Multicast Traffic Grooming with Leaking Strategy in WDM Mesh Networks

Rongping Lin, *Student Member, IEEE*, Wen-De Zhong, *Senior Member, IEEE*, Sanjay Kumar Bose, *Senior Member, IEEE*, Moshe Zukerman, *Fellow, IEEE*

***Abstract*—The ever-increasing popularity and traffic volume of multicast applications motivates the need for development of methodologies for traffic management and network design that especially cater for multicast traffic. Addressing the disparity between the bandwidth offered by a wavelength and the bandwidth required by a single connection is a key challenge in the efficient usage of any WDM network. This problem is also relevant to WDM networks that support multicast traffic, and can be mitigated by multicast traffic grooming. This paper considers multicast traffic grooming with a leaking strategy where a light-tree may deliver the traffic of a multicast connection to nodes which are not in the destination set of the connection. This leaking strategy improves the sharing of light-trees and add/drop ports, leading to lower blocking ratios. Two multicast traffic grooming algorithms with leaking strategy, namely, multicast traffic leaky grooming (MTLG) and multicast traffic hybrid grooming (MTHG), are proposed. MTLG grooms traffic to light-trees if the traffic leaked is less than a given threshold value. MTHG first grooms traffic to light-trees without leaking; if some destinations remain, it then grooms traffic to light-trees with leaking. MTHG is an improvement over MTLG as it can attain higher light-tree sharing with less traffic leaked. Simulations show that the two proposed algorithms perform better than other algorithms at low add/drop port ratios with MTHG showing better performance.**

***Index Terms*—Multicast, grooming with leaking, light-tree, wavelength- division multiplexing (WDM)**

I. Introduction

MULTICASTING applications such as IPTV, video conferencing, streamed video broadcasting and distance learning are becoming increasingly popular and are expected to be the major drivers of Internet traffic growth. The traffic generated by these applications is likely to be transported over optical wavelength-division multiplexed (WDM) networks in the foreseeable future. While a wavelength channel can carry traffic at multi-gigabit rate, most multicast applications generally require only sub-wavelength bandwidth; e.g., standard definition TV (SDTV) and high definition TV (HDTV) require bandwidth of about 6 Mb/s and 25 Mb/s respectively [1]. This creates potential inefficiency because of the disparity between the bandwidth required by a connection and the bandwidth offered by a wavelength channel. Multicast traffic grooming is a technique that enables multiple sub-wavelength multicast connections or flows to share a wavelength channel so that the overall bandwidth can be efficiently utilized.

Since the ultimate goal of optical network design is to minimize cost, it is important to realize that the more significant CAPEX cost is that of the higher layer electronic components (e.g., Add/Drop Multiplexers (ADMs), or IP routers [2]–[5]) rather than the cost of the bandwidth. Moreover, the ever-increasing number of wavelengths that can be simultaneously transmitted in a fiber tends to decreases the wavelength cost over time. It should also be noted that any higher layer electronic component will have only a limited number of electronic ports. Therefore significant CAPEX savings on electronic components can be made by minimizing the number of ports used.

A. Background and Related Work

Multicast traffic from a single source to multiple destinations normally follows a tree topology. Therefore, it is natural that a *light-tree* (i.e. an extension of a lightpath to a tree topology) can significantly improve efficiency and reduce the cost of supporting multicast applications [6]. In [7] [8], a light-tree is called *logical one hop tree* (LOHT) in the logical layer, where it is represented as a set of direct links from a root to each destination. A transmission from the root to each destination takes only one hop and is done all-optically without any intervention at the intermediate nodes. In [7] [8], to improve the resource utilization, traffic grooming strategies not only allow multiple connections to share a light-tree, but a low bandwidth connection request can also be supported by several interconnected smaller light-trees to form a larger tree reaching all the destinations. Electronic packet switching with optical-electronic-optical (OEO) conversion forwards traffic from an upstream light-tree to downstream light-trees through the connecting nodes.

Based on whether or not the connection requests are known a priori, multicast traffic grooming is categorized as being either dynamic or static [9]. In the static case, the set of multicast connection requests is known a priori and the optimal solution for traffic grooming is usually obtained by solving an Integer

The work on this paper was done when R. Lin was a PhD student with EEE, Nanyang Technological University (NTU). He is now with the Dept. of EE, City University of Hong Kong, Hong Kong SAR, (e-mail: linr0012@e.ntu.edu.sg). W. D. Zhong is with the School of EEE, NTU, Singapore, (email: ewdzhong@ntu.edu.sg).

S. K. Bose is with the Indian Institute of Technology, Guwahati, India (email: skbose@iitg.ernet.in).

M. Zukerman is with the Dept. of EE, City University of Hong Kong, Hong Kong SAR, (e-mail: m.zu@cityu.edu.hk).

Linear Programming (ILP) problem with the objective of minimizing the resource used (e.g. wavelength or add/drop port) or maximizing the network throughput. In [10], the routing information of all multicast connection requests is known a priori and wavelength conversion is used. Therefore, the multicast traffic grooming problem was reduced to a bin packing problem on each link. The authors of [11] presented a tri-partite graph model to solve the multicast flow aggregation problem under a static traffic scenario. In order to improve light-tree utilization, multicast traffic aggregated to a light-tree is allowed to leak to unrelated nodes, which is called "tail waste". However in [11], a multicast traffic connection is accommodated by a single light-tree, which may result in high tail wastage (i.e., low utilization of light-trees). In this work, instead of using a single light-tree, we consider multiple small light-trees to support multicast connections so as to improve the sharing of light-trees. In [12], an ILP optimization problem is formulated to design a light-tree based logical topology with delay bounds. Here, the light-trees are somehow randomly chosen which reduces the complexity of the problem, but may not achieve the optimal result. In our earlier work [13], we studied the problem of optimizing the cost for multicast traffic grooming based on light-tree, and proposed a light-tree based ILP formulation to minimize network cost in terms of the number of higher layer electronic ports and the number of wavelengths used.

In the dynamic case, multicast connection requests arrive dynamically and both routing and wavelength assignment are dynamically decided for new requests [7] [8] [14]–[19]. The objective is to maximize the resource utilization while maintaining an acceptable level of blocking probability. In [14], both single hop (SH) and multi-hop (MH) algorithms were proposed. In SH, connections with the same source and destinations can be groomed together. The MH algorithm grooms a connection to a light-tree which has the same destinations and then uses a lightpath to connect the source of the connection request to the root of that light-tree. Since the probability of multiple connection requests having exactly the same destinations is very low, the light-tree sharing is quite poor. In [15], a Multicast Tree Decompose (MTD) algorithm was proposed to improve the blocking probability by grooming traffic partially to several light-trees. However, achieving high utilization with low blocking still remains a challenge in this system. In our earlier work [7] [8], we divided light-trees into smaller ones to improve sharing so that low OEO overhead can be achieved with low blocking probabilities.

B. Motivations and Contributions

As mentioned earlier, overall costs can be reduced by minimizing the use of electronic ports. This is especially important in the case of light-tree based WDM networks because every light-tree uses multiple ports, i.e. an add port and several drop ports. For general WDM networks, considerable research [2]–[8] has been reported on reducing the usage of add/drop ports or increasing their utilizations so that more connection requests can be efficiently accommodated in the network. These general principles also apply here, i.e., increasing the utilization of light-trees reduces the usage of electronic ports and lowers the overall cost.

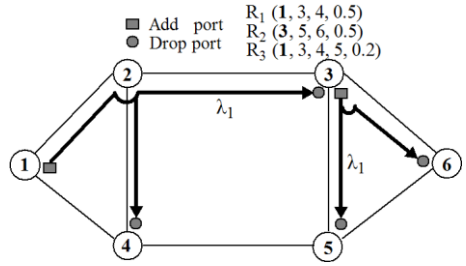

Fig. 1. Multicast traffic grooming with leaking.

In [7] [8], we proposed a multicast traffic grooming algorithm *Light-Tree Division - Adjacent Node Component based Grooming algorithm* (LTD-ANCG), which can increase the light-tree sharing (add/drop ports utilization) by dividing light-trees into adjacent node components (ANC). These are small light-trees within two optical hops, and traffic is groomed to those ANCs whose destination sets are subsets of the destinations to be reached. Although this strategy increases the sharing of light-trees, applying the strict condition that the destinations of the ANC must be a subset of the destination nodes to be reached limits the extent of sharing when the add/drop port resource is limited. In the following, a multicast connection request is denoted R(**s**, d_1, d_2, … d_n, *b*), where **s** is the source, d_1 to d_n are the destinations, and *b* is the required bandwidth. A light-tree is denoted as a LOHT t{r -> d_1, d_2, … d_m} where r is the root, and d_1 to d_m are the destinations. As an example, consider the scenario of Fig. 1 where every node has only one transceiver (one add port and one drop port) and where two multicast connection requests, R_1 (**1**, 3, 4, *0.3*) and R_2 (**3**, 5, 6, *0.3*), with bandwidth requirements of 0.3 each are already in the network. Two light-trees t_1{1->3,4} and t_2{3->5,6} are used to accommodate the two connection requests R_1 and R_2, respectively. Assuming the (normalized) bandwidth of a wavelength to be 1, this leaves a bandwidth of 0.7 on each tree to accommodate other connection requests, e.g., a new request R_3 with a bandwidth requirement of 0.2. According to the LTD-ANCG algorithm, the existing light-trees t_1{1->3,4} is selected to groom the traffic as the destination set {3,4} of the light-trees is a subset of the connection request destinations {3,4,5}. The light-trees t_2{3->5,6} is not selected to groom traffic R_3 as node 6 is not included in the destinations of R_3. The LTD-ANCG algorithm then tries to build a new light-tree (which in this case, is actually a lightpath) from node 4 or node 3 to node 5 since this is the only remaining destination. However, this cannot be done as there are no drop ports left at node 5, since the only one available has already been used by t_2{3->5,6}). This blocks the new connection request R_3 in the LTD-ANCG algorithm. However, if traffic grooming with leaking is allowed, after t_1{1->3,4} is selected to reach node 3 and node 4, t_2{3->5,6} is selected to transmit the traffic of R_3 to node 5 even through this will cause traffic to leak to node 6, i.e., node 6 then discards the traffic of R_3. This example illustrates

that multicast traffic grooming with leaking may increase the sharing of light-trees at the cost of leaking traffic to unintended nodes, i.e., sharing t_2\{3->5,6\} with leaking enables to accommodate the connection request R_3.

Apart from improving the utilization of light-trees, traffic grooming with leaking may also increase the probability that a connection request is successfully accommodated. Since a multicast session is formed by multiple small light-trees, if there is traffic leaking to unrelated nodes, then starting from these nodes, we may be able to find optimal light-trees to reach other destinations. This results in more light-trees being sourced from the leaves of the grooming trees than when traffic grooming without leaking is implemented.

In this paper, we investigate multicast traffic grooming with this leaking strategy in dynamic traffic scenarios. When the add/drop port resource is limited, this leaky grooming strategy has the potential to improve the sharing of light-trees. We propose two multicast traffic grooming algorithms with the leaking strategy: *Multicast Traffic Leaky Grooming* algorithm (MTLG) and *Multicast Traffic Hybrid Grooming* algorithm (MTHG). MTLG grooms traffic to light-trees only if the leaked traffic is less than a pre-specified threshold. MTHG first grooms traffic to light-trees without leaking; if this fails to reach some destinations, then it further grooms the traffic to light-trees with leaking. Our studies demonstrate that, for the examples considered, both the proposed algorithms perform better than the LTD-ANCG algorithm proposed earlier for low add/drop port ratios, and much better than MTD, SH and MH algorithms given in the literature. MTHG shows a further improvement over MTLG as it can utilize light-trees better with less leaking traffic. We also consider the routing problem of multicast connections, where multiple light-trees form a larger tree session to accommodate a multicast connection request. A new routing algorithm, *Constrained Light-tree Multicast Routing* (CLMR), is also proposed which consumes fewer add/drop ports and exhibits better performance.

C. Organization

The remainder of this paper is organized as follows. In Section II, a formal statement of the multicast traffic grooming problem is presented. Section III investigates the routing of multiple light-trees forming a larger tree session and also presents the new routing algorithm CLMR. In Section IV, we present the strategy for multicast traffic grooming with leaking and the details of our proposed grooming algorithms, MTLG and MTHG. Section V presents simulation results for the two algorithms and compares their performance with LTD-ANCG, MTD, SH and MH algorithms. Section VI concludes this paper.

II. General Problem Statement

The problem of grooming sub-wavelength multicast traffic flows into high bandwidth wavelength channels on a given WDM network can be stated as follows:

1) A physical topology $G(V, E)$ is an undirected graph representing a network, where V is the set of network nodes and E is the set of physical fiber links connecting the network nodes. (We assume that there are two fibers joining two adjacent nodes and transmitting in opposite directions if there is a link between them.)
2) Fibers of each link are identical and each fiber can simultaneously carry up to W wavelengths each having capacity C [b/s].
3) There are no wavelength converters in the network. This implies a light-tree must be set up on the same wavelength.
4) The arrivals of dynamic multicast connection requests are assumed to follow a Poisson process with sub-wavelength bandwidth requirements where the destinations are randomly (uniformly) selected out of the available destinations.

For a network of this type, our goal is to accommodate as many connection requests as possible, i.e., to minimize the connection blocking probability.

III. Routing with Add/drop Port Limitation

Since the light-tree based LTD-ANCG algorithm proposed in [7] [8] has been demonstrated to perform the best among the existing light-tree based multicast traffic grooming algorithms, we adopt similar ideas in our new traffic grooming algorithm with leaking. In particular, we use small light-trees (ANCs) to construct larger multicast sessions and encourage sharing of a small light-tree between several multicast connections.

Before describing our new grooming algorithms, we investigate first the routing problem of small light-trees as that directly affects the performance of our grooming algorithms. The *Light-Tree Division - Adjacent Node Component* (LTD-ANC) sub-algorithm was proposed in [7] [8] to divide a light-tree into adjacent node components which are small light-trees within two optical hops. The grooming algorithm LTD-ANCG uses the LTD-ANC sub-algorithm to route new light-trees for the remaining nodes which could not be reached by that grooming procedure. In Fig. 2, two kinds of light-trees are included in adjacent node components, referred to as 1-hop light-trees and 2-hop branches. Here optical splitting is done at the roots of the 1-hop light-trees and the intermediate nodes of the 2-hop branches. With these two kinds of light-trees, a larger tree session can be formed by OEO conversion at the connection nodes so that the smaller light-trees can be interconnected.

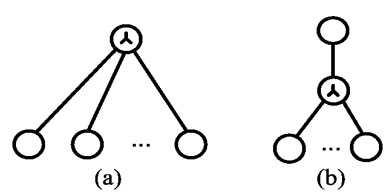

Fig. 2. Adjacent node component (ANC). (a) 1-hop light-trees (b) 2-hop branches.

Existing multicast routing algorithms (including LTD-ANC) usually try to build light-trees with the minimal total link cost (or minimal number of links) to accommodate a multicast connection request. However, when dividing a light-tree into

small components, add/drop ports need to be introduced at the dividing nodes to connect the component light-trees. Since the routing algorithm does not guarantee minimal usage of add/drop ports, building a light-tree with a minimal number of wavelinks and then dividing it into smaller ones may substantially increase the usage of add/drop ports. For networks where add/drop ports are a scarce resource, new algorithms need to be developed to reduce the usage of add/drop ports rather than reduce the usage of wavelinks. We propose next a new *Constrained Light-tree Multicast Routing* (CLMR) algorithm to efficiently perform multicast routing in networks where the add/drop port resource is limited.

A. Constrained Light-tree Multicast Routing algorithm (CLMR)

Using multiple small light-trees to construct a large tree session for a multicast connection can improve the sharing degree of light-trees in multicast grooming scenarios, as small light-trees can be easily shared by multiple connections. Since multicast connections consume more drop ports than add ports (the example of Fig. 1 also illustrates this), drop ports are usually exhausted before add ports in a multicast traffic network. This implies that we should preferentially focus on reducing the usage of drop ports when designing such networks. We have observed that connecting multiple small light-trees at a node to form a bigger tree consumes one drop port and multiple (at least one) add ports. The idea of the CLMR algorithm is to build small light-trees (ANCs) between the source and the destinations in a way that reduces the number of connecting nodes, i.e., nodes other than the source and destination nodes. It may also be noted that connecting light-trees at a destination node of a light-tree can reduce one drop port usage since such a port is already used there anyway. The pseudo codes of the CLMR algorithm are given as follows:

Pseudo Codes of CLMR Algorithm

Input:
A network $G(V, E)$, and a multicast connection request (s, D)
Output:
A set of adjacent node components T, which forms a larger tree session to support the multicast connection request
Algorithm BEGIN

```
S = {s}
While D ≠ Φ do
    //2-hop branch building
        Start from every node in S, to find a 2-hop branch with
         the maximal destination size, whose root i is in S and
         whose destinations D' is a subset of D
        If such a 2-hop branch exists
            Save this 2-hop branch into set T
            Delete D' from D
            Add D' into S
            Continue    //start while loop again
    //1-hop light-tree building
        Start from every node in S, to find a 1-hop light-tree
         with the maximal destination size, whose root i is in S
         and whose destinations D' is a subset of D
        If such a 1-hop light-tree exists
            Save this 1-hop light-tree into set T
            Delete D' from D
            Add D' into S
            Continue    //start while loop again
    // extension path building
        Check all shortest paths of node pairs between S and
         D, select the path P with the shortest length.
        Save the path P into set T
        Add the destination of P into S
        Continue    //start while loop again
End While
END
```

The algorithm consists of three parts: 2-hop branch building, 1-hop light-tree building and extension path building. These three parts are included in the **While** loop, so every time after an ANC has been built, the **Continue** statement starts the **While** loop once again. The 2-hop branch building is positioned at the beginning of the algorithm, so that only when no further 2-hop branches are available, does the algorithm go to the 1-hop light-tree building to derive 1-hop light-trees. When neither the 2-hop branch nor 1-hop light-tree are available, the algorithm goes to the extension path building. The reason for introducing the extension path building in the algorithm is that using 2-hop branches and 1-hop light-trees to construct a large tree session to accommodate connections may fail as some destination nodes may be very far from all the other nodes (i.e. more than 2 hops away). In the extension path building, a shortest path to the closest destination node is built. Then from the destination of the path, a new ANC may be derived. The CLMR algorithm tries to build ANCs between the source node and the destinations in such a way that it can take advantage of the drop ports already at the destination nodes.

In the 2-hop branch building, a 2-hop branch is selected after $O(|V||D|^2)$ computations. This is because starting from a source node, it checks every node of D as the first hop and then check all nodes in D as the second hop (actually, only adjacent nodes are checked). In the 1-hop light-tree building, the complexity is $O(|V||D|)$ as every node in D is checked once when a source node is fixed. The process of constructing the extension path has complexity of $O(|V|^2|D|)$. This shows that the complexity of the CLMR algorithm is $O(|V|^2|D|^2)$.

B. Optimal Routing

We use ILP formulation to model the optimal routing for a multicast connection request. A multicast connection request is given, and the ILP formulation is to find the optimal routing of ANCs for the request. This ILP is a simpler version of the one in [13], where it was used for multiple connection requests and had a larger light-tree search space.

Given

m and *n*: two endpoints of a physical fiber link.

{*s*, *D*}: a 2-tuple of the elements *s* and *D* representing a multicast connection request, where *s* is the source of the request, $D = \{d_1, d_2, d_3, \ldots\}$ is the destination set of the request.

(*i*, *J*): the LOHT sourcing at node *i* terminating at the destination set *J* in the logical layer. An ANC in the optical layer maps to a LOHT in the logical layer.

J_m : all LOHT destination sets from node *m*, which includes all ANCs from node *m* and also every unicast destination with more than two hops from node *m*.

Variables

L_{iJ} : a Boolean variable. It is 1 if the connection request traverses LOHT (*i*, *J*), otherwise 0.

Q_j^{iJ} : a Boolean variable. It is 1 if the connection request traverses LOHT (*i*, *J*) and node *j* is included in the set *J*. i.e. $j \in J$, otherwise 0.

H_n : an integer variable used to break loops , which could be the delay from source *s* to node *n*.

F_{mn}^{iJ} : an integer variable, the number of streams which traverse link (*m*, *n*) in LOHT (*i*, *J*). Each node in *J* needs one stream; therefore, there are |*J*| streams out of source node *i*.

Minimize:
$$\sum_i \sum_{J \in J_i} L_{iJ} + \sum_i \sum_{J \in J_i} \sum_{j \in J} Q_j^{iJ} \tag{1}$$

The objective function is to minimize the number of higher layer electronic ports (add ports and drop ports) to accommodate the multicast connection request {*s*, *D*}. The first part of the objective function is the number of add ports used, and the second part is the number of drop ports used. The set of constraints is given below.

$$Q_j^{iJ} = L_{iJ} \qquad \forall i, J \in J_i, j \in J \tag{2}$$

$$\sum_{J \in J_S} L_{sJ} \geq 1 \tag{3}$$

$$\sum_i \sum_{J \in J_i} Q_j^{iJ} = 1 \qquad \forall j \in D, j \in J \tag{4}$$

$$\sum_{J \in J_q} L_{qJ} \leq |J_q| \cdot \sum_i \sum_{J \in J_i} Q_q^{iJ} \qquad \forall q \neq s \tag{5}$$

$$H_n \geq H_m + 1 - (1 - \sum_{J \in J_m} Q_n^{mJ}) \cdot |V| \qquad \forall n, \forall m \neq n \tag{6}$$

$$\sum_m F_{mi}^{iJ} = 0 \qquad \forall i, J \in J_i \tag{7}$$

$$\sum_n F_{in}^{iJ} = |J| \cdot L_{iJ} \qquad \forall i, J \in J_i \tag{8}$$

$$\sum_m F_{mj}^{iJ} - \sum_n F_{jn}^{iJ} = Q_j^{iJ} \qquad \forall i, J \in J_i, j \in J \tag{9}$$

Equation (2) is to set the value of Q_j^{iJ} to be 1 if LOHT (*i*, *J*) is used by the connection request and node *j* belongs to the destination set of the LOHT. Equation (3) ensures that the number of outgoing streams from the source node s is not less than one. Equation (4) constrains the incoming stream of each destination to be 1. Equation (5) ensures that except the source node, the root of a LOHT that supports the connection request must be a destination of another LOHT which also supports the request. Equation (6) ensures that, if node *m* and *n* are in the same LOHT that supports the connection request, the delay from the source *s* to node *n* is larger than that to node *m*, where *m* is the root of the LOHT and *n* belongs to the destinations of the LOHT.

Equations (7)-(9) are to constrain the routing of LOHTs used by the connection request. Equation (7) ensures that there is no stream flowing into the source of LOHT. Equation (8) ensures that if a LOHT is used by the connection request, the number of streams flowing out of the source equals the number of the LOHT destinations. Equation (9) ensures that the outgoing stream is one less than the incoming stream for the destination node of LOHT, and that for other intermediate nodes, which are neither a source nor one of the destinations, the incoming and outgoing steams are the same.

C. Simulation Results

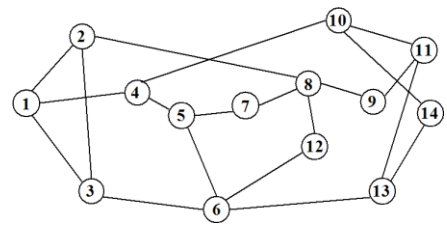

Fig. 3. NSFNET topology.

We compare the performances of CLMR, LTD-ANC and the optimal results derived by the ILP for the NSFNET network with 14 nodes shown in Fig. 3. The source of the connection request is randomly chosen from the network nodes, and the destination nodes are also randomly selected from the network nodes (excluding the source node). In each simulation experiment, one multicast connection request is generated as the input. To obtain the optimal solution, we use a commercial ILP solver, CPLEX [20], to solve the ILP formulation. We here compare the performance of the algorithms in terms the numbers of electronic ports (add ports and drop ports) and wavelinks required by a multicast connection. The results are shown in Figs. 4 and 5 where the values at each destination size are the average value of 100 simulation runs.

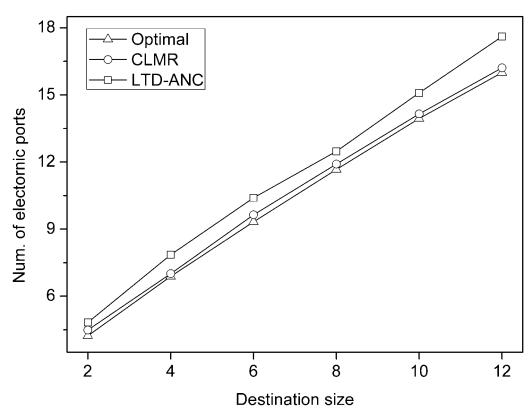

Fig. 4 Comparison of number of electronic ports required by a multicast connection for different routing algorithms.

Fig. 4 shows that for all algorithms, when the destination size increases, more electronic ports are consumed as more destinations would lead to bigger multicast sessions. We can see that the optimal ILP approach consumes the least number of electronic ports. Our proposed algorithm CLMR uses fewer electronic ports than LTD-ANC. It may be noted that CLMR uses almost the same number of electronic ports as the optimal ILP approach and that this is about 9% less than what is needed on the average for LTD-ANC. The numbers of wavelinks used by these schemes are compared in Fig. 5. As shown, more wavelinks are used when destination size increases as the larger multicast sessions occupy more wavelinks. The optimal ILP approach has the lowest value, followed by LTD-ANC and then CLMR. In summary, we find that our proposed CLMR algorithm uses almost as many electronic ports as the optimal ILP approach but performs better than LTD-ANC.

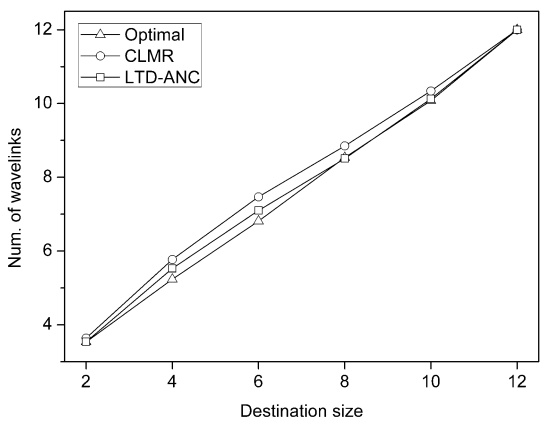

Fig. 5 Comparison of number of wavelinks required by a multicast connection for different routing algorithms.

IV. Multicast Traffic Grooming Algorithms with Leaking Strategy

Traffic grooming with leaking can improve the utilization of add/drop ports at the expense of traffic leaking to unrelated nodes. This may cause some other undesirable side effects. For example, if too much traffic is leaked then the utilization of add/drop ports may degrade as resources are wasted. This implies that traffic leaking and resource sharing should be balanced and its impact on blocking performance should be examined. Consider a multicast connection request (s, D), where s is the source and D is the set of destinations. If a LOHT (r, L), where r is the root and L is the destination set of the LOHT, is selected to groom the traffic, then the leaking ratio is defined as

$$\text{leaking ratio} = \frac{|L| - |D'|}{|L|} \tag{10}$$

where $D' = D \cap L$. If leaking ratio is 0, then there is no leaking and the grooming strategy is the same as that for LTD-ANCG. For our proposed grooming algorithms, if the traffic leaking of a selected LOHT is below a given leaking ratio threshold then the grooming operation for that LOHT is considered acceptable. We propose here two multicast traffic grooming algorithms with the leaking strategy, i.e., *Multicast Traffic Leaky Grooming* (MTLG) and *Multicast Traffic Hybrid Grooming* (MTHG) algorithms. The pseudo codes of MLTG algorithm are given as follows:

Pseudo Codes of MTLG algorithm

Input:
Ex contains all existing LOHTs in the current graph, a multicast connection request (s, D, f), and a leaking ratio threshold b
Output:
A set of LOHTs R which is the traffic routing of the request
Algorithm BEGIN
//select existing LOHTs to groom request with leaking,
1. **While** $D \neq \Phi$ **do**
2. Select a LOHT for which its destination set has the maximal intersection to D under the requirements of having sufficient available bandwidth and leaking ratio $< b$. If no LOHT that meets these requirements is found, break the While loop
3. Reduce available bandwidth of the LOHT by f
4. Delete all the destinations of the LOHT from D
5. Add the LOHT into R
6. **End While**
//Build new light-trees
7. **If** $D \neq \Phi$
8. **If** $f < C$
9. Call **CLMR algorithm** to build light-trees from s to D, if it cannot build the trees, block the connection
10. **If** $f == C$
11. Build a light-tree from s to D, if can not build the tree, block the connection
12. Assign wavelengths, add/drop ports to light-tree(s), if cannot assign resources, block the connection
13. Set available bandwidth of the light-tree(s) as $C-f$, and add light-trees into R

END

In MTLG algorithm, the **While** loop is to groom the traffic to those existing LOHTs which have maximal intersection to the destinations to be reached D while ensuring that the traffic leaking ratio is lower than the leaking ratio threshold b and with enough available bandwidth. It is noted that in our code implementations, if the LOHT has destinations which have already been reached, this LOHT will not be selected, and if the root of LOHT is not in the multicast session, the root is added into the unreached destination set D. Lines 3 and 4 are to update the available bandwidth of the selected LOHT and the unreached destination set D. Line 5 is to include the selected LOHT into R.

After the leaky grooming, if there are destinations still unreached then new light-trees are derived to support them from lines 7 to 13. The CLMR algorithm is called to build light-trees at line 9, then resources are allocated at line 12, and finally the available bandwidth of new light-trees is set to $C-f$. We note that the connection requests with the full bandwidth requirement are not groomed. Instead, a new light-tree with the minimal total link cost is derived by the same algorithm as LTD-ANC in lines

10 and 11, as this consumes the least add/drop ports and wavelinks.

The MTHG algorithm is similar to the MTLG algorithm, except that the MTHG grooms traffic to LOHTs without leaking first. If some destinations remain to be reached, it then grooms traffic to LOHTs with leaking in the same way as in the MTLG algorithm. Therefore, MTHG is an improvement over MTLG as it is able to support more resource sharing with less leaking traffic. The pseudo codes of the MTHG algorithm are given as follows:

Pseudo Codes of MTHG algorithm

Input:
Ex contains all existing *LOHTs* in the current graph, a multicast connection request (*s, D, f*), and a leaking ratio threshold *b*
Output:
A set of LOHTs *R* which is the traffic routing of the request
Algorithm BEGIN
//select existing LOHTs to groom traffic without leaking,
1. **While** $D \neq \Phi$ **do**
2. Select a LOHT for which its destination set is the maximal subset of *D* under the requirement of having sufficient available bandwidth. If no LOHT that meets the requirement is found, break the While loop
3. Reduce available bandwidth of the LOHT by *f*
4. Delete all the destinations of the LOHT from *D*
5. Add the LOHT into *R*
6. **End While**
//select existing LOHTs to groom traffic with leaking
7. Using lines 1 to 13 of the MTLG algorithm to groom the traffic for the rest of unreached destination *D*

END

In the **While** loop of the MTHG algorithm, the LOHT with the maximal subset of the unreached destination set *D* and with enough available bandwidth is selected. Since there is no traffic leaking at this stage, the destinations of the selected LOHT must be a subset of *D*. The root of LOHT is added into *D* if the root is not in the multicast session. If some destinations remain to be reached, MTHG grooms traffic with leaking in the same manner as the MTLG algorithm.

In the MTHG algorithm, the first while loop is to select existing LOHTs to groom traffic without leaking, which checks all existing LOHTs. This has complexity of $O(|V|\,2^{|r|^2})$, where *r* is the maximal nodal degree (the complexity is from the checking of $|V|$ nodes each of $2^{|r|^2}$ LOHTs). Therefore, the first loop has complexity of $O(|V|^2\,2^{|r|^2})$. It is clear that the grooming with leaking in the second loop is also with the complexity $O(|V|^2\,2^{|r|^2})$. Building a new light-tree has complexity $O(|V|^2 \log|V|)$, while using CLMR to build light-trees has complexity $O(|V|^2|D|^2)$. Therefore the MTHG algorithm has overall complexity of $O(|V|^2\,2^{|r|^2} + \max(|V|^2 \log|V|, |V|^2|D|^2))$. The MTLG algorithm also has the same complexity.

V. Simulation Results

The LTD-ANCG algorithm was proposed in [7] [8], where it was seen to be the best light-tree based multicast grooming algorithm when compared with SH, MH and MTD by simulations. In this section, to make a complete comparison, we compare our two proposed algorithms to LTD-ANCG, SH, MH and MTD. We present simulation results for the NSFNET topology with the number of wavelengths, *W*, set to 32. Multicast connection request arrivals are assumed to follow a Poisson process with rate λ and holding times are negatively exponentially distributed with a mean $1/\mu$. Unless otherwise specified, the bandwidth required by connection requests is assumed to be uniformly distributed between (0, *C*], where *C* is the bandwidth of a wavelength channel. The source of a multicast connection request is randomly chosen from the network nodes. The multicast destination size is also randomly chosen. In the present case the choice for the destination size is between 1 and 13, and destination nodes are randomly selected from network nodes (excluding the source node). A connection request is immediately rejected if the network cannot accommodate it. A transmitter and a receiver are usually built together as a transceiver. Since a transmitter is connected to an add port for transmitting optical signals and a receiver is connected to a drop port for receiving optical signals, we assume that the number of add ports and drop ports are equal at a node. From cost considerations, add/drop ports may be sparse in a network, and the add/drop ratio is defined as the ratio of the number of add/drop ports per fiber to the number of wavelengths per fiber [21].

In Fig. 6, blocking ratios are compared when network load is 50 erlangs and the add/drop ratio is set to 0.3. When the leaking ratio threshold is varied from 0 to 0.9, the MTHG algorithm achieves the best performance, followed by MTLG, LTD-ANCG, MTD, MH and SH. MH and SH have almost the same blocking ratio which is the highest of all other algorithms. This is because these two algorithms use a similar procedure, and the high level of blocking ratio can be explained by the fact that the probability of multiple connection requests having exactly the same destinations is very low. MTD has a better performance than SH and MH, but worse than LTD-ANCG. The reason is that LTD-ANCG divides light-trees into small ones, which increases the utilization of the resources. The MTHG performs better than the MTLG, as MTHG grooms traffic without leaking in the first phase and then grooms traffic in the same way as MTLG in the second phase, resulting in less leakage (i.e. less wastage of resources). If leaking is not allowed (the leaking ratio threshold is 0), MTLG and MTHG have the same blocking ratio, as then MTHG becomes effectively the same as MTLG. In this case, even though all the algorithms operate without leaking, the blocking ratio of MTLG and MTHG is still smaller than that of LTD-ANCG. The reason for this is that the CLMR algorithm used by MTLG and MTHG can build light-trees with less add/drop ports than the LTD-ANC

algorithm used by LTD-ANCG, and can therefore accommodate more connections. When the leaking ratio threshold increases from 0 to 0.2, the blocking ratios of MTLG and MTHG decrease significantly. This is because of the gain provided by the leaking strategy. (When the leaking ratio threshold is 0.2, the blocking ratios reach the lowest value for this simulation scenario.) If the leaking ratio threshold is greater than 0.2, it is observed that the blocking ratios of both MTLG and MTHG increase with increase in the leaking ratio threshold. This means that in this case, traffic leaking of about 20% is preferred to attain the lowest blocking ratio. If the leaking ratio threshold is greater than 0.2, traffic leaking causes more resource wastage which consequently degrades the gains obtained from the leaky grooming strategy.

As shown in Fig. 7, if the add/drop ratio is changed to 0.5, SH, MH and MTD still perform worse than the others, MTHG still has the best performance and the lowest blocking ratio is still at the leaking ratio threshold of 0.2. However, MTLG now has a higher blocking ratio than LTD-ANCG when the leaking ratio threshold is larger than 0.8. The reason for this is that too much traffic is leaked when the leaking ratio threshold is increased to 0.8 leading to high resource wastage in MTLG.

In Fig. 8, we provide (for the purpose of completion) an extreme and probably unrealistic example, with add/drop ratio of 0.9, where our proposed solution is not superior. In this scenario, there are a fairly large number of add/drop ports and more light-trees are built. This means that the number of add/drop ports is sufficient, and therefore the wavelengths available now becomes the crucial resource in the network. In Fig. 8, SH, MH and MTD still have higher blocking ratios than the other algorithms due to low grooming efficiencies. As expected, LTD-ANCG is observed to perform the best to groom traffic without leaking as it builds light-trees with fewer wavelinks and grooming without leaking does not waste wavelinks. In contrast, in this extreme case, our solution suffers from two weaknesses that lead to wastage of wavelinks and higher blocking ratios. Firstly, the CLMR algorithm, used in grooming algorithms with leaking strategy, requires more wavelinks when it builds light-trees, and secondly, leaky grooming wastes a large number of wavelinks. We also observe in Fig. 8 that when leaking is not allowed (i.e. the leaking ratio threshold is 0), MTLG and MTHG have their lowest blocking ratios. This is because if the leaking ratio is zero, the second weakness becomes irrelevant. However, since the first weakness is still relevant, LTD-ANCG performs better than either MTLG or MTHG. When traffic leaking is allowed (leaking ratio threshold is non-zero), the second weakness is valid once again, so that the blocking ratios of MTLG and MTHG are larger than that without leaking.

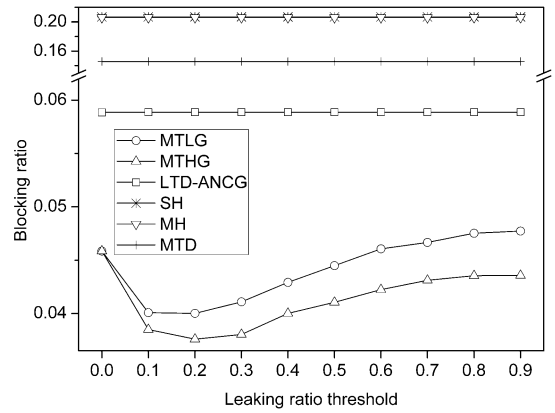

Fig. 6. Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.3.

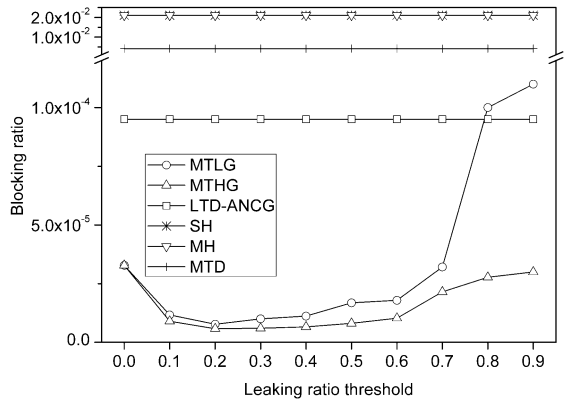

Fig. 7. Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.5.

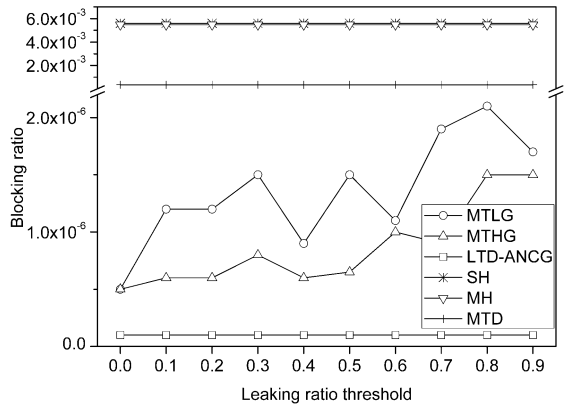

Fig. 8. Blocking ratio vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.9.

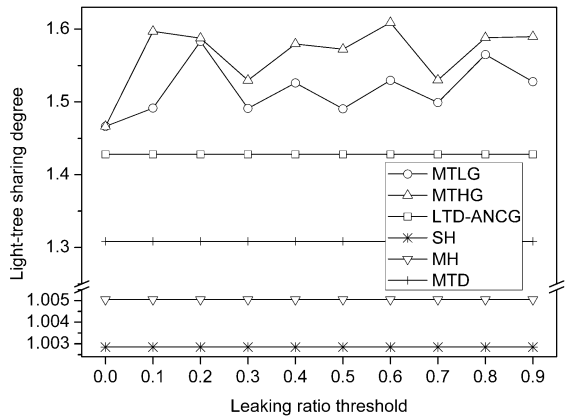

Fig. 9. Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.3.

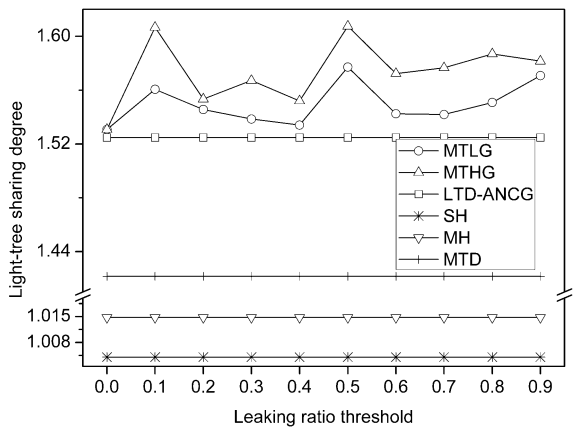

Fig. 10. Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.5.

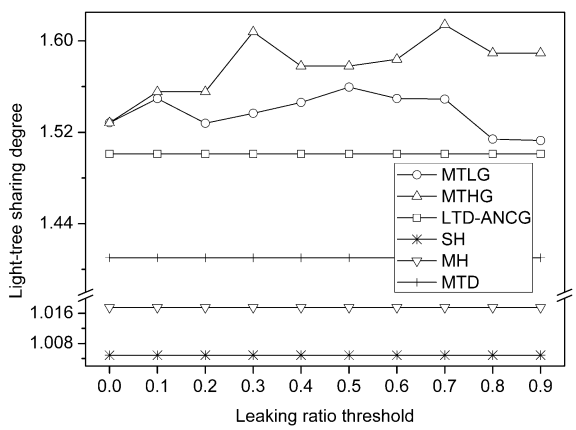

Fig. 11. Light-tree sharing degree vs. the leaking ratio threshold when network load is 50 erlangs and the add/drop ratio is 0.9.

We compare the six algorithms next in terms of a measure that we call *light-tree sharing degree* defined as follows,

$$\text{light-tree sharing degree} = \frac{\sum_{i=1}^{T} \text{num. of connections on tree } i}{T} \tag{11}$$

where T is the number of light-trees set up in the network. Figs. 9-11 show the average light-tree sharing degree vs. the leaking ratio threshold for three different values of the add/drop ratio. In these figures, each point is the average value of twenty sampling results over the simulation running time. Generally, the MTHG algorithm achieves the highest light-tree sharing degree, followed by MTLG, LTD-ANCG, MTD, MH and SH. We observe that the two proposed grooming algorithms with leaking strategy have higher light-tree sharing degrees than all other algorithms.

Fig. 12 shows the blocking ratio vs. the network load for the six algorithms when the add/drop ratio is 0.5 and the leaking ratio threshold is 0.2. In this scenario, the MTHG algorithm always performs the best, followed by MTLG, LTD-ANCG, MTD, SH and MH. Fig. 13 shows the relationship between the blocking ratio and the add/drop ratio for the six algorithms when network load is 50 erlangs and the leaking ratio threshold is 0.2. Here SH, MH and MTD still have higher blocking ratios than the other three algorithms because of their low grooming efficiencies. It is noted that, for this scenario, when add/drop ratio is less than 0.6, MTHG outperforms the others, but when add/drop ratio is 0.6, LTD-ANCG performs the best. This is because when add/drop ratio is 0.6, the add/drop port resource is sufficient so that LTD-ANCG uses wavelengths more efficiently than MTHG; here, traffic grooming with leaking wastes wavelengths which may lead to higher blocking ratios.

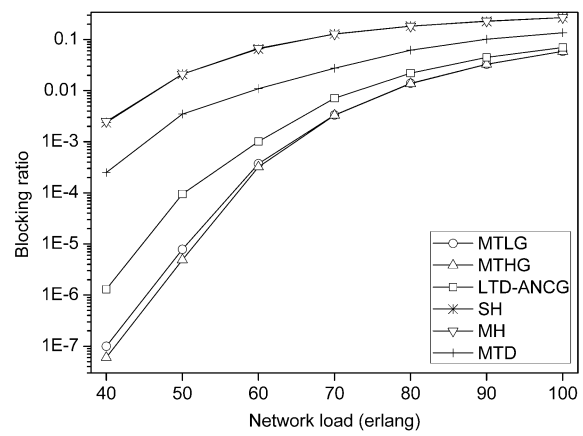

Fig. 12. Blocking ratio vs. the network load when the add/drop ratio is 0.5 and leaking ratio threshold is 0.2.

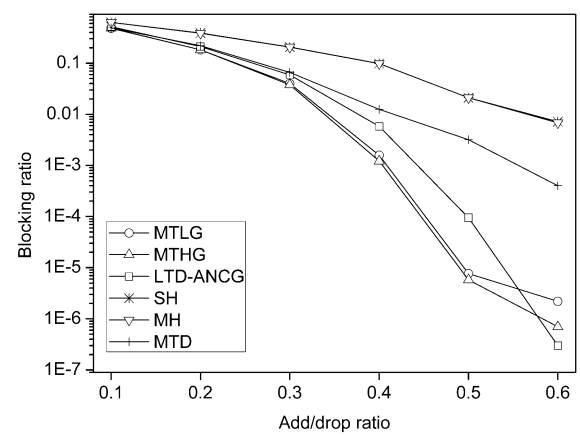

Fig. 13. Blocking ratio vs. the add/drop ratio when network load is 50 erlangs and the leaking ratio threshold is 0.2.

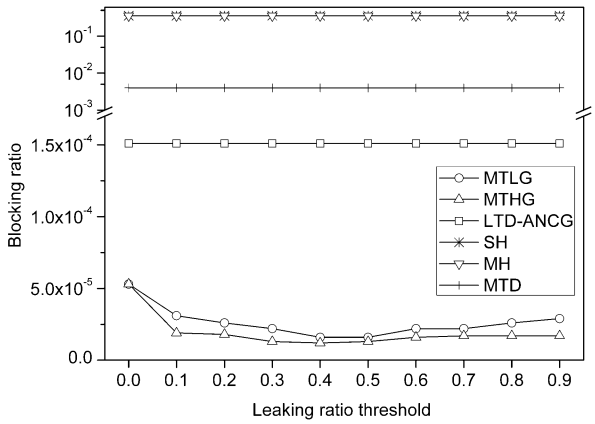

Fig. 14. Blocking ratio vs. the leaking ratio threshold when network load is 80 erlangs and the add/drop ratio is 0.3.

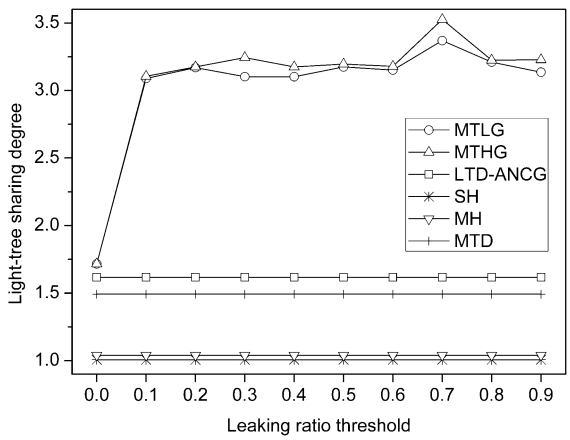

Fig. 15. Light-tree sharing degree vs. the leaking ratio threshold when network load is 80 erlangs and the add/drop ratio is 0.3.

We next consider a scenario, where connections with lower bandwidth have a larger portion, implying that the average connection rate requirement is further lower than wavelength capacity. Specifically, we assume that there are three different ranges of bandwidths (0, 0.2C], (0.2C, 0.7C] and (0.7C, C] in proportion of 10:2:1, respectively, and the bandwidths required in each range are uniformly distributed. We provide the simulation results under this bandwidth requirement model in Figs. 14 and 15, where the network load is set to 80 erlangs and the add/drop ratio is 0.3. Fig. 14 shows the blocking ratios for the six algorithms. It is clear that the two proposed leaking algorithms perform much better than the other four algorithms, and MTHG performs the best. Fig. 15 compares the light-tree sharing degree for the six algorithms. MTHG has the highest light-tree sharing degree, followed by MTLG, LTD-ANCG, MTD, MH and SH. We also note that the light-tree sharing degrees of MTLG and MTHG in Fig. 15 are much higher than those in Figs. 9-11 where the required bandwidth is uniformly distributed between (0, C]. This is because there are more lower bandwidth connections in the new traffic scenario, which allows more connections to share a light-tree whose capacity is of a full wavelength capacity C. We also observe that the difference between the newly proposed leaking algorithms and LTD-ANCG increases; this is because with smaller bandwidth requirements, leaking strategy can make more connections share one light-tree than LTD-ANCG, and hence can achieve a higher grooming gain.

Since our proposed multicast traffic grooming algorithms with leaking strategy use add/drop ports more efficiently, they can perform significantly better than other algorithms at low add/drop ratios, which is the practical situation in WDM networks. By adjusting the leaking ratio threshold, proper tradeoff between the add/drop port utilization and the blocking ratio can be achieved. We have tested other network topologies, like USnet and COST239. These give similar results which have not been shown here for the sake of brevity.

VI. Conclusion

As add/drop ports are more expensive than wavelengths in WDM networks, we have investigated the problem of multicast traffic grooming with leaking strategy to increase network resource utilization. Two multicast traffic grooming algorithms with leaking strategy, namely, multicast traffic leaky grooming (MTLG) and multicast traffic hybrid grooming (MTHG), have been proposed. The MTLG algorithm grooms traffic to light-trees if the traffic leaked is below a given threshold value. The MTHG algorithm grooms traffic to light-trees without leaking first; then if some destinations are left, it grooms traffic to light-trees with leaking. The MTHG algorithm is an improvement over MTLG as it attains higher light-tree sharing with less traffic leaked. Simulation results have demonstrated that both proposed algorithms perform significantly better than MTD, MH and SH, and better than LTD-ANCG for low add/drop ratios, while MTHG performs better than MTLG. The tradeoff between the add/drop port utilization and the blocking ratio can be suitably adjusted by adjusting the leaking ratio threshold.

References

[1] B. Haskell, A. Netravali, and A. Puri, Digital Video: An Introduction to Mpeg-2. Springer, 1996.

[2] R. Dutta and G. N. Rouskas, "Traffic grooming in WDM networks: Past and future," *IEEE Network*, vol. 16, no. 6, pp. 46–56, 2002.

[3] X. Zhang and C. Qiao, "On scheduling all-to-all personalized connections and cost-effective designs in WDM rings," *IEEE/ACM Trans. Networking*, vol. 7, no. 3, pp. 435–443, Jun. 1999.

[4] A. Chiu and E. Modiano, "Traffic grooming algorithms for reducing electronic multiplexing costs in WDM ring networks," *J. Lightwave. Technol.*, vol. 18, no. 1, pp. 2–12, Jan. 2000.

[5] E. Modiano and P. J. Lin, "Traffic grooming in WDM networks," *IEEE Commun. Mag.*, vol. 39, no. 7, pp. 124–129, 2001.

[6] L. H. Sahasrabuddhe and B. Mukherjee, "Light trees: optical multicasting for improved performance in wavelength routed networks," *IEEE Commun. Mag.*, vol. 37, pp. 67–73, 1999.

[7] R. Lin, W. D. Zhong, S. Bose, and M. Zukerman, "Dynamic sub-light-tree based traffic grooming for multicast in WDM networks," *GLOBECOM 2010*, Dec. 2010.

[8] R. Lin, W. D. Zhong, S. Bose, and M. Zukerman, "Light-tree configuration for multicast traffic grooming in WDM mesh networks," *Photonic Network Communications*, vol. 20, no. 2, pp. 151–164, 2010.

[9] L. Guo, J. Cao, H. Yu, and L. Li, "Path-based routing provisioning with mixed shared protection in WDM mesh networks," *J. Lightwave Technol.*, vol. 24, no. 3, pp. 1129–1141, Mar. 2006.

[10] A. Billah, B. Wang, and A. Awwal, "Multicast traffic grooming in WDM optical mesh networks, in *Proc. of GLOBECOM*, vol. 5, Dec. 2003, pp. 2755–2760.

[11] Y. Zhu, Y. Jin, W. Sun, W. Guo, W. Hu, W. Zhong, and M. Wu, "Multicast flow aggregation in IP over optical networks," *IEEE J. Select. Areas Commun*., vol. 25, no. 5, pp. 1011–1021, Jun. 2007.

[12] D. N. Yang and W. Liao, "Design of light-tree based logical topologies for multicast streams in wavelength routed optical networks," *INFOCOM 2003*, vol. 1, pp. 32–41, 2003.

[13] R. Lin, W. D. Zhong, S. Bose, and M. Zukerman, "Design of WDM Networks With Multicast Traffic Grooming," *J. Lightwave Technol.*, vol. 29, no. 16, pp. 2337–2349, 2011.

[14] A. Khalil, A. Hadjiantonis, C. M. Assi, A. Shami, G. Ellinas, and M. A. Ali, "Dynamic provisioning of low-speed unicast/multicast traffic demands in mesh-based WDM optical networks," *J. Lightwave Technol*., vol. 24, no. pp. 681–693, Feb. 2006.

[15] C. Lu, X. Nie, S. Wang, and L. Li, "Efficient dynamic multicast traffic grooming algorithm on WDM networks," in *Proc. SPIE*, 2005, pp. 602230.1–602230.10.

[16] L. Guo, X. Wang, J. Cao, W. Hou, and L. Pang, "Multicast grooming algorithm in waveband switching optical networks," *J. Lightwave Technol.*, vol. 28, no.19, pp. 2856–2864, Oct. 2010.

[17] X. Huang, F. Farahmand, and J. P. Jue, "Multicast traffic grooming in wavelength-routed WDM mesh networks using dynamically changing light-trees," *J. Lightwave Technol*., vol. 23, pp. 3178–3187, 2005.

[18] S. Lee., H. Yen, A. Chen, "Traffic grooming for IP multicast over WDM networks using light-path and light-tree schemes," in *Proc. of Internationl Conference on Networks (ICN)*, vol. 1, 2010, pp. 291–293.

[19] W. Hou, L. Guo, J. Cao, J. Wu, and L. Hao, "Green multicast grooming based on optical bypass technology," *Optical Fiber Technology*, no. 17, pp. 111–119, 2011.

[20] ILOG CPLEX, ILOG, Inc., Mountain View, CA[Online]. Available: http://www.ilog.com/products/cplex/.

[21] W. D. Zhong, X. Niu, B. Chen, S. Bose, "Performance Comparison of Overlay and Peer Models in IP/MPLS over Optical Networks," *Photonic Network Communications*, vol. 9, pp. 121–131, 2005.