Approximation Algorithms for Many-to-Many Traffic Grooming in Optical WDM Networks

Mohammad A. Saleh Ahmed E. Kamal

Abstract—**A large number of network applications today allow several users to interact together using the *many-to-many* service mode. In many-to-many communication, also referred to as group communication, a session consists of a group of users (we refer to them as *members*), where each member transmits its traffic to all other members in the same group. In this paper, we address the problem of grooming sub-wavelength many-to-many traffic (e.g., OC-3) into high-bandwidth wavelength channels (e.g., OC-192) in optical WDM mesh networks.**

The cost of an optical WDM network is dominated by the cost of higher layer electronic ports (i.e., *transceivers*). A transceiver is needed for each initiation and termination of a lightpath. Therefore, our objective is to minimize the total number of lightpaths established. Unfortunately, the grooming problem even with unicast traffic has been shown to be NP-hard. In this work, we introduce two novel approximation algorithms for the many-to-many traffic grooming problem. We also consider the routing and wavelength assignment problem with the objective of minimizing the number of wavelengths used. Through extensive experiments, we show that the proposed algorithms use a number of lightpaths that is very close to that of a derived lower bound. Also, we compare the two algorithms on other important objectives such as the number of logical hops traversed by a traffic stream, total amount of electronic switching at a node, and Min-Max objectives.

I. INTRODUCTION

In optical wavelength routing networks, using wavelength division multiplexing (WDM), it is feasible to have hundreds of wavelengths per fiber each operating at 10 to 40 Gbps. Bandwidth requirements of user sessions, however, are usually of sub-wavelength granularities. For example, an MPEG compressed HDTV channel requires less than 20 Mbps of bandwidth. In order to reduce this huge bandwidth gap, *traffic grooming* was introduced to allow a number of sessions with sub-wavelength granularities to share the bandwidth of a wavelength channel.

Early network applications such as TELNET and FTP are characterized as unicast or "one-to-one". A large portion of network applications today, however, are of the multipoint type. For example, video distribution and file distribution are examples of multicast or "one-to-many" applications, while resource discovery and data collection are examples of many-to-one or "inverse multicasting" applications. Recently, another set of multipoint network applications has emerged such as multimedia conferencing, e-science applications, distance learning, distributed simulations, and collaborative processing [1]. In these applications, each of the participating entities both contributes and receives information to and from the other entities in the same communication session, and therefore are characterized as "many-to-many". In many-to-many communication, also referred to as group communication [2], a session consists of a group of users (we refer to them as *members*), where each member transmits its traffic to all other members in the same group (see Fig. 1).

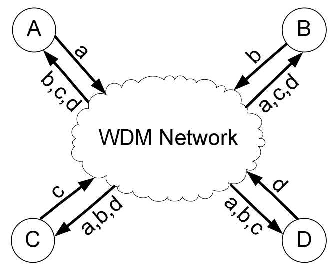


Fig. 1. A many-to-many session with members $\{A, B, C, D\}$ each with traffic denoted as a, b, c and d, respectively.

Most of the early work on traffic grooming has focused on unicast traffic on ring and mesh topologies. Since a large portion of network applications today are of the multipoint type, many of the recent studies on traffic grooming have focused on multicast and many-to-one traffic types. In this work, we consider the many-to-many traffic grooming problem in optical WDM mesh networks, which is an important new research problem. In this problem, a collection of many-to-many session requests, each with an arbitrary sub-wavelength traffic demand, are given and the objective is to find a set of lightpaths and to find the corresponding routing and grooming of each of the traffic demands onto these lightpaths. For a complete design of the optical WDM network, the traffic grooming problem is followed by the routing and wavelength assignment (RWA) problem which finds routes and assigns wavelengths to each of the lightpaths on the optical WDM network. The objective of the RWA problem is to minimize the total number of wavelengths used. The RWA has been extensively studied in the literature and it has been shown to be NP-complete. Therefore, in this work, we focus on the many-to-many traffic grooming problem at the virtual topology level. However, for completeness and for comparison purposes, we use one of the best existing heuristics for the RWA problem (the LFAP heuristic [3]).

The cost of an optical WDM network is dominated by the cost of higher layer electronic ports such as IP router ports, MPLS Label Switching Router (LSR) ports and SONET ADM ports (we refer to these ports as *transceivers*). A transceiver is needed for each initiation and termination of a lightpath

Part of this paper has been presented at the IEEE Infocom 2010 conference. This research was supported in part by grant CNS-0626741 from the National Science Foundation.

The authors are with the Department of Electrical and Computer Engineering, Iowa State University, Ames, Iowa 50011, USA (e-mail: msaleh@iastate.edu; kamal@iastate.edu).

(i.e, each lightpath requires two transceivers). Therefore, our objective in the many-to-many traffic grooming problem is to minimize the total number of lightpaths established.

Many-to-many group communication is closely related to multicast communication. In fact,a many-to-many session with N embers can be viewed as a set of N multicast sessions each sourced at one of the N members and destined to the remaining $N-1$ members. However, many-to-many communication has certain properties that makes it worth studying on its own merit and not as a multicast traffic problem[1]. This will become more clear after we introduce the concept of a *lightpath cycle*, which was first introduced in our earlier work [25], in Section II.

A. Problem Statement

An input to the many-to-many traffic grooming problem is an optical WDM network with an arbitrary physical topology represented by an undirected graph $G(V, E)$ with a set of nodes V numbered 0,1, ... ,$N-1$ ($N = |V|$) and a set of links E. Each undirected link corresponds to two unidirectional fibers in opposite directions. The number of wavelengths per fiber is the same among all fibers and is denoted by W_{max}, while the bandwidth capacity of a wavelength channel (i.e., the grooming factor) is g units of traffic (the unit of traffic may be, e.g., an OC-3 circuit). Another input to the many-to-many traffic grooming problem is a collection of K many-to-many session requests, where each session s_k ($1 \leq k \leq K$) has a set of members $m_{s_k} \subseteq V$ with cardinality $N_{s_k} = |m_{s_k}|$. We assume uniformity of traffic within the same session, that is, each member in m_{s_k} has the same traffic demand t_{s_k}, where $1 \leq t_{s_k} \leq g$.[2] Let N_{min} and t_{min} denote the minimum session size and the minimum traffic demand among all the K sessions, respectively and let k_i denote the number of sessions of which node i is a member.

The output of the many-to-many traffic grooming problem is a set of lightpaths and the corresponding routing and grooming of each of the many-to-many traffic demands onto these lightpaths. More precisely, the output is the number of lightpaths P_{ij} to be established between each ordered pair of nodes (i, j), where $i, j \in V$, and the sequence of lightpaths that each traffic stream follows. Note that for each member in a many-to-many session s_k, there is a traffic stream of t_{s_k} traffic units that needs to be delivered to the other $N_{s_k} - 1$ members in the same session.

The objective of the many-to-many traffic grooming problem is to minimize the total number of lightpaths established P ($P = \sum_{i,j} P_{ij}$). It was shown in [4] that the unicast traffic grooming problem without the RWA problem is NP-hard. In our earlier work [25], Theorem 1, we have proved that the many-to-many traffic grooming problem is NP-hard by showing that the bidirectional unicast traffic grooming problem is a special case of many-to-many traffic grooming (a bidirecitional unicast session is simply a many-to-many session with two members).

B. Related Work

Traffic grooming has been extensively studied for unicast traffic [5]-[14]. Some of the studies were restricted to ring topologies [13], [11], [8], [9], while others were for general mesh topologies [5], [6], [12], [7], [14], [10]. In [13], the authors addressed the traffic grooming problem on a number of WDM ring architectures with the objective of minimizing the overall network cost. In [8], the authors proposed optimal and near-optimal algorithms for traffic grooming in SONET WDM rings with the objective of minimizing the number of wavelengths and SONET ADMs. In [10], the authors proposed an auxiliary graph model for traffic grooming in heterogeneous WDM mesh networks and developed an integrated traffic grooming algorithm that jointly solves the traffic grooming subproblems. In [5], the authors showed that the traffic grooming problem is APX-hard, which means that the optimum cannot be approximated arbitrarily closely. They also proposed approximation algorithms for minimizing the total equipment cost and for minimizing the lightpath count. In [6], the authors provided a hierarchical framework for traffic grooming in a WDM mesh network. For a survey of advances in unicast traffic grooming, the reader is referred to [14].

Traffic grooming has also been considered for multicast traffic [15]-[21]. Similar to unicast traffic, some of the studies were restricted to ring topologies [15], [16], while others were for general mesh topologies [18], [19], [17], [21]. In [15], the authors addressed the multicast traffic grooming problem in metropolitan WDM ring networks with the objective of minimizing the electronic copying. In [16], the authors introduced a graph based heuristic for the multicast traffic grooming problem in unidirectional SONET/WDM rings and compared it to the multicast extension of the best known unicast traffic grooming heuristic in [8]. In [18] and [20], the authors addressed the multicast and the many-to-one traffic grooming problems, respectively in WDM mesh networks. They provided MILP formulations and also developed heuristic solutions. For a survey of advances in multicast and many-to-one traffic grooming, the reader is referred to [21], [22] Chapter 14.

Many-to-many traffic grooming is a new research problem that has been only considered in [23]-[25]. In [23], the authors addressed the many-to-many traffic grooming problem in WDM ring networks with the objective of reducing the overall network cost. In our previous works [24]-[25], MILP formulations and heuristic solutions were introduced for the many-to-many traffic grooming problem in optical WDM mesh networks. Although the MILPs guaranteed an optimal solution, their complexity was too high. Also, the heuristics introduced did not guarantee any upper bound on the number of lightpaths required. This work, which extends our work [26], is the first to propose approximation algorithms for the many-to-many traffic grooming problem in optical WDM networks. In [26], only symmetric many-to-many traffic was considered where members within the same session have the

[1]This is analogous to viewing a multicast session as a group of unicast sessions from the same source to all destinations. The consideration of group communication properties results in a more efficient provisioning than using multiple unicast sessions.

[2]In Section III, we relax this assumption and consider many-to-many sessions where each member in the session may have different traffic demand.

TABLE I
LIST OF SYMBOLS USED IN THE PAPER

Symbol	Definition
$G(V, E)$	undirected graph with a set of nodes V and a set of links E, which represents the physical topology of the WDM network.
N	number of nodes in the network ($N = \|V\|$).
K	number of many-to-many sessions.
g	grooming factor.
W_{max}	total number of wavelengths per fiber link.
W	total number of wavelengths used ($W \leq W_{max}$).
s_k	many-to-many session number k ($1 \leq k \leq K$).
m_{s_k}	set of members in session s_k.
N_{s_k}	number of members in session s_k ($N_{s_k} = \|m_{s_k}\|$).
t_{s_k}	traffic demand of members in session s_k.
k_i	number of sessions where node i is a member.
N_{min}	minimum session size among all the K sessions.
t_{min}	minimum traffic demand among all the K sessions.
S_i	set of sessions where node i is a member ($k_i = \|S_i\|$).
S_{ij}	set of sessions where both nodes i and j are members.
S_{iFj}	set of sessions where member i follows member j immediately in the session's LCs.
P_{ij}	total number of lightpaths from node i to node j.
P	total number of lightpaths in the network ($P = \sum_{i,j} P_{ij}$).
rem_{ij}	remaining unused capacity on lightpaths from i to j if we place j after i in the LCs for all sessions in the set S_{ij}.
$l^{s_k,ij}_{Alg}$	the number of logical hops traversed by the traffic stream originating from member $i \in m_{s_k}$ and destined to member $j \in m_{s_k}$ according to the Alg Algorithm.
$l^{s_k,i}_{Alg}$	the average number of logical hops traversed by a traffic stream originating from member $i \in m_{s_k}$ according to the Alg Algorithm.
$l^{s_k}_{Alg}$	the average number of logical hops traversed by a traffic stream in session s_k according to the Alg Algorithm.
l_{Alg}	the average number of logical hops traversed by a traffic stream in any of the K sessions according to the Alg Algorithm.
e^i_{Alg}	the total amount of electronic switching at node i according to the Alg Algorithm.
e_{Alg}	the total amount of electronic switching in the whole network (at all nodes) according to the Alg Algorithm.
P^{Alg}_{max}	maximum number of lightpaths incoming or outgoing at a node according to the Alg Algorithm.
e^{Alg}_{max}	maximum amount of electronic switching at a node according to the Alg Algorithm.

same traffic demand. In this work, we extend the work in [26] by considering asymmetric many-to-many traffic where members within the same session may have different traffic demands. We also extend the work in [26] by studying and comparing the performance of the proposed approximation algorithms on other important objectives such as the number of logical hops traversed by a traffic stream, total amount of electronic switching at a node, and Min-Max objectives.

C. Contributions and Paper Organization

In this paper, we study the many-to-many traffic grooming problem in optical WDM mesh networks, which is an important new research problem. The paper organization and contributions are as follows:

- In Section II, we introduce two novel approximation algorithms for the many-to-many traffic grooming problem. The first algorithm is based on lightpath cycles (which we will introduce in Section II.A) and has an approximation ratio of $\min\{g, 1 + \frac{g}{(N_{min}-1)t_{min}}, N - N_{min} + 1\}$. The second algorithm is based on a hub node that collects and distributes traffic and has a 2-approximation ratio. We also consider the RWA problem with the objective of minimizing the total number of wavelengths used W.
- In Section III, we consider the asymmetric many-to-many traffic grooming problem where members within the same many-to-many session may have different traffic demands. In this problem, we extend the algorithms proposed for the symmetric traffic case and show that the approximation ratios still hold.
- In Section IV, we study and compare the performance of the two algorithms on other important objectives such as the number of logical hops traversed by a traffic stream, total amount of electronic switching at a node, and Min-Max objectives.
- In Section V, we conduct extensive experiments to show that the algorithms proposed use a number of lightpaths that is extremely close to that of a derived lower bound. We also compare the algorithms on the several objectives mentioned in the paper.
- In Section VI, we conclude the paper. For a quick reference, Table I lists all the symbols used in the paper.

II. APPROXIMATION ALGORITHMS

As we stated before, the general many-to-many traffic grooming problem is NP-hard. In this section, we introduce two novel approximation algorithms for the many-to-many traffic grooming problem in optical WDM mesh networks. This section is organized as follows. First, we present *lightpath cycles* which was originally introduced in our earlier work [25]. Then, we derive a lower and an upper bounds on the total number of lightpaths required by any many-to-many traffic grooming algorithm. Then, we introduce the lightpath cycles and the hub approximation algorithms and we analyze their complexities. Finally, we address the routing and wavelength assignment problem.

A. Lightpath Cycles

In our earlier work [25], we introduced *lightpath cycles* as the optimal solution for a number of special cases where the many-to-many traffic grooming problem is tractable. In this subsection, we briefly present lightpath cycles to allow the introduction of our new contributions to the many-to-many traffic grooming problem. For detailed information on lightpath cycles and their optimality results, the reader is referred to [25]. Next, we have the following definition.

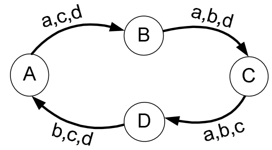


Fig. 2. LC for a many-to-many session s_k with a set of members $m_{s_k} = \{A, B, C, D\}$ each with one traffic unit denoted as a, b, c and d, respectively ($g = 3$).

Definition 1. *A lightpath cycle (LC) for a many-to-many session s_k is a simple cycle of N_{s_k} lightpaths that visits each member in m_{s_k} exactly once.*

An example of a LC for a many-to-many session s_k with a set of members $m_{s_k} = \{A, B, C, D\}$ is shown in Fig. 2. Note that LCs only describe a virtual topology and their mapping to the physical topology is part of the RWA problem.

As we described earlier in Section I, many-to-many communication is closely related to multicast communication. Existing multicast traffic grooming algorithms focus on provisioning multicast sessions using a tree structure and try to optimize network resources by efficiently grooming traffic from different multicast trees (see Section I.B). In many-to-many communication, however, we have proved in [25] that the optimal provisioning of a many-to-many session is a cycle structure (lightpath cycle) not a tree structure. In addition, in Section II and III we introduce two novel apporximation algorithms that are based on lightpath cycles and produce solutions that are very close to optimal. These reasons justify and motivate the study of the many-to-many traffic grooming problem in its own merit and not as a multicast traffic grooming problem. To further emphasize this, we introduce an example, shown in Fig. 3, of a many-to-many session s_1 with a set of members $m_{s_k} = \{A, B, C\}$ each with one traffic unit denoted as a,b, and c, respectively ($g = 2$). Fig. 3.(a) shows the session provisioning using the well-know shorest path tree (SPT) multicast traffic grooming algorithm. A SPT is originated at each of the members A, B, and C and terminated at the other two members. Note that the SPT originating at member C is chosen to groom member C traffic on existing lightpaths $C \rightarrow A$ and $A \rightarrow B$ to save the number of lightpaths established. Fig. 3.(b) shows the session provisioning using a lightpath cycle. We can see from this example that viewing the many-to-many session as a set of multicast sessions and provisioning each multicast session using the SPT results in using more lightpaths and wavelengths than provisioning the many-to-many session using a lightpath cycle.

B. A lower bound

We derive a lower bound on the number of lightpaths required by a many-to-many traffic grooming algorithm by considering each node in the network separately. The minimum number of lightpaths incoming to a node i can be found by counting the total traffic that this node should receive from all sessions s_k where $i \in m_{s_k}$. Let S_i denotes the set of sessions where node i is a member (note that $|S_i| = k_i$) . The total traffic that node i should receive is $\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}$. Therefore, at least $\left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k}-1)t_{s_k}}{g} \right\rceil$ lightpaths should be incoming to i in order to receive this traffic. Summing over all the nodes in the network, we obtain a lower bound L on the total number of lightpaths required:

$$L = \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil \quad (1)$$

We note that this is just a lower bound on the number of lightpaths and it does not necessarily correspond to a feasible solution to the many-to-many traffic grooming problem. Note, however, that for a single many-to-many session lightpath cycles are optimal since the number of lightpaths used meets the lower bound L (See Fig. 2). Next, we obtain an upper bound on the number of lightpaths required by any many-to-many traffic grooming algorithm.

C. An upper bound

We consider the worst case scenario where no traffic grooming is performed between any two traffic streams even within the same session. In this case, each node i will have a direct lightpath incoming from each of the other $N_{s_k} - 1$ members in the same session s_k for all sessions $s_k \in S_i$. Therefore, the total number of lightpaths P required according to this worst case scenario is given by:

$$P = \sum_{i=0}^{N-1} \sum_{s_k \in S_i} N_{s_k} - 1 \leq \sum_{i=0}^{N-1} \sum_{s_k \in S_i} (N_{s_k} - 1) \frac{t_{s_k}}{t_{min}}$$

$$\leq \frac{1}{t_{min}} \sum_{i=0}^{N-1} g \left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil = \frac{g}{t_{min}} L \quad (2)$$

The first inequality holds due to the fact that $\frac{t_{s_k}}{t_{min}} \geq 1$.Since this is the worst case scenario, then it serves as an upper bound for any many-to-many traffic grooming algorithm. Hence, we have the following result:

Theorem 1. *Any many-to-many traffic grooming algorithm with any grooming policy is a $\frac{g}{t_{min}}$ approximation algorithm.*

Next, we propose two novel approximation algorithms for the many-to-many traffic grooming problem in optical WDM mesh networks.

D. Lightpath Cycles Algorithm

In this algorithm, we assume that many-to-many sessions are provisioned through lightpath cycles (LCs). Although the optimality of LCs was only for certain special cases, we will show that this assumption generally gives near-optimal solutions. First, let us assume that each session s_k is provisioned through $\left\lceil \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil$ identically ordered LCs for s_k and ignore inter-session grooming (LCs in this case only perform intra-session grooming between members within the same session, see Fig. 2.(a)). In this case, node i will have $\left\lceil \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil$ lightpaths incoming from each session $s_k \in S_i$.

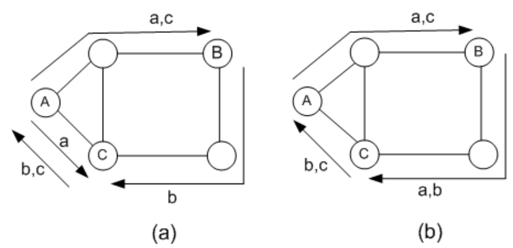


Fig. 3. A many-to-many session with members $\{A, B, C\}$ each with one traffic unit denoted as a, b and c, respectively ($g = 2$). (a): Session provisioning using a set of multicast SPTs each sourced at one of the members a,b and c. (b): Session provisioning using a lightpath cycle

Hence, the total number of lightpaths P required according to this algorithm is given by:

$$P = \sum_{i=0}^{N-1} \sum_{s_k \in S_i} \left\lceil \frac{(N_{s_k} - 1)t_{s_k}}{g} \right\rceil$$

$$\leq \sum_{i=0}^{N-1} \left(\left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil + k_i \right)$$

$$= L + \sum_{i=0}^{N-1} k_i \quad (3)$$

The inequality holds due to the fact that $\sum_{m=0}^{M-1} \lceil x_m \rceil \leq \left\lceil \sum_{m=0}^{M-1} x_m \right\rceil + M$ for any positive integer M and positive real values $x_1, x_2, ..., x_{M-1}$.

Now, let us consider the lower bound L again:

$$L = \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil$$

$$\geq \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} (N_{min} - 1)t_{min}}{g} \right\rceil$$

$$\geq \sum_{i=0}^{N-1} \frac{\sum_{s_k \in S_i} (N_{min} - 1)t_{min}}{g}$$

$$= \frac{(N_{min} - 1)t_{min}}{g} \sum_{i=0}^{N-1} k_i \quad (4)$$

Substituting (4) in (3), we have:

$$P \leq L + \sum_{i=0}^{N-1} k_i \leq L + \frac{Lg}{(N_{min} - 1)t_{min}}$$

$$= (1 + \frac{g}{(N_{min} - 1)t_{min}})L \quad (5)$$

Hence, we have the following result:

Theorem 2. *Any many-to-many traffic grooming algorithm that assumes that each session s_k is provisioned through $\left\lceil \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil$ identically ordered LCs for s_k is a $1 + \frac{g}{(N_{min}-1)t_{min}}$ approximation algorithm.*

An interesting case is when $(N_{min} - 1)t_{min} \geq g$ where we obtain an approximation ratio of at most 2. This relatively good approximation ratio is intuitive since when $(N_{min} - 1)t_{min} \geq g$, then each session's traffic efficiently fills at least half of its LCs. The best approximation ratio we can obtain is when $N_{min} = N$ and $t_{min} = g$ where we get an approximation ratio of $(1 + \frac{1}{(N-1)})$. On the other extreme, when $(N_{min} - 1)t_{min}$ is too small (e.g., equals to 1), then we obtain a $1 + g$ approximation ratio. This is also intuitive since when $(N_{min} - 1)t_{min} = 1$, then $\left\lceil \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil$ LCs for each session s_k may be a significant waste without inter-session grooming.

To further improve this algorithm we still assume that each session s_k is provisioned through $\left\lceil \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil$ identically ordered LCs for s_k. However, we now perform inter-session grooming so that LCs of different sessions may share lightpaths (i.e., lightpaths may groom traffic from different sessions and not just traffic from different members within the same session). The algorithm performs inter-session grooming as follows. Between each pair of nodes i and j, it grooms the $\sum_{s_k} (N_{s_k} - 1)t_{s_k}$ traffic units for all sessions s_k where $i, j \in m_{s_k}$ and member j follows member i immediately in the session's LCs. Note that the order of the members in the LCs is significant and must be taken into account to make inter-session grooming efficient. We start by assuming that members are ordered randomly in each session's LCs. Let S_{iFj} denotes the set of sessions where member i follows member j immediately in the session's LCs. The number of lightpaths P required according to this algorithm is given by:

$$P = \sum_{i=0}^{N-1} \sum_{j=0; j \neq i}^{N-1} \left\lceil \frac{\sum_{s_k \in S_{iFj}} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil$$

$$\leq \sum_{i=0}^{N-1} \sum_{j=0; j \neq i}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil$$

$$= \sum_{j=0}^{N-2} \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil$$

$$= (N - 1)L \quad (6)$$

The first inequality holds since S_{iFj} is a subset of S_i. The exchange of the summations in the second equality is valid since what is inside the inner summation $\left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k}-1)t_{s_k}}{g} \right\rceil$ is independent of j. This algorithm so far has an approximation ratio of min$\{\frac{g}{t_{min}}, 1 + \frac{g}{(N_{min}-1)t_{min}}, N - 1\}$. A better approximation ratio can be found by making a more intelligent ordering of the members in each session's LCs. We first order the nodes in the network in a list according to some criteria (e.g., ascending or descending order). Afterwards, for each session s_k, we order members in the session's LCs according to the list of ordered nodes. More precisely, we order members in a session s_k LCs by placing the first member as the first node in the list that is a member in session s_k and the second member as the second node in the list that is a member in session s_k and so on until we place all the members. Note that the first member immediately follows the last member in the ordered LCs. Based on this ordering of the members in the sessions' LCs, we have the following lemma:

Lemma 3. *A node i in the LCs Algorithm cannot have direct* *lightpaths incoming from more than $N - N_{min} + 1$ other nodes.*

Proof: We prove the lemma by proving that the $N_{min} - 2$ nodes that immediately follow i in the list of ordered nodes cannot have direct lightpaths outgoing to i (note that the first node in the list immediately follows the last node in the list). To prove this, we consider any node j in these $N_{min} - 2$ nodes. We have two cases for j. Either j comes after i in the list (i.e., between i and the last node in the list) or before i in the list (i.e., between the first node in the list and i).

In the first case when j comes after i in the list, the only way that j could have a direct lightpath outgoing to i is when i is the first member in the LCs for a session and j is the last one. Since the session size is at least N_{min} then there should be at least $N_{min} - 2$ other nodes in the session. Also, since members in the LCs are ordered according to the list of ordered nodes, then these $N_{min} - 2$ nodes must be between i and j in the list. However, there are at most $N_{min} - 3$ nodes between i and j which makes a contradiction.

In the second case when j comes before i in the list, the only way that j could have a direct lightpath outgoing to i is when j immediately precedes i in the LCs for a session. This prevents all the nodes between j and i in the list (which are at least $N - (N_{min} - 2) - 1 = N - N_{min} + 1$) to be members in the session. Hence, only $N - (N - N_{min} + 1) - 2 = N_{min} - 3$ nodes are left to be members in the session. However, since the session size is at least N_{min} then there should be at least $N_{min} - 2$ other nodes in the session, which makes a contradiction. Therefore, the $N_{min} - 2$ nodes that immediately follow i in the list of ordered nodes cannot have direct lightpaths outgoing to i, which means that i cannot have direct lightpaths incoming from more than $N - N_{min} + 1$ other nodes. ∎

After this ordering of the members in each session's LCs, between each pair of nodes i and j, the LCs Algorithm grooms the $\sum_{s_k} (N_{s_k} - 1)t_{s_k}$ traffic units for all sessions $s_k \in S_{jFi}$. The number of lightpaths P required by the LCs Algorithm is given by:

$$P = \sum_{i=0}^{N-1} \sum_{j=0;j\neq i}^{N-1} \left\lceil \frac{\sum_{s_k \in S_{iFj}} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil$$

However, from lemma 3, j cannot take more than $N - N_{min} + 1$ values and since S_{iFj} is a subset of S_i, then we have:

$$\begin{aligned} P &\leq \sum_{i=0}^{N-1} \sum_{j=0}^{N-N_{min}} \left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil \\ &= \sum_{j=0}^{N-N_{min}} \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil \\ &= (N - N_{min} + 1)L \end{aligned} \quad (7)$$

Therefore, we have the following result:

Theorem 4. *The LCs Algorithm is a min$\{g, 1 + \frac{g}{(N_{min}-1)t_{min}}, N - N_{min} + 1\}$ approximation algorithm.*

Note that when $N_{min} = N$ (i.e., all-to-all communication), then the LCs Algorithm guarantees an optimal solution. On the other extreme when $N_{min} = 2$, then we are back to the $\min\{g, 1 + \frac{g}{(N_{min}-1)t_{min}}, N - 1\}$ approximation ratio.

1 **Algorithm 1.** Lightpath Cycles Algorithm
2 Initialize lists $\mathcal{U} = \phi$, $\mathcal{Y} = V$, $\mathcal{X}_{s_k} = \phi$ (for all $1 \leq k \leq K$) and counters $c_1 = 1$ and $c_2 = 0$.
3 **for** *each ordered pair of nodes* (i, j) **do**
4     $rem_{ij} = g - \left(\sum_{s_k \in S_{ij}} (N_{s_k} - 1)t_{s_k} \right) \% g.$
5   **if** $rem_{ij} = g$ **then**
6         $rem_{ij} = 0.$
7   **end**
8 **end**
9 select a node $v \in \mathcal{Y}$ randomly and let $\mathcal{U}[0] = v$.
10 remove v from $\mathcal{Y}$.
11 **while** $\mathcal{Y}$ *is not empty* **do**
12   select a node $w \in \mathcal{Y}$ that has the smallest rem_{vw} value.
13     $\mathcal{U}[c_1{+}{+}] = w.$
14   remove w from $\mathcal{Y}$.
15     $v = w.$
16 **end**
17 **for** *each session* s_k, $1 \leq k \leq K$ **do**
18     $c_2 = 0.$
19   **for** $i = 0, 1..., N - 1$ **do**
20   **if** $\mathcal{U}[i] \in m_{s_k}$ **then**
21             $\mathcal{X}_{s_k}[c_2{+}{+}] = \mathcal{U}[i].$
22   **end**
23   **end**
24 **end**
25 **for** *each ordered pair of nodes* (i, j) **do**
26     $P_{ij} = \left\lceil \frac{\sum_{s_k \in S_{jFi}} (N_{s_k} - 1)t_{s_k}}{g} \right\rceil.$
27 **end**

Although any order of the nodes in the network will guarantee the above approximation ratio, the LCs Algorithm orders the nodes in a way to make inter-session grooming efficient (the full description of the algorithm is shown in Algorithm 1). For each ordered pair of nodes (i, j), the algorithm computes the rem_{ij} value (lines 3-8) which represents the remaining unused capacity on lightpaths from i to j if we place j after i in the LCs for all sessions in the set S_{ij}. If this value is low (e.g., close to 0), then placing j after i results in an efficient grooming of traffic into lightpaths. However, when this value is high (e.g., close to $g - 1$), then placing j after i results in an inefficient grooming where lightpaths are low utilized. The algorithm then orders the nodes in the network in the list $\mathcal{U}$ (lines 9-16) according to the rem_{ij} values as follows. It selects the first node v in the list randomly and then places the next node w in the list as the node with the smallest rem_{vw} value and it keeps doing this until it selects all the nodes in the network. Afterwards, for each session s_k, the algorithm orders members in the session's LCs in the list $\mathcal{X}_{s_k}$ (lines 17-24) as follows. It places the first member in $\mathcal{X}_{s_k}$ as the first node in the list $\mathcal{U}$ that is a member in session s_k and the second member in $\mathcal{X}_{s_k}$ as the second node in the list $\mathcal{U}$ that is a member in session s_k and it keeps doing this until it places all the members. Finally, the algorithm computes the total number of lightpaths needed between each ordered pair of nodes (i, j) to groom the total traffic $\sum_{s_k} (N_{s_k} - 1)t_{s_k}$ from all sessions $s_k \in S_{jFi}$ (lines 25-27).

1 **Algorithm 2.** Many-to-Many Traffic Grooming: Hub-Based
2 $max = 0$.
3 **for** $i = 0, 1..., N-1$ **do**
4 $\quad I_i = \left\lceil \sum_{s_k \in S_i} \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil$.
5 $\quad O_i = \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k}}{g} \right\rceil$.
6 $\quad$ **if** $I_i + O_i > max$ **then**
7 $\qquad h = i$.
8 $\qquad max = I_i + O_i$.
9 $\quad$ **end**
10 **end**
11 **for** $i = 0, 1..., N-1$ $(i \neq h)$ **do**
12 $\quad P_{ih}$=O_i.
13 $\quad P_{hi}$=I_i.
14 **end**

E. Hub Algorithm

In this algorithm, a hub node h is chosen from the set of nodes in the network. The traffic between any two members in a many-to-many session is routed as follows. First, the traffic is routed through a direct lightpath from the first member to the hub and then through a direct lightpath from the hub to the second member. Note that when the hub is the first member then the first step is not needed and when it is the second member then the second step is not needed. According to this algorithm, for each node $i \neq h$ to receive all its traffic, it needs $\lceil \sum_{s_k \in S_i} \frac{(N_{s_k}-1)t_{s_k}}{g} \rceil$ lightpaths incoming from the hub and it needs $\lceil \sum_{s_k \in S_i} \frac{t_{s_k}}{g} \rceil$ lightpaths outgoing to the hub to send all its traffic. Therefore, the total number of lightpaths P required according to this algorithm is given by:

$$\begin{aligned} P &= \sum_{i=0;i\neq h}^{N-1} \left(\left\lceil \sum_{s_k \in S_i} \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil + \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k}}{g} \right\rceil \right) \\ &\leq \sum_{i=0;i\neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil + \sum_{i=0;i\neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{(N_{s_k}-1)t_{s_k}}{g} \right\rceil \\ &\leq 2L \end{aligned} \tag{8}$$

Therefore, we have the following result:

Theorem 5. *The Hub Algorithm is a 2-approximation algorithm.*

Note that the optimal way to select the hub node is to select the node h with the largest $\lceil \sum_{s_k \in S_h} \frac{(N_{s_k}-1)t_{s_k}}{g} \rceil + \lceil \sum_{s_k \in S_h} \frac{t_{s_k}}{g} \rceil$ value. This minimizes the total number of lightpaths in the network. The full description of this algorithm is shown in Algorithm 2. The algorithm first computes the values of I_i and O_i for all the nodes in the network and selects the hub node h as the node with the largest $I_i + O_i$ value (lines 3-10). Afterwards, the algorithm computes the total number of lightpaths needed between each node and the hub and between the hub and each node (lines 11-14).

F. Complexity Analysis

The LCs Algorithm requires a preprocessing step that constructs the sets S_{ij}. This step requires visiting all the K sessions for each pair (i, j), which in total requires $O(KN^2)$ time. Once these sets are constructed, then the rem_{ij} values can be computed in $O(N^2)$ time (lines 3-8). Afterwards, the list $\mathcal{U}$ is constructed in $O(N^2)$ time (lines 9-16) and the lists $\mathcal{X}_{s_k}$ are constructed in $O(KN)$ time (lines 17-24). Then, the LCs Algorithm needs to construct the sets S_{jFi}. This requires visiting all the members in all the lists $\mathcal{X}_{s_k}$ for each pair (i, j), which in total requires $O(KN^3)$ time. Once these sets are constructed, then the P_{ij} values can be computed in $O(N^2)$ time (lines 25-27). This drives the time complexity of the LCs Algorithm to $O(KN^3)$. The Hub Algorithm, on the other hand, requires a preprocessing step that constructs the sets S_i. This step requires visiting all the K sessions for each node i, which in total requires $O(KN)$ time. Once these sets are constructed, then the I_i, O_i and h values can be computed in $\mathbf{O(KN)}$ time (lines 3-10). Afterwards, the P_{ih} and P_{hi} values are computed in $O(N)$ time (lines 11-14). This drives the time complexity of the Hub Algorithm to $O(KN)$.

G. Routing and Wavelength Assignment

Once we solve the many-to-many traffic grooming problem and determine the set of lightpaths to be established, we can then consider the routing and wavelength assignment (RWA) problem. In this problem, we need to provision each of the lightpaths on the optical WDM network by determining: 1) the physical route of each lightpath on the network, and 2) the wavelength to assign to each lightpath while taking the wavelength continuity constraint (i.e., the same wavelength is used on all the links traversed by a lightpath) into account. The objective is to minimize the total number of wavelengths used W.

It is to be noted that the RWA becomes completely independent of the fact that we are studying many-to-many traffic once the grooming problem has been solved. In addition to this, the RWA problem has been extensively studied in the literature and it has been proven to be NP-complete. Therefore, we use one of the best existing heuristics for the RWA problem (the LFAP heuristic [3]) which has been shown to use a number of wavelengths that is close to that of a derived lower bound. For a detail description of the LFAP heuristic, the reader is referred to [3].

III. Asymmetric Traffic

All of the previous analysis and algorithms apply only to the symmetric traffic case where members within the same many-to-many session have the same traffic demand. In this section, we address the many-to-many traffic grooming problem in the more general asymmetric traffic case where members within the same session may have different traffic demands. In terms of notation, all of the symbols remain the same except t_{s_k} which will now be replaced by $t_{s_k,l}$ to denote the traffic demand of member $l \in m_{s_k}$ in session s_k, where $1 \leq t_{s_k,l} \leq g$.

First, we note that the definition of lightpath cycles given in Section II.A does not apply to the asymmetric traffic case since each of the members in a session may require a different amount of incoming traffic, and hence different number of incoming lightpaths. We note, however, that the cycle structure

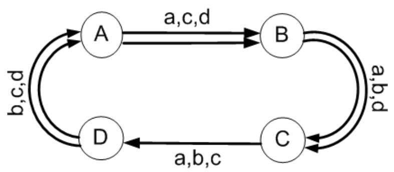


Fig. 4. Optimal provisioning of a single many-to-many session s_k with a set of members $m_{s_k} = \{A, B, C, D\}$ each with traffic denoted as a=1,b=2,c=3 and d=4, respectively (g=6).

with the same routing strategy used in lightpath cycles remains the optimal solution for a single session. To illustrate this, Fig. 4 shows the optimal provisioning of a single many-to-many session with a set of members $\{A, B, C, D\}$ each generating traffic denoted as $\{a, b, c, d\}$, respectively. The traffic demands are as follows $\{a = 1, b = 2, c = 3, d = 4\}$ and $g = 6$. Note that we are using the same routing strategy used in lightpath cycles where each member transmits its traffic in the cycle until it reaches the member just before it in the cycle. As in the symmetric traffic case, the optimality of this cycle structure comes from the fact that it is a feasible virtual topology and it meets a lower bound.

Next, we recompute the lower bound L as follows:

$$L = \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil \quad (9)$$

Note that Eq. (9) is a generalization of Eq. (1). Considering the same worst case scenario where no traffic grooming is perofrmed, we obtain the following upper bound:

$$P = \sum_{i=0}^{N-1} \sum_{s_k \in S_i} N_{s_k} - 1 \leq \sum_{i=0}^{N-1} \sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} \frac{t_{s_k,l}}{t_{min}}$$

$$\leq \sum_{i=0}^{N-1} \frac{g}{t_{min}} \left\lceil \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil = \frac{g}{t_{min}} L \quad (10)$$

Next, we extend the two algorithms introduced for the symmetric traffic case to the general asymmetric traffic case.

A. Lightpath Cycles Algorithm

This is the same algorithm as the one introduced for the symmetric traffic case except that many-to-many sessions are now provisioned using the general cycle structure described earlier rather than lightpath cycles. The description of the algorithm is shown in Algorithm 3. The only differences between this algorithm and Algorithm 1 are in lines 4 and 26 where the rem_{ij} and the P_{ij} values are calculated. The algorithm has exactly the same approximation ratios as Algorithm 1. To show this, we follow the same analysis provided in the symmetric traffic case. The total number of lightpaths P required according to this algorithm is given by:

$$P = \sum_{i=0}^{N-1} \sum_{s_k \in S_i} \left\lceil \frac{\sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil$$

$$\leq \sum_{i=0}^{N-1} \left(\left\lceil \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil + k_i \right)$$

$$= L + \sum_{i=0}^{N-1} k_i \quad (11)$$

Now, let us consider the lower bound L again:

$$L = \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil$$

$$\geq \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{min}}{g} \right\rceil$$

$$\geq \sum_{i=0}^{N-1} \frac{\sum_{s_k \in S_i} (N_{min} - 1) t_{min}}{g}$$

$$= \frac{(N_{min} - 1) t_{min}}{g} \sum_{i=0}^{N-1} k_i \quad (12)$$

Substituting (12) in (11), we have:

$$P \leq (1 + \frac{g}{(N_{min} - 1) t_{min}}) L \quad (13)$$

Also, the total number of lightpaths P required according to this algorithm can be expressed as follows:

$$P = \sum_{i=0}^{N-1} \sum_{j=0; j \neq i}^{N-1} \left\lceil \frac{\sum_{s_k \in S_{iFj}} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil$$

However, from lemma 3, j cannot take more than $N - N_{min} + 1$ values and since S_{iFj} is a subset of S_i, then we have:

$$P \leq \sum_{i=0}^{N-1} \sum_{j=0}^{N-N_{min}} \left\lceil \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil$$

$$= \sum_{j=0}^{N-N_{min}} \sum_{i=0}^{N-1} \left\lceil \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil$$

$$= (N - N_{min} + 1) L \quad (14)$$

B. Hub Algorithm

This is the same algorithm as the one introduced for the symmetric traffic case except that the total number of lightpaths between nodes and the hub is computed differently. The total number of lightpaths P is given by:

$$P = \sum_{i=0; i \neq h}^{N-1} \left(\left\lceil \sum_{s_k \in S_i} \frac{\sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil + \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k,i}}{g} \right\rceil \right)$$

$$\leq L + \sum_{i=0; i \neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k,i}}{g} \right\rceil \quad (15)$$

Note that $\sum_{i=0; i \neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k,i}}{g} \right\rceil \leq \sum_{i=0; i \neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{\sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil$. The reason is that $\sum_{i=0; i \neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k,i}}{g} \right\rceil$ counts each traffic stream $t_{s_k,i}$ exactly once at the source member i of that traffic stream, while $\sum_{i=0; i \neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{\sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil$ counts each traffic stream $t_{s_k,i}$ once at each destination member interested in

Algorithm 3. Lightpath Cycles Algorithm: Asymmetric Traffic

Initialize lists $\mathcal{U} = \phi,\ \mathcal{Y} = V,\ \mathcal{X}_{s_k} = \phi$ (for all $1 \leq k \leq K$) and counters $c_1 = 1$ and $c_2 = 0$.
for *each ordered pair of nodes* (i, j) **do**
    $rem_{ij} = g - \left(\sum_{s_k \in S_{ij}} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l} \right) \% g.$
    if $rem_{ij} = g$ **then**
        $rem_{ij} = 0.$
    end
end
select a node $v \in \mathcal{Y}$ randomly and let $\mathcal{U}[0] = v$.
remove v from $\mathcal{Y}$.
while $\mathcal{Y}$ *is not empty* **do**
    select a node $w \in \mathcal{Y}$ that has the smallest rem_{vw} value.
    $\mathcal{U}[c_1{+}{+}] = w.$
    remove w from $\mathcal{Y}$.
    $v = w.$
end
for *each session* $s_k,\ 1 \leq k \leq K$ **do**
    $c_2 = 0.$
    for $i = 0, 1..., N-1$ **do**
        if $\mathcal{U}[i] \in m_{s_k}$ **then**
            $\mathcal{X}_{s_k}[c_2{+}{+}] = \mathcal{U}[i].$
        end
    end
end
for *each ordered pair of nodes* (i, j) **do**
    $P_{ij} = \left\lceil \frac{\sum_{s_k \in S_{jFi}} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil.$
end

Algorithm 4. Hub Algorithm: Asymmetric Traffic

2 $max = 0$
3 **for** $i = 0, 1..., N-1$ **do**
4     $I_i = \left\lceil \sum_{s_k \in S_i} \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil.$
5     $O_i = \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k,i}}{g} \right\rceil.$
6   **if** $I_i + O_i > max$ **then**
7         $h = i.$
8         $max = I_i + O_i.$
9   **end**
10 **end**
11 **for** $i = 0, 1..., N-1$ $(i \neq h)$ **do**
12   P_{ih}=O_i.
13   P_{hi}=I_i.
14 **end**

receiving this traffic stream ($l \in m_{s_k}, l \neq i$). Hence, we have:

$$P \leq 2L \tag{16}$$

Therefore, the Hub Algorithm in the asymmetric traffic case is also a 2-approximation algorithm. Note that the optimal way to select the hub node is to select the node h with the largest $\left\lceil \sum_{s_k \in S_i} \frac{\sum_{s_k \in S_i} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil + \left\lceil \sum_{s_k \in S_i} \frac{t_{s_k,i}}{g} \right\rceil$ value. This minimizes the total number of lightpaths in the network. The full description of this algorithm is shown in Algorithm 4. The only differences between this algorithm and Algorithm 2 are in lines 4 and 5 where the I_i and the O_i values are calculated.

IV. Performance on Other Objectives

Although the main objective of the LCs and the Hub Algorithms is to minimize the total number of lightpaths, we in this section, study the performance of the two algorithms on other important objectives such as the number of logical hops traversed by a traffic stream, total amount of electronic switching at a node, and Min-Max objectives.

A. Number of Logical Hops

The number of logical hops (i.e., lightpaths) traversed by a traffic stream is considered an important performance metric in optical networks since it reflects the number of times the traffic stream undergoes optical-to-electronic (*O/E*) conversion which in turn affects the end-to-end delay. Let $l_{LCs}^{s_k,ij}$ be the number of logical hops traversed by the traffic stream originating from member $i \in m_{s_k}$ and destined to member $j \in m_{s_k}$ according to the LCs Algorithm. According to the LCs Algorithm, the traffic stream originating from i traverses one lightpath to the member that immediately follows i in the LCs for s_k and two lightpaths to the member after it and finally $N_{s_k} - 1$ lightpaths to the member that immediately comes before i in the LCs for s_k. Therefore, we have the following upper bound:

$$l_{LCs}^{s_k,ij} \leq N_{s_k} - 1 \tag{17}$$

Let $l_{LCs}^{s_k,i}$ be the average number of logical hops traversed by a traffic stream originating from member $i \in m_{s_k}$ according to the LCs Algorithm. $l_{LCs}^{s_k,i}$ can be computed as follows:

$$l_{LCs}^{s_k,i} = \frac{1 + 2 + ... + (N_{s_k} - 1)}{N_{s_k} - 1} = \frac{(N_{s_k} - 1)N_{s_k}}{2(N_{s_k} - 1)} = \frac{N_{s_k}}{2} \tag{18}$$

Let $l_{LCs}^{s_k}$ be the average number of logical hops traversed by a traffic stream in session s_k according to the LCs Algorithm. Note that the value of $l_{LCs}^{s_k,i}$ is the same for all $i \in m_{s_k}$. Therefore, $l_{LCs}^{s_k}$ is equal to $l_{LCs}^{s_k,i}$ for any $i \in m_{s_k}$. Finally, the average number of logical hops traversed by a traffic stream according to the LCs Algorithm (l_{LCs}) can be computed as follows:

$$l_{LCs} = \frac{\sum_{s_k} l_{LCs}^{s_k}}{K} = \frac{\sum_{s_k} N_{s_k}}{2K} \tag{19}$$

Following the same notations for the Hub Algorithm, we have:

$$l_{Hub}^{s_k,ij} = \begin{cases} 2, & \text{if } i \neq h \text{ } and \text{ } j \neq h \\ 1, & \text{otherwise} \end{cases} \tag{20}$$

Therefore, we have the following upper bound:

$$l_{Hub}^{s_k,ij} \leq 2 \tag{21}$$

To compute the values of $l_{Hub}^{s_k,i}$, we first consider the case where $h \in m_{s_k}$ and $i \neq h$. In this case, we have:

$$\begin{aligned} l_{Hub}^{s_k,i} = \frac{1 \times 1 + (N_{s_k} - 2) \times 2}{N_{s_k} - 1} &= \frac{2(N_{s_k} - 1) - 1}{N_{s_k} - 1} \\ &= 2 - \frac{1}{N_{s_k} - 1} \end{aligned} \tag{22}$$

The other two cases is when $h \notin m_{s_k}$ where we have $l_{Hub}^{s_k,i} = 2$ and when $h \in m_{s_k}$ and $i = h$ where we have $l_{Hub}^{s_k,i} = 1$. The three cases are summarized as follows:

$$l_{Hub}^{s_k,i} = \begin{cases} 2, & \text{if } h \notin m_{s_k} \\ 2 - \frac{1}{N_{s_k}-1}, & \text{if } h \in m_{s_k} \text{ } and \text{ } i \neq h \\ 1, & \text{if } h \in m_{s_k} \text{ } and \text{ } i = h \end{cases} \tag{23}$$

To compute the values of $l_{Hub}^{s_k}$, we have two cases. In the first case where $h \notin m_{s_k}$, we have $l_{Hub}^{s_k} = 2$. In the second case where $h \in m_{s_k}$, we have:

$$\begin{aligned} l_{Hub}^{s_k} &= \frac{1 \times 1 + (N_{s_k} - 1) \times (2 - \frac{1}{N_{s_k}})}{N_{s_k}} \\ &= \frac{2(N_{s_k} - 1)}{N_{s_k}} + \frac{1}{N_{s_k}^2} \end{aligned} \tag{24}$$

The two cases are summarized as follows:

$$l_{Hub}^{s_k} = \begin{cases} 2, & \text{if } h \notin m_{s_k} \\ \frac{2(N_{s_k}-1)}{N_{s_k}} + \frac{1}{N_{s_k}^2}, & \text{if } h \in m_{s_k} \end{cases} \tag{25}$$

Finally, we have:

$$l_{Hub} = \frac{\sum_{s_k : h \in m_{s_k}} \left(\frac{2(N_{s_k}-1)}{N_{s_k}} + \frac{1}{N_{s_k}^2} \right) + \sum_{s_k : h \notin m_{s_k}} 2}{K} \tag{26}$$

B. Total Amount of Electronic Switching

The amount of electronic switching at a node equals to the total number of traffic streams that this node needs to switch in the electronic domain. This is considered an important cost metric in optical networks since it directly affects the size of the switch at that node. According to the LCs Algorithm, a node i will receive $N_{s_k} - 1$ traffic streams from each session $s_k \in S_i$. For each session $s_k \in S_i$, node i terminates one of the traffic streams, switches $N_{s_k} - 2$ traffic streams and adds its own traffic stream (see Fig. 4). Let e_{LCs}^i and e_{LCs} denote the total amount of electronic switching at node i and the total amount of electronic switching in the whole network (at all nodes) according to the LCs Algorithm, respectively. Then, we have the following:

$$e_{LCs}^i = \sum_{s_k \in S_i} (N_{s_k} - 2) \tag{27}$$

$$e_{LCs} = \sum_{i=0}^{N-1} \sum_{s_k \in S_i} (N_{s_k} - 2) \tag{28}$$

To bound the values of e_{Alg1}^i and e_{Alg1}, we consider the worst case scenario where each of the K sessions has N members. In this case, node i has to switch $K(N-2)$ traffic units. Therefore, we have:

$$e_{LCs}^i \leq K(N-2) \tag{29}$$

$$e_{LCs} \leq KN(N-2) \tag{30}$$

According to the Hub Algorithm, the only node that performs electronic switching is the hub node h. Note that a traffic stream received at the hub and needs to be delivered to multiple recipients requires the hub to duplicate the traffic stream and to switch each copy separately. Following the same notations for the Hub Algorithm, we have the following:

$$\begin{aligned} e_{Hub} = e_{Hub}^h = &\sum_{s_k : h \notin m_{s_k}} N_{s_k}(N_{s_k} - 1) \\ &+ \sum_{s_k : h \in m_{s_k}} (N_{s_k} - 1)(N_{s_k} - 2) \end{aligned} \tag{31}$$

To bound the value of e_{Alg2}, we consider the worst case scenario where each of the K sessions has N members. In this case, node h has to switch $K(N-1)(N-2)$ traffic streams. Hence, we have:

$$e_{Hub} = e_{Hub}^h \leq K(N-1)(N-2) \tag{32}$$

C. Min-Max Objectives

In many situations, it is desirable to minimize the maximum of a certain cost metric among all the nodes in the network (e.g., minimizing the maximum number of lightpaths incoming/outgoing at a node or minimizing the maximum amount of electronic switching at a node). Note that if the objective is just to minimize the total number of lightpaths in the network, we may end up with a solution where certain nodes have a large number of lightpaths incoming and outgoing while other nodes have very few. This is generally not desirable since the first kind of nodes may be too expensive or impractical to deploy [27].

First, we consider the maximum number of lightpaths incoming or outgoing at a node according to the LCs and the Hub Algorithms (P_{LCs}^{max} and P_{Hub}^{max}, respectively). According to the LCs Algorithm, the total number of lightpaths incoming to a node is equal to the total number of lightpaths outgoing. Hence, we only focus on the maximum number of lightpaths incoming at a node which can be expressed as follows:

$$P_{LCs}^{max} = max_i \left\{ \sum_{j=0; j \neq i}^{N-1} \left\lceil \frac{\sum_{s_k \in S_{iFj}} \sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil \right\} \tag{33}$$

To bound P_{Alg1}^{max}, we consider the worst case scenario where each of the K sessions has N members each with traffic demand g. In this case, we have the following upper bound:

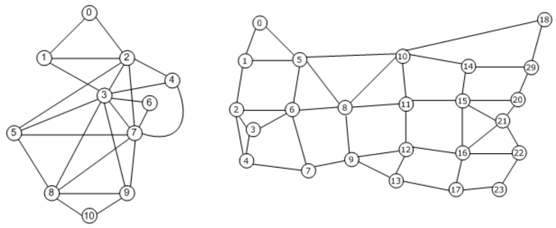
$$P_{LCs}^{max} \leq K(N-1) \tag{34}$$

According to the Hub Algorithm, the hub h has the maximum number of lightpaths outgoing among all the nodes in the network. Hence, we have:

$$P_{Hub}^{max} = \sum_{i=0; i \neq h}^{N-1} \left\lceil \sum_{s_k \in S_i} \frac{\sum_{l \in m_{s_k}; l \neq i} t_{s_k,l}}{g} \right\rceil \tag{35}$$

$$P_{Hub}^{max} \leq K(N-1)^2 \tag{36}$$

Next, we consider the maximum amount of electronic switching at a node according to the LCs and the Hub Algorithms (e_{LCs}^{max} and e_{Hub}^{max}, respectively). These values are



(a) NJ-LATA (b) USNET

Fig. 6. Networks used in the results

computed and bounded as follows:

$$e^{max}_{LCs} = max_i \left\{ \sum_{s_k \in S_i} (N_{s_k} - 2) \right\} \leq K(N-2) \quad (37)$$

$$e^{max}_{Hub} = e^{h}_{Hub} \leq Kg(N-1)(N-2) \quad (38)$$

V. Numerical Results

In this section, we conduct extensive experiments to evaluate the performance of the LCs and the Hub Algorithms. First, we show that the two algorithms use a number of lightpaths that is significantly close to that of the derived lower bound L. Second, we compare the performance of the two algorithms on the several objectives mentioned in the paper including the number of lightpaths, number of wavelengths, number of logical hops traversed by a traffic stream, total amount of electronic switching at a node, and Min-Max objectives.

We consider three sample networks in our experiments. One is the NJ-LATA network (shown in Fig. 6.(a)) consisting of 11 nodes and 23 links, the USNET (shown in Fig. 6.(b)) consisting of 24 nodes and 43 links, and the 47-node, 96-link network (which appeared in [28]). We run our experiments using the symmetric LCs algorithm (Algorithm 1) and the symmetric Hub algorithm (Algorithm 2). We randomly generate K many-to-many session requests as follows. The size of a session is randomly selected between [N_{min},N], while members in a session are randomly selected between [0,$N-1$]. The traffic demand of a member in a many-to-many session is randomly selected between [1,8]. We study the performance of each algorithm by varying one of the parameters K, g and N_{min} at a time. Figs. 5.(a), 5.(d) and 5.(g) plot the number of lightpaths P versus the number of sessions K on NJ-LATA, USNET and the 47-node network, respectively with $g = 32$ and $N_{min} = 2$. Figs. 5.(b), 5.(d) and 5.(h) plot the number of lightpaths P versus the grooming factor g on NJ-LATA, USNET and the 47-node network, respectively with $K = 100$ and $N_{min} = 2$. Finally, Figs. 5.(c), 5.(f) and 5.(i) plot the number of lightpaths P versus the minimum session size N_{min} on NJ-LATA, USNET and the 47-node network, respectively with $K = 100$ and $g = 32$.

We can see from the results in Fig. 5 that solutions obtained from the LCs and the Hub Algorithms are significantly close to the derived lower bound L on a wide range of network parameters K, g and N_{min}. Since the optimal solution lies between the lower bound and the best of the two algorithms,

TABLE II
Comparison between the LCs and the Hub Algorithms on objectives $\bar{l}$, $\bar{e}$, $\overline{P_{max}}$, $\overline{e_{max}}$ and $\overline{W}$ on NJ-LATA

	$\bar{l}$	$\bar{e}$	$\overline{P_{max}}$	$\overline{e_{max}}$	$\overline{W}$
LCs Algorithm	3.3	6,801	78	432	66
Hub Algorithm	1.34	6,283	991	6,283	195

TABLE III
Comparison between the LCs and the Hub Algorithms on objectives $\bar{l}$, $\bar{e}$, $\overline{P_{max}}$, $\overline{e_{max}}$ and $\overline{W}$ on USNET

	$\bar{l}$	$\bar{e}$	$\overline{P_{max}}$	$\overline{e_{max}}$	$\overline{W}$
LCs Algorithm	6.8	37,730	173	1,067	305
Hub Algorithm	1.4	36,441	5,358	36,441	1,501

we conclude that the two algorithms give near-optimal solutions and that the lower bound L is tight. We also note that we conducted the same experiments using the asymmetric LCs and Hub algorithms and the same conclusions were drawn.

Next, we compare the LCs and the Hub Algorithms on the various objectives mentioned in the paper. Let $\bar{l}$, $\bar{e}$, $\overline{P_{max}}$, $\overline{e_{max}}$ and $\overline{W}$ denote the average value of the number of logical hops traversed by a traffic stream, total amount of electronic switching in the network, maximum number of lightpaths incoming/outgoing at a node, maximum amount of electronic switching at a node, and the number of wavelengths used, respectively for all the experiments conducted above on a certain network by a certain algorithm. Tables II and III show the values $\bar{l}$, $\bar{e}$, $\overline{P_{max}}$, $\overline{e_{max}}$ and $\overline{W}$ on NJ-LATA and USNET, respectively using the two algorithms.

We can see from Tables II and III that traffic streams in the Hub Algorithm traverse fewer number of logical hops than traffic streams in the LCs Algorithm, while the total amount of electronic switching in the network by the two algorithms is almost the same. We can also see that the Hub Algorithm performs poorly with Min-Max objectives compared to the LCs Algorithm. This is expected since the hub node in the Hub Algorithm terminates and originates a large number of lightpaths and switches a large number of traffic streams, while the LCs Algorithm distributes and balances the number of lightpaths and traffic streams among the different nodes in the network through the use lightpath cycles.

Next, we compare the LCs and the Hub Algorithms in terms of the number of lightpaths required. Note that the approximation ratio $1 + \frac{g}{(N_{min}-1)t_{min}}$ of the LCs Algorithm becomes better than the 2-approximation ratio of the Hub Algorithm when $(N_{min}-1)t_{min} > g$, while it is worst when $(N_{min}-1)t_{min} < g$. Hence, the comparison between the two algorithms is dependent on traffic granularities and on the size of many-to-many sessions.

First, we assume that the size of many-to-many sessions is randomly selected between [2,N] and we compare the two algorithms by varying traffic granularities of sessions in the network. To make the comparison, we assume a static uniform traffic with all members in all sessions in an experiment having the same traffic demand t, where $1 \leq t \leq g$. We generate 50 experiments on the USNET each with 100 many-to-many session requests as follows.

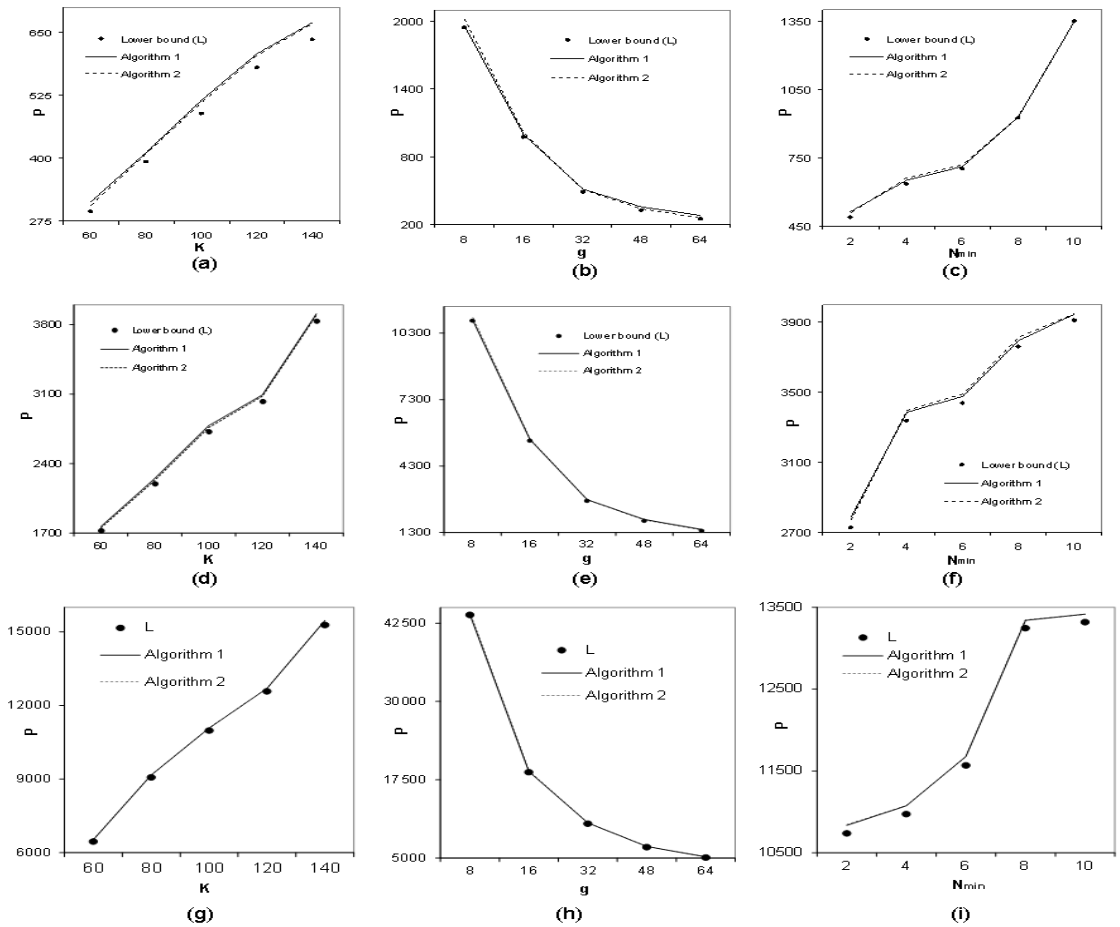


Fig. 5. Performance study of the symmetric LCs algorithm (Algorithm 1) and the symmetric Hub algorithm (Algorithm 2) on the number of lightpaths P versus number of sessions K, grooming factor g and minimum session size N_{min} on the NJ-LATA topology (a)-(c); on the USNET topology (d)-(f); and on the 47-node and 96-edge network in [28] (g)-(i).

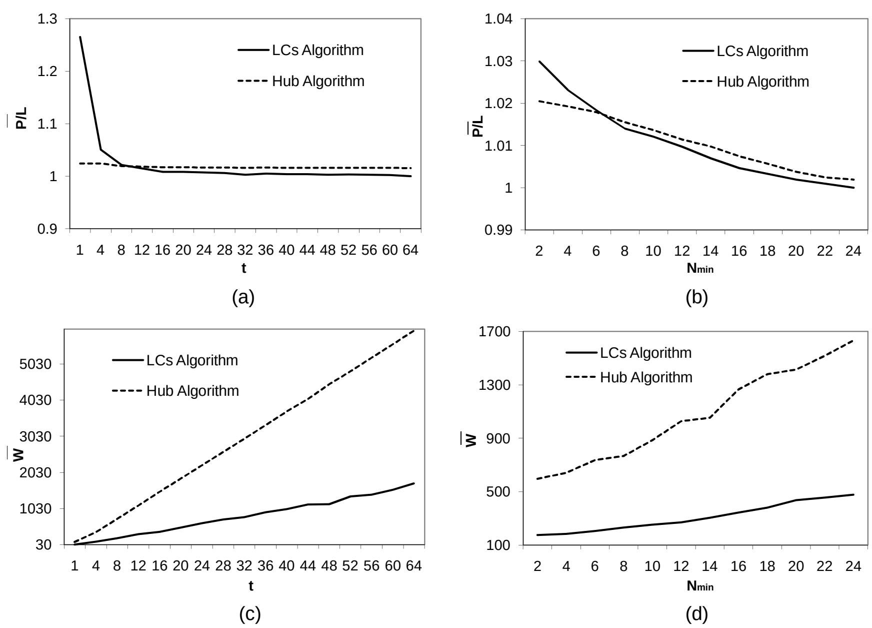


Fig. 7. (a): $\overline{P/L}$ versus t on USNET. (b): $\overline{P/L}$ versus N_{min} on USNET. (c): $\overline{W}$ versus t on USNET. (d): $\overline{W}$ versus N_{min} on USNET

The size of a session is randomly selected between [2,24], while members in a session are randomly selected between [0,23]. Given the uniform traffic assumption, each of the 50 experiments is conducted for each value of $t = \{1, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44, 48, 52, 56, 60, 64\}$ ($g = 64$) by each algorithm. We define the *normalized* *number of lightpaths* as the ratio of the number of lightpaths P to the lower bound L (P/L) in an experiment. We also define $\overline{P/L}$ to be the average value of all P/L values obtained from the 50 experiments at a particular value of t by a certain algorithm. The corresponding values of $\overline{P/L}$ are shown in Fig. 7.(a).

Second, we assume that the traffic demand of a member in a session is randomly selected between [1,12] ($g = 64$) and we compare the two algorithms by varying the minimum session size N_{min}. At each value of $N_{min} = \{2,4,6,8,10,12,14,16,18,20,22,24\}$, we conduct 50 experiments on the USNET each with 100 many-to-many session requests as follows. The size of a session is randomly selected between [N_{min},24], while members in a session are randomly selected between [0,23]. The traffic demand of a member in a session is randomly selected between [1,12] ($g = 64$). The resulting values of $\overline{P/L}$, which is now defined as the average value of all P/L values obtained from the 50 experiments at a particular value of N_{min} by a certain algorithm, are shown in Fig. 7.(b).

After determining the set of lightpaths for each experiment at each value of t (or N_{min}) by each algorithm, these lightpaths are routed and assigned a wavelength according to the LFAP heuristic [3]. We define $\overline{W}$ to be the average value of all W values obtained from the 50 experiments at a particular value of t (or N_{min}) by a certain algorithm. The resulting values of $\overline{W}$ versus t and versus N_{min} are shown in Figs. 7.(c) and 7.(d), respectively.

We can see from Fig. 7.(a) that the Hub Algorithm is more cost-effective than the LCs Algorithm in packing and grooming low granularity traffic (e.g., $t \leq \frac{g}{8}$), while the LCs Algorithm is more cost-effective when traffic granularities of sessions are relatively high (e.g., $t > \frac{g}{8}$). Also, from Fig. 7.(b), we can see that the Hub Algorithm is more cost-effective than the LCs Algorithm when the minimum session size is relatively low (e.g., $N_{min} \leq \frac{N}{3}$), while the LCs Algorithm is more cost-effective when the minimum session size is relatively high (e.g., $N_{min} > \frac{N}{3}$). Finally, from Figs 7.(c)-(d), we can see that the Hub Algorithm consumes much more wavelengths than the LCs Algorithm. The reason is that all the lightpaths generated by the Hub Algorithm are between a certain pair of nodes (nodes and the hub). This results in a large number of lightpaths routed on the same link (hence, using a large number of wavelengths). The LCs Algorithm, on the other hand, distributes the number of lightpaths among the different pairs of nodes in the network through the use of lightpath cycles. This balances the number of lightpaths to be routed on the same link resulting in a fewer number of wavelengths used.

VI. Conclusions and Future Work

In this paper, we have studied the many-to-many traffic grooming problem in optical WDM mesh networks. First, two novel approximation algorithms were introduced for the symmetric traffic case. The LCs Algorithm, which is based on lightpath cycles, has an approximation ratio of $\min\{g, 1 + \frac{g}{(N_{min}-1)t_{min}}, N - N_{min} + 1\}$, while the Hub Algorithm, which is based on a hub node that collects and distributes traffic, has a 2-approximation ratio. These two algorithms were extended to the asymmetric traffic case and it was shown that the approximation ratios still hold. We have also studied the two algorithms on other important objectives such as the number of logical hops traversed by a traffic stream, total amount of electronic switching at a node, and Min-Max objectives. Through extensive experiments, we have shown that the two algorithms perform significantly close to the derived lower bound L.

Although we have compared the performance of our algorithms on other important objectives in Section IV, we have not designed the algorithms to produce solutions that are close to optimal or to some derived lower bounds for those objectives. We believe this multi-objective optimization problem is a very interesting and challenging problem that we plan to investigate in our future work.

References

[1] B. Quinn and K. Almeroth, “IP multicast applications: challenges and solutions,” IETF Request for Comments (RFC) 3170, Sept. 2001.

[2] C. Diot et al.“Multipoint Communication: A Survey of Protocols, Functions, and Mechanisms,” *IEEE J. Select. Areas Commun.,* vol. 15, pp. 277-290, 1997.

[3] H. Siregar, H. Takagi, and Y. Zhang, “Efficient routing and wavelength assignment in wavelength-routed optical networks,” *Proc. 7th Asia-Pacific Network Operations and Management Symp.* Oct. 2003.

[4] R. Dutta, S. Huang, and G. N. Rouskas, “Traffic grooming in path, star, and tree networks: Complexity, bounds, and algorithms,” *IEEE J. Sel. Areas Commun.* vol. 24, no. 4, pp. 66-82, Apr. 2006.

[5] S. Antonakopoulos and L. Zhang,“Approximation Algorithms for Grooming in Optical Network Design,” in *Proc. IEEE Infocom’09.*

[6] B. Chen, G. Rouskas, and R. Dutta, “On Hierarchical Traffic Grooming in WDM Networks,” *IEEE/ACM Transactions on Networking,* vol. 16, no. 5, pp. 1226-1238, Oct. 2008.

[7] J.Q. Hu et al., Traffic Grooming, Routing, and Wavelength Assignment in Optical WDM Mesh Networks, in *Proc. IEEE Infocom’04.*

[8] X. Zhang and C. Qiao,“An effective and comprehensive approach for traffic grooming and wavelength assignment in SONET/WDM rings,” *IEEE/ACM Trans. Networking,* vol. 8, no. 5, pp. 608-617, Oct. 2000.

[9] Ul-Mustafa, R. and A. E. Kamal,“Grooming of Non-Uniform Traffic on Unidirectional and Bidirectional Rings,” *Elsevier Computer Communications,* vol. 29, no. 8, pp. 1065-1078, May 2006.

[10] 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,* vol. 11, no. 2, pp. 285-299, April 2003.

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

[12] K. Zhu and B. Mukherjee,“Traffic Grooming in a WDM Mesh Network,” *IEEE J. Select. Areas Commun.,* vol. 20, pp. 122-133, 2002.

[13] O. Gerstel, R. Ramaswami, and G. Sasaki, “Cost-effective Traffic Grooming in WDM Rings,” *IEEE/ACM Transactions on Networking,* vol. 8, no. 5, pp. 618-630, Oct. 2000.

[14] R. Dutta and G. N. Rouskas,“Traffic grooming in WDM networks: past and future,” *IEEE Network,* vol. 16, no. 6, pp. 46-56, Nov./Dec. 2002.

[15] H. Madhyastha et al., “Grooming of multicast sessions in metropolitan WDM ring networks,” *Computer Networks,* vol. 49, pp. 561-579, 2005.

[16] A. Rawat, R. La, S. Marcus, and M. Shayman, “Grooming Multicast Traffic in Unidirectional SONET/WDM Rings,” *IEEE Journal on Selected Areas in Communication,* vol. 25, no. 6, pp. 70-83, August 2007.

[17] A. Billah, B. Wang, and A. Awwal, “Multicast traffic grooming in WDM optical mesh networks,” in *Proc. GLOBECOM’03.*

[18] R. Ul-Mustafa and A. E. Kamal,“Design and provisioning of WDM networks with multicast traffic grooming,” *IEEE J. Select. Areas Commun.,* vol. 24, no. 4, pp. 37-53, Apr. 2006.

[19] G. Chowdhary and C. S. R. Murthy,“Grooming of multicast sessions in wdm mesh networks,” in *Workshop on Traffic Grooming,* 2004.

[20] R. Ul-Mustafa and A. E. Kamal,“Many-to-one Traffic Grooming with Aggregation in WDM Networks ” *IEEE J. Select. Areas Commun.,* vol. 24, pp. 68-81, 2006.

[21] A. E. Kamal,“Algorithms for Multicast Traffic Grooming in WDM Mesh Networks,” *IEEE Communications* pp. 96-105, Nov. 2006.
[22] R. Dutta, A. Kamal, and G. Rouskas ,“Traffic Grooming for Optical Networks: Foundations, Techniques and Frontiers” *Springer,* August 2008.
[23] Long L. and A. E. Kamal, “Reducing network cost of many-to-many communication in unidirectional WDM rings with network coding,” accepted for publication in *IEEE/OSA Journal of Lightwave Technology*.
[24] M. Saleh and A. Kamal, “Many-to-Many Traffic Grooming in WDM Networks,” *IEEE/OSA Journal of Optical Communications and Networking*, pp. 376-391, vol. 1, no. 5, Oct. 2009.
[25] M. Saleh and A. Kamal, “Design and Provisioning of WDM Networks with Many-to-Many Traffic Grooming,” *IEEE/ACM Transactions on Networking* vol. 18, no. 6, pp. 1869-1882, Dec. 2010.
[26] M. Saleh and A. Kamal, “Approximation Algorithms for Many-to-Many Traffic Grooming in WDM Mesh Networks,” in *proc. IEEE Infocom 2010*.
[27] B. Chen, G. Rouskas and R. Dutta, “Traffic Grooming in WDM Ring Networks with the Min-Max Objective ,” in *Proc. of IFIP NETWORKING 04,*.
[28] P. Baran,“On distributed communications networks,” *IEEE Trans. Commun.*, vol. COM-12, no. 1, pp. 19, Mar. 1964.

PLACE PHOTO HERE

Mohammad A. Saleh is a software engineer at Google Inc. He received his Ph.D. in Computer Engineering from Iowa State University, USA, in 2011. His research interests include optical WDM networks, wireless networks and network survivability.

PLACE PHOTO HERE

Ahmed E. Kamal (S’82-M’87-SM’91-F’12) received a B.Sc. (distinction with honors) and an M.Sc. both from Cairo University, Egypt, and an M.A.Sc. and a Ph.D. both from the University of Toronto, Canada, all in Electrical Engineering in 1978, 1980, 1982 and 1986, respectively. He is currently a professor of Electrical and Computer Engineering at Iowa State University. Earlier he held faculty positions in the Department of Computing Science at the University of Alberta, Canada, and the Department of Computer Engineering at Kuwait University, Kuwait. He was also an adjunct professor at the Telecommunications Research Labs, Edmonton, Alberta.

Kamal’s research interests include optical networks, wireless sensor networks, cognitive radio networks, and performance evaluation. He is a Fellow of the IEEE, a senior member of the Association of Computing Machinery, and a registered professional engineer. With his students, Kamal received the 1993 IEE Hartree Premium for papers published in Computers and Control in IEE Proceedings for his paper entitled Study of the Behaviour of Hubnet, and the best paper award of the IEEE Globecom 2008 Symposium on Ad Hoc and Sensors Networks Symposium.

Kamal served on the technical program committees of numerous conferences. He was als the chair or co-chair of the Technical Program Committees of a number of conferences including the IEEE Communication Networks Services Research (CNSR) conference 2006, the Optical Symposium of IEEE/CreatNet Broadnets 2006, and the Optical Networks and Systems Symposia of the IEEE Globecom 2007, and the IEEE Globecom 2010, the 2008 ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-08), the ACM International Conference on Information Science, Technology and Applications, 2009, and the IEEE Multimedia Communications and Information Technology conference, 2010. He will be serving as the co-chair of the IEEE Globecom Cognitive Radios and Networks Symposium in 2012. He is on the editorial boards of the Computer Networks journal, the Optical Switching and Networking journal, both published by Elsevier, and the IEEE Communications Surveys and Tutorials.