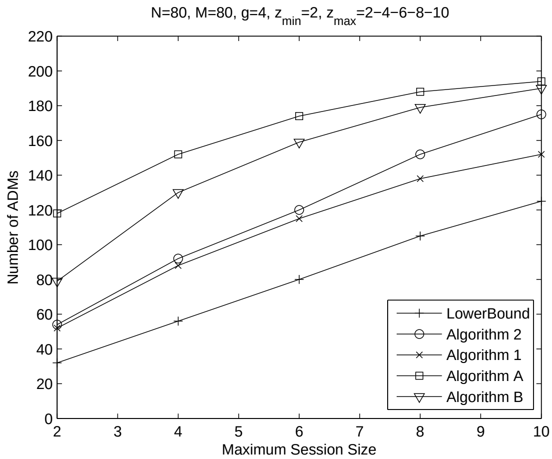
Algorithm B described in Section V-B.2 first colors the contention graph G defined in Section IV-B using Karapetian's algorithm [26]. This requires $O(M^2)$ time. The total number of colors used is upper-bounded by $\max\{\chi, M\}$ where χ is the chromatic number of graph G. The the colors are then randomly split into groups of size g. Based on these grouping the given traffic requests are partitioned into clusters such that the cluster corresponding to a particular group of colors contains all the traffic requests that were assigned colors from



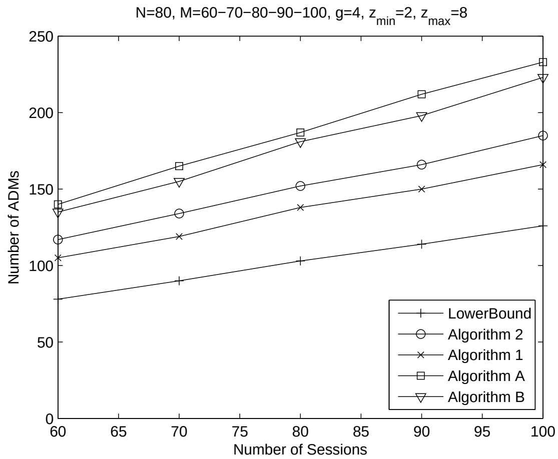
(a) Varying network size.



(b) Varying grooming ratio.



(c) Varying session size.



(d) Varying number of sessions.

Fig. 4. ADMs required by Algorithms 1, 2, A, B and lower bound L.

that group. This clustering can be done in $O(M)$ steps. Let there be K such clusters $C_0, \ldots, C_{K-1}$. Let the number of traffic requests in cluster C_i be g_i. All the traffic requests in cluster C_i are routed on wavelength λ_i and the set of network nodes which should be equipped with ADM on wavelength λ_i is given by (36). As argued above in the complexity analysis of Algorithm A, the number of steps required for evaluating S_{λ_i} is Ng_i. So the total number of steps required for determining the placement of ADMs at each network node on all the wavelengths is

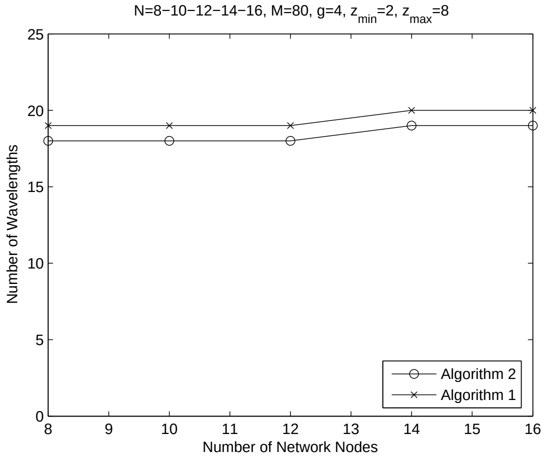
$$\sum_{i=0}^{K-1} Ng_i = N \sum_{i=0}^{K-1} g_i = NM \tag{38}$$

Therefore the overall complexity of Algorithm B is $O(NM + M^2)$.

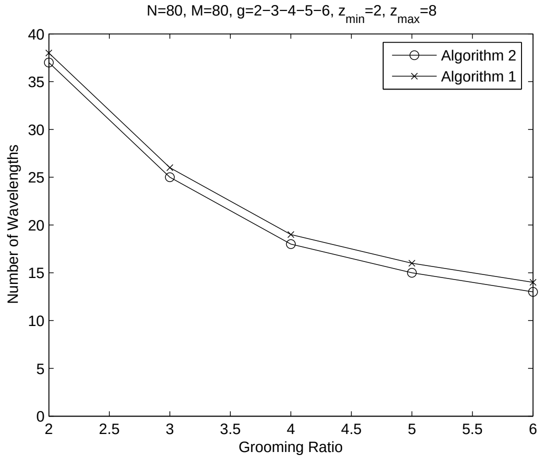
C. Heuristic

Consider the graph based traffic grooming heuristic presented as Algorithm 1. As described in Section IV-B, we start off by assigning a different wavelength to every traffic request. In each iteration on the heuristic, we update the wavelength assignment by first determining the best (as described in Section IV-B) reducible wavelength pair and then assigning all the traffic requests on the two wavelengths to a single wavelength. We continue to update the wavelength assignment iteratively till there are no more reducible wavelength pairs left. The wavelength assignment after completing n steps of the heuristic is maintained as graph $H(n) = (\Lambda(n), L(n))$ where the vertex set represents the set of wavelengths. For each wavelength $\lambda_i \in \Lambda(n)$ we maintain S_{λ_i}, the set of network nodes which act as source or destination nodes for any multicast session being groomed on wavelength λ_i. Also for every wavelength pair $\lambda_i, \lambda_j \in \Lambda(n)$, we maintain $|S_{\lambda_i} \bigcup S_{\lambda_j}|$, $|S_{\lambda_i} \bigcap S_{\lambda_j}|$ and whether the wavelength pair is reducible or not.

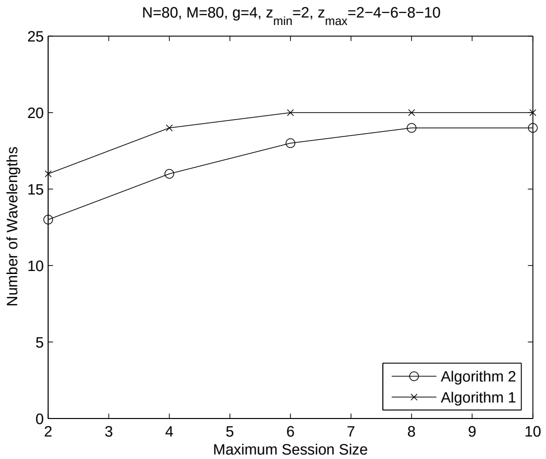
First we study the complexity of the $n+1$-st iteration in Algorithm 1. Since in each iteration we reduce the number of wavelengths by 1, $|\Lambda(n)| = |\Lambda(0)| - n = M - n$. So the



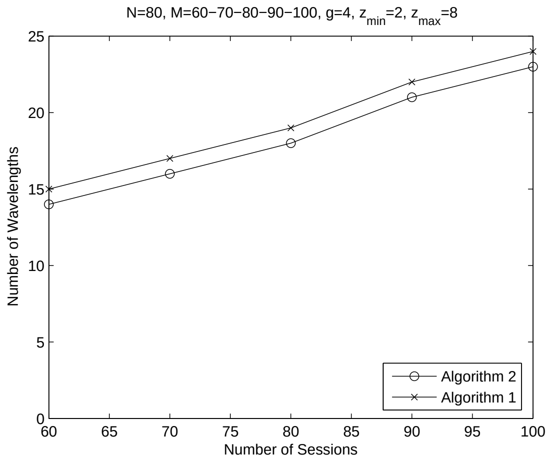
(a) Varying network size.



(b) Varying grooming ratio.



(c) Varying session size.



(d) Varying number of sessions.

Fig. 5. Wavelengths per fiber required by Algorithms 1, 2.

number of wavelength pairs to consider is $\frac{(M-n)(M-n-1)}{2}$. The number of steps required to determine the best reducible wavelength pair is linear in the number of wavelength pairs. After determining the best reducible wavelength pair $\lambda_\alpha, \lambda_\beta \in \Lambda(n)$ we update graph $H(n)$ to $H(n+1)$ with vertex set $\Lambda(n+1) = (\lambda(n) \bigcup \{\lambda_{\alpha\beta}\}) \setminus \{\lambda_\alpha, \lambda_\beta\}$ where all the traffic requests previously groomed on wavelengths λ_α and λ_β are now assigned on the new wavelength $\lambda_{\alpha\beta}$. As explained in Section IV-B, we need to compute $S_{\lambda_{\alpha\beta}} = S_{\lambda_\alpha} \bigcup S_{\lambda_\beta}$ and for every other wavelength $\lambda_i \in \Lambda(n+1)$ we need to evaluate $|S_{\lambda_{\alpha\beta}} \bigcup S_{\lambda_i}|$, $|S_{\lambda_{\alpha\beta}} \bigcap S_{\lambda_i}|$ and whether the wavelength pair $\lambda_{\alpha\beta}, \lambda_i$ is reducible or not. Since $S_\lambda \subseteq \{0, \ldots, N-1\}$ for any wavelength λ, evaluating $S_{\lambda_{\alpha\beta}}$, $|S_{\lambda_{\alpha\beta}} \bigcup S_{\lambda_i}|$ and $|S_{\lambda_{\alpha\beta}} \bigcap S_{\lambda_i}|$ require $O(N)$ steps. For any wavelength λ_k, let C_k denote the set of traffic requests that are assigned wavelength λ_k. To determine whether wavelength pair $\lambda_{\alpha\beta}, \lambda_i$ is reducible or not, we check if we can color graph $G[C_i \bigcup C_{\alpha\beta}]$, the contention graph of all the traffic requests that are assigned wavelengths λ_i or $\lambda_{\alpha\beta}$, using at most g colors or not. We employ Tucker's algorithm [23] for coloring the circular arc graph which requires $O(|C_i \bigcup C_{\alpha\beta}|^2)$ time. Since $|C_i \bigcup C_{\alpha\beta}| \leq M$, checking if wavelength pair $\lambda_i, \lambda_{\alpha\beta}$ is reducible or not takes $O(M^2)$ time. Therefore the number of steps required in $n+1$-st iteration of the heuristic is $(M-n-2)O(N+M^2)$. As already explained after each iteration, the size of the vertex set of the graph decreases by 1, so there can be a maximum of $M-1$ iterations. Hence the iterations in the heuristic require $O(M^2(N+M^2))$ steps.

Now we count the number of steps required to initialize the graph $H(0)$ in the first step of the heuristic. Note that checking whether a wavelength pair $\lambda_i, \lambda_j \in \Lambda(0)$ is reducible or not requires $O(1)$ steps. This is because every wavelength corresponds to just a single traffic request. Again, determining $|S_{\lambda_i} \bigcup S_{\lambda_j}|$ and $|S_{\lambda_i} \bigcap S_{\lambda_j}|$ require $O(N)$ steps. Since there are $\frac{M(M-1)}{2}$ wavelengths pairs, the construction of graph $H(0)$ requires $O(NM^2)$ steps.

Therefore the overall complexity of Algorithm 1 is $O(M^2(N+M^2))$.

VII. Numerical Results

Since presently there is no other heuristic for grooming multicast traffic in unidirectional rings with which we can compare our heuristic, we extend the unicast traffic grooming algorithm presented in [1] to the multicast case. We do this by simply starting with multicast sessions in place of unicast sessions in the circle construction phase. More specifically, we try to put as many multicast sessions on circles without introducing gaps. In [1], the authors do this for unicast sessions by assuming each unicast session to be a *connection* and then combining two connections with common end points to form complete circles. After constructing the maximum possible circles in this way, they then apply *Algorithm IV:Construct Circles - Non-Uniform Traffic* to construct the rest of the circles. Each circle here corresponds to a subwavelength channel. After all the connections have been assigned to some circle, the circles are groomed into wavelengths. In our extension of this algorithm, we consider multicast sessions to be the starting connections and construct the circles in exactly the same way. After we have the circles, the circle grooming heuristic is exactly the same as in [1]. We refer to our heuristic as *Algorithm* 1 and this extended heuristic as *Algorithm* 2.

We evaluate the performance of both Algorithms 1 and 2 in terms of the number of ADMs required. For a more complete picture, we also compare the performance of both the heuristics to our lower bound as given in equation (4). Since the number of wavelengths required also contributes to the network cost (albeit, not as much as the ADMs), we also compare the wavelengths required by the two heuristics. For the sake of completeness we also compare the two simple multicast traffic grooming schemes presented as Algorithm A and Algorithm B in Section V-B.

We identify the problem of grooming multicast traffic on unidirectional rings by the five parameters: N, M, g, z_{min} and z_{max}. Here the parameters N, M and g denote the number of nodes in the ring, the number of multicast sessions and the grooming ratio respectively. Parameters z_{min} and z_{max} denote the minimum and the maximum possible size of the multicast sessions. For the purpose of simulation, while generating a multicast session, each node is given equal probability of being selected as the source. The size of each multicast session is selected uniformly from z_{min} to z_{max}. After the source node and the size z of the multicast session are fixed, destination nodes are selected such that every subset of size $z-1$ of the remaining $N-1$ nodes (since one node has already been selected as the source) has equal probability of being the destination set.

For simulation, we consider a nominal ring network having 10 nodes, 80 multicast sessions, with each session size selected uniformly between 2 to 8 and having grooming ratio 4. We study the performance of both the heuristics by varying one parameter of the problem at a time in this nominal network. More specifically, we vary the grooming ratio from 2 to 6, the network size (number of nodes in the network) from 8 to 16, the number of multicast sessions from 60 to 100 and the maximum size of multicast sessions from 2 to 10. Figure 4 presents the simulation results comparing the number of ADMs required by the various grooming schemes as well as the number of ADMs specified by our lower bound L. The simulation results comparing the wavelengths per fiber required by Algorithm 1 and 2 are presented in Figure 5. Each point in the plots is generated by taking an average of 20 randomly selected grooming problem instances with the required parameters.

We can see from the plots that, as measured by the number of ADMs required, our Algorithm 1 always outperforms Algorithm 2. This is true even for unicast traffic (the case for which Algorithm 2 was designed in [1]). We also note that our Algorithm 1 usually requires more wavelengths than Algorithm 2. But the increase in the number of wavelengths is never more than 2, and is overshadowed by the savings in the number of (more expensive) ADMs.

Form the plots we also observe that of the three grooming schemes presented in this work, our graph based grooming heuristic (Algorithm 1) always outperforms the simple grooming from Section V-B (Algorithms A and B). And among the two simple schemes, Algorithm B always outperforms Algorithm A. We can justify this trend in the light of the complexity analysis of the three schemes presented in Section VI. Assuming that the number of traffic requests to be groomed is much larger than the number of network nodes (which is usually the case and is true for our simulations as well), we observe that based on their time complexities, Algorithm A is the simplest, Algorithm 1 is the hardest and Algorithm B lies somewhere in-between the two. Since we get what we pay for, the relative performances of the three schemes is as expected. Although not presented in the plots, the number of wavelengths required by Algorithms A and B are also very similar to that required by Algorithms 1 and 2.

Also from the plots, we can see that the lower bound L given in (4) tracks the performance curves of the heuristics as we vary the grooming ratio, the number of sessions or the size of sessions. This suggests that the bound tracks the changes in these parameters quite well. But we observe that this is not so in the case of the size of network. This is consistent with our discussion in Section V, and it is easy to verify that the value of the lower bound (averaged over 20 runs) closely matches our estimate given in (8).

VIII. Conclusion

In this paper we have studied the problem of grooming non-uniform multicast traffic on a unidirectional SONET/WDM ring. We consider two different costs, (*i*) number of wavelengths and (*ii*) number of ADMs. We observe that minimizing the number of wavelengths can be modeled as a standard arc-graph coloring problem. We then give a graph based heuristic for minimizing the number of ADMs. Based on extensive simulations we observe that our heuristic performs better than the multicast extension of the best known unicast traffic grooming heuristic for rings given in [1]. We also develop a lower bound for the problem and look at some interesting relations between the lower bound and a couple of upper bounds.

References

[1] X. Zhang and C. Qiao, “An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings,” *IEEE/ACM Trans. Networking*, Oct. 2000.

[2] A. L. Chiu and E. H. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *IEEE/OSA J. Lightwave Technol.*, Jan. 2000.

[3] X. Zhang and C. Qiao, "On scheduling all-to-all personalized connections and cost-effective designs in WDM rings," *IEEE/ACM Trans. Networking*, June 1999.

[4] ——, "Scheduling in unidirectional WDM rings and its extensions," in *Proc. SPIE All Optical Communication Systems: Architecture, Control and Network Issues III*, 1997.

[5] A. R. B. Billah, B. Wang, and A. A. S. Awwal, "Effective traffic grooming in WDM rings," in *Proc. IEEE Globecom 2002*.

[6] O. Gerstel and R. Ramaswami, "Cost-effective traffic grooming in WDM rings," *IEEE/ACM Trans. Networking*, Oct. 2000.

[7] K. Zhu and B. Mukherjee, "Traffic grooming in an optical WDM mesh network," *IEEE J. Sel. Areas Commun.*, Jan. 2002.

[8] H. Zhu, H. Zang, K. Zhu, and B. Mukherjee, "A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks," *IEEE/ACM Trans. Networking*, April 2003.

[9] F. Farahmandz, X. Huang, and J. P. Jue, "Efficient online traffic grooming algorithms in WDM mesh networks with drop-and-continue node architecture," in *Proc. IEEE Broadnets*, Oct. 2004.

[10] X. Huang, F. Farahmandz, and J. P. Jue, "An algorithm for traffic grooming in WDM mesh networks with dynamically changing light-trees," in *Proc. IEEE Globecom 2004*.

[11] K. Zhu and B. Mukherjee, "A review of traffic grooming in WDM optical networks: Architectures and challenges," *Optical Networks Mag.*, March/April 2003.

[12] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Network*, Nov./Dec. 2002.

[13] G. V. Chowdhary and C. S. R. Murthy, "Grooming of multicast sessions in WDM mesh networks," in *Proc. IEEE Broadnets*, Oct. 2004.

[14] X. Huang, F. Farahmandz, and J. P. Jue, "Multicast traffic grooming in wavelength-routed WDM mesh networks using dynamically changing light-trees," *IEEE/OSA J. Lightwave Technol.*, Oct. 2005.

[15] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, and M. A. Ali, "On multicast traffic grooming in WDM networks," in *Proc. IEEE International Symposium on Computers and Communications*, June 2004.

[16] A. Khalil, C. Assi, A. Hadjiantonis, G. Ellinas, N. Abdellatif, and M. A. Ali, "Multicast traffic grooming in WDM networks," in *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, May 2004.

[17] A. R. B. Billah, B. Wang, and A. A. S. Awwal, "Multicast traffic grooming in WDM optical mesh networks," in *Proc. IEEE Globecom*, Dec. 2003.

[18] G. V. Chowdhary and C. S. R. Murthy, "Dynamic multicast traffic engineering in WDM groomed mesh networks," in *Proc. IEEE Broadnets*, Oct. 2004.

[19] A. Khalil, A. Hadjiantonis, G. Ellinas, and M. Ali, "Sequential and hybrid grooming approaches for multicast traffic in WDM networks," in *Proc. IEEE Globecom 2004*.

[20] H. V. Madhyastha, N. Srinivas, G. V. Chowdhary, and C. S. R. Murthy, "Grooming of multicast sessions in WDM ring networks," in *Proc. SPIE Opticomm*, Oct. 2003.

[21] J. Wang, W. Cho, V. R. Vemuri, and B. Mukherjee, "Improved approaches for cost-effective traffic grooming in WDM rings: ILP formulations and single-hop and multihop connections," *IEEE/OSA J. Lightwave Technol.*, Nov. 2001.

[22] M. Ali and J. S. Deogun, "Cost-effective implementation of multicasting in wavelength-routed networks," *IEEE/OSA J. Lightwave Technol.*, Dec. 2000.

[23] A. Tucker, "Coloring a family of circular arcs," *SIAM J. Applied Mathematics*, 1975.

[24] M. Garey, D. Johnson, G. Miller, and C. Papadimitriou, "The complexity of coloring circular arcs and chords," *SIAM J. Algebraic and Discrete Methods*, 1980.

[25] V. Kumar, "An approximation algorithm for circular arc coloring," *Algorithmica*, 2001.

[26] I. A. Karapetian, "On the coloring circular arc graphs," *Docladi (Reports) of the Academy of Science of the Armenian Soviet Socialist Republic*, 1980.

[27] T. A. McKee and F. R. McMorris, *Topics in Intersection Graph Theory*. Philadelphia, PA: SIAM, 1999.

[28] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*. New York: Academic Press, 1980.

[29] S. Olariu, "An optimal greedy heuristic to color interval graphs," *Information Processing Lett.*, 1991.

Anuj Rawat received the B.Tech. in Electrical Engineering from the Indian Institute of Technology, Kanpur, India, in 2002. From 2002 to 2003, he was a member of technical staff at Atrenta, in India. Since fall 2003 he has been working towards his Ph.D. degree in Electrical and Computer Engineering at the University of Maryland at College Park, MD, USA. His research interests include wireless sensor networks, design of algorithms for optical networks, traffic engineering and routing.

Richard J. La (S'98 - M'01) received the B.S.E.E. from the University of Maryland at College Park in 1994, and the M.S. and PhD. degrees in Electrical Engineering from the University of California at Berkeley in 1997 and 2000, respectively. From 2000 to 2001 he was a senior engineer in the Mathematics of Communication Networks group at Motorola. Since August 2001 he has been on the faculty of the ECE department at the University of Maryland at College Park.

Steven I. Marcus (Fellow, IEEE) received the B.A. degree in electrical engineering and mathematics from Rice University in 1971 and the S.M. and Ph.D. degrees in electrical engineering from the Massachusetts Institute of Technology in 1972 and 1975, respectively. From 1975 to 1991, he was with the Department of Electrical and Computer Engineering at the University of Texas at Austin, where he was the L.B. (Preach) Meaders Professor in Engineering. He was Associate Chairman of the Department during the period 1984-89. In 1991, he joined the University of Maryland, College Park, where he was Director of the Institute for Systems Research until 1996. He was Chair of the Electrical and Computer Engineering Department at the University of Maryland during the period 2000-2005. He is currently Professor in the Electrical and Computer Engineering Department and the Institute for Systems Research. His research is focused on stochastic control and estimation, with applications in manufacturing and communication networks. His email address is: marcus@umd.edu.

Mark A. Shayman received the Ph.D. degree in Applied Mathematics from Harvard University, Cambridge, MA, in 1981. He served on the faculty of Washington University, St. Louis, MO, and the University of Maryland, College Park, where he is currently Professor of Electrical and Computer Engineering. His research interests are in communication networks, including optical, wireless and sensor networks. He received the Donald P. Eckman award from the American Automatic Control Council and the Presidential Young Investigator Award from the National Science Foundation. He has served as Associate Editor of the *IEEE Transactions on Automatic Control*.