Efficient Dynamic Traffic Grooming in Service-differentiated WDM Mesh Networks ⋆

Harsha V. Madhyastha [a], C. Siva Ram Murthy [b]

[a] *Dept. of Computer Science and Engineering, University of Washington, Seattle, WA 98195, USA*

[b] *Dept. of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai - 600036, India*

Abstract

In this paper, we address the issue of traffic grooming in arbitrary WDM mesh networks. We present a novel groomer architecture wherein a combination of grooming at two different granularities is utilized to make the setup cost-effective without compromising on efficiency. We put forward an efficient algorithm for dynamic routing and wavelength assignment of sub-wavelength connections. We also propose a means of rerouting connections dynamically to facilitate increase in the average call acceptance ratio. Rerouting in the dynamic scenario has never been considered before. The connections, which may have arbitrary bandwidth requirements, are considered to be service- differentiated based on the need for protection. We show the advantage of employing segmented backup in place of end-to-end backup as well as the benefits of the rerouting mechanism we propose through extensive simulation results.

Key words: WDM optical mesh network, Survivability, Service-differentiation, Traffic grooming, Grooming architecture, Rerouting, Segmented backup

1. Introduction

The introduction of Wavelength Division Multiplexed (WDM) networks as the backbone of the Internet has led to a tremendous increase in available bandwidth. Though each channel can now support traffic of the order of Gbps, individual connection requests are still in the Mbps range. This necessitates the strategic assignment of route and wavelength (RWA) for each connection. The RWA problem for sub-wavelength demands given the constraints on network resources, such as number of wavelengths and number of grooming ports at each node, is called the *traffic grooming* problem. The significance of this problem lies behind the fact that a wavelength has to be converted from optical to electronic domain at the source and destination of each connection it has been assigned to, and this conversion accounts for a sizable portion of the network cost.

The problem of traffic grooming has been extensively studied for WDM SONET rings [1–3], though not much work has been done in the case of arbitrary traffic [4]. However, there has been a

⋆ This work was supported by the Department of Science and Technology, New Delhi, India.

Email addresses: harsha@cs.washington.edu (Harsha V. Madhyastha), murthy@iitm.ac.in (C. Siva Ram Murthy).

recent surge in efforts to solve the traffic grooming problem for mesh networks. The traffic grooming problem can be considered in either the static or the dynamic setting. Dynamic grooming is the problem of routing and assigning wavelengths for a new demand, given the current state of the network, whereas in static grooming, the traffic demands are known apriori and all of them have to be assigned routes and wavelengths either to minimize required resources or maximize the throughput given the resource constraints. In [5], the problem of static grooming has been addressed with the objective of minimizing the number of transceivers required to support all the traffic while [6] tackles the same problem aiming to maximize the throughput given the constraints on resources. Dynamic grooming of connections which require provisioning of a backup is considered in [7]. In [8] too, dynamic grooming has been addressed but without the aspect of survivability.

In all of the above work, each groomer is assumed to have unlimited grooming capability (ability to switch traffic among streams), which is justified in [6] by the fact that switching is done in software in the MPLS/IP architecture. This assumption is not a practical one as routing each packet by examining its header involves a large overhead, which makes the setup incapable of handling the large bandwidth of an optical WDM link. Hence, full-scale grooming, *i.e.*, as much switching ability as required, at extremely fine granularities is not practically feasible. (Grooming at fine granularities involves switching streams which carry very low traffic while grooming at coarse granularities can only switch higher-rate traffic streams.) In the other node architecture considered - SONET/WDM - the switching cost is proportional to the square of the number of ports on it. Though using the SONET/WDM architecture will lead to lesser grooming equipment cost as the number of traffic streams it can groom is limited, the overall cost will be high as grooming can only be done at coarse granularities, which will lead to the need for a greater number of wavelengths. In short, though MPLS/IP is efficient, it is infeasible and not cost-effective because of high processing overhead and SONET/WDM, though feasible, is neither efficient nor cost-effective because of grooming at coarse levels which results in large number of wavelengths.

In this paper, we propose a new node architecture which does away with the shortcomings of the above two architectures and combines their advantages to achieve the right combination of feasibility, efficiency and cost- effectiveness. We do so by having groomers at multiple granularities at each node. The concept of using a multi-layer node architecture was also considered in [9]. However, it failed to identify the full potential of grooming at multiple levels. In the architecture used in [9], any add-drop traffic has to pass through the complete hierarchy from bottom to top. Due to this dependence between levels, switching cost benefits are obtained only at the intermediate nodes of lightpaths. Our node architecture ensures saving in switching cost at all nodes as the groomers at different layers are completely independent, which makes use of the true strength of grooming at multiple granularities.

The problem we address in this paper is that of dynamic grooming with the mixed groomer node architecture. We consider a service-differentiated scenario wherein each connection may or may not require a backup. The existing work on dynamic grooming [7] [8] assumes that a new connection has to be routed on existing lightpaths without rerouting or splitting them as that would cause traffic disruption. We propose a means of rerouting that does not disrupt traffic which helps to increase the average call acceptance ratio. To the best of our knowledge, rerouting of lightpaths in the scenario where connections arrive and leave dynamically has never been considered before in the literature. Also, for setting up the secondary path for each survivable connection, we employ the segmented backup mechanism [10] in place of the end-to-end backup approach followed in [7]. The rerouting method we propose when used in combination with the segmented backup approach leads to significant increase in average call acceptance ratio.

The rest of the paper is organized as follows. A detailed description of the mixed groomer node architecture is given in Section 2. The heuristic we propose for dynamic grooming is outlined in Section 3 and its working is made clear with an example in Section 4. The results of the simulations performed are given in Section 5 and we finally conclude in Section 6.

2. Node Architecture

Our proposed novel node architecture involves the use of two groomers - one at a coarse level and the other at a finer level of granularity, which we call the higher level and lower level groomer, respectively [11]. To make this setup practically feasible, unlike the MPLS/IP architecture, we work with the practical assumption that the number of ports on the lower level groomer is limited. Though limited, the capability to groom at finer levels helps in efficient grooming by reducing the number of required wavelengths compared to that possible with the higher level groomer alone. The additional cost of the lower level groomer is more than offset by the decrease in infrastructure cost due to fewer wavelengths. In addition to the coarse and fine granularity groomers, our node architecture also makes use of a mapper, which has negligible cost as it does no processing; it just multiplexes/demultiplexes the add/drop traffic assuming best possible packing of the lower level streams into the higher level streams. Its low cost is due to the fact that it does not perform any switching.

The mixed groomer node architecture we present is shown in Fig. 1. This architecture is a very generic one and can be used on any hierarchy of traffic streams, for example, OC-48/OC-12/OC-3 or STM-16/STM-4/STM-1 or STM-1/VC-3/VC-12. From now on, for the sake of convenience, we will refer to a wavelength as OC-48, a higher level stream as OC-12 and a lower level stream as OC-3. So, in the node architecture shown, the OC-48s that need to be groomed are converted from optical to electronic form by the Receiver Array (RX) and fed as input to the higher level groomer. The function performed by the higher level groomer is to switch OC-12s among the different OC-48s it receives as input. The OC-12 groomer also receives OC-12s which do not need to be groomed (because they might be completely packed with OC-3s setup between the same source-destination pair), padded up to OC-48s, as input from the mapper. The mapper can also be implemented such that every OC-12, in which all the OC-3s on it are between the same source-destination pair, can be directed from the mapper to the OC-12 groomer. However, doing so when the OC-12 is not completely packed entails higher implementation complexity (as detection of padded up OC-3s is required). Hence, in our proposal, we only require the mapper to redirect OC-12s completely packed with OC-3s between the same source-destination pair to the OC-12 groomer.

If there is also a need to switch OC-3s among the OC-12s, then the OC-12 groomer feeds the corresponding OC-12 streams as input to the OC-3 groomer. Also, among the OC-12s generated by the mapper from the add/drop traffic, the ones which are not completely packed are routed to the OC-3 groomer. The streams between the OC-12 and the OC-3 groomer are essentially OC-48s, but only the OC-3 groomer can index the OC-3s within each OC-48 and switch them if required. The mapper receives the local add/drop traffic as input in the form of OC-3s padded up to OC-12s, and tries to pack them into OC-12s optimally. It does this by taking groups of 4 (the *groom factor* in this case) OC-12s and mapping the single OC-3s on them onto one OC-12. This has very low processing overhead as the OC-3s can be statically mapped to respective OC-12s. Since the mapper receives OC-3s padded up to OC-12s as input, if some OC-12 is assigned just one OC-3, then that OC-12 can be directly padded up to a OC-48 by the mapper and sent to the OC-12 groomer, bypassing the OC-3 groomer. The outgoing traffic from the node is converted from electronic to optical domain by the Transmitter Array (TX).

Essentially, the mixed groomer architecture can be divided into two logical units - the multiplexing/demultiplexing section (mapper) and the switching section (OC-3 and OC-12 groomers). The add/drop traffic that goes in and out of the groomer is in the form wherein each OC-3 is on a distinct OC-12. The mapper performs the task of multiplexing the OC-3s which are on the OC-12s which constitute the add traffic. This multiplexing is carried out to ensure best possible packing, *i.e.*, the OC-3s on every 4 OC-12s are multiplexed into 1 OC-12. The drop traffic is also similarly

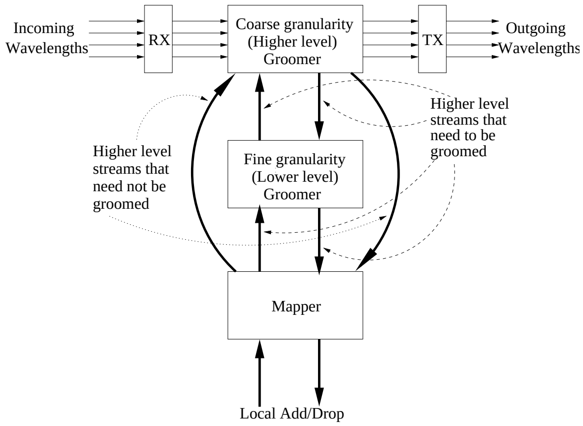


Fig. 1. Mixed-groomer Node Architecture

packed in the best possible manner. The mapper demultiplexes the OC-12s which constitute the drop traffic such that each of the OC-3s on these OC-12s is on a distinct OC-12. The task of switching traffic is completely handled by the OC-3 and OC-12 groomers. Since the mapper does not perform any switching, its cost is negligible in comparison with that of the groomers. The role of the OC-3 groomer is to switch OC-3s among OC-12s. Similarly, the function of the OC-12 groomer is to switch OC-12s among OC-48s. The number of switching ports taken up on the OC-3 groomer is the number of OC-12 streams it has to switch traffic amongst. Hence, from Fig. 1, it is clear that the number of OC-3 switching ports required is the sum of two quantities. The first being the number of OC-12s between it and the mapper. And, the second is the number of OC-12s between it and the OC-12 groomer. From the above explanation of how the mapper works, the number of OC-12 streams between the OC-3 groomer and the mapper is equal to $\lceil$ (Total add/drop traffic in terms of OC-3s)/(groom factor) $\rceil$. Similarly, the number of ports required on the OC-12 groomer is also the sum of two quantities. In this case, the first is the number of OC-48s fed as input to the groomer. The second is the number of OC-48s onto which it has to switch OC-12s, which are then fed as input to the OC-3 groomer. At the maximum, the value both these quantities take up is the number of OC-48s supported on the links incident at the node.

In our node architecture, the number of ports on the OC-3 groomer is constrained as this is a major contributor towards the cost of the setup. On the other hand, the number of ports on the OC-12 groomer can be assumed to be practically unlimited as grooming at a coarse level is comparatively inexpensive. Moreover, the number of ports required for full-scale grooming is lesser. To get an estimate of this, consider an OC-768 backbone, *i.e.*, each node in the network can handle bandwidth equivalent to OC-768. Since OC-768 is equal to 16 OC-48s, unlimited grooming capability at the OC-12 level would require 32 ports. This is because 16 ports would be required for the OC-48s on the link and another 16 for the add/drop traffic. On the other hand, since OC-768 is equivalent to 64 OC-12s, the number of ports required on the OC-3 groomer for input from the OC-12 groomer is 64. Also, 64 ports would be required for the OC-12s received from the mapper. This implies that a total of 128 ports are required on the OC-3 groomer. This quantity is 4 times as many as that on the OC-12 groomer. Since switching cost is proportional to the square of the number of ports, the switching cost at the OC-3 level is more than 16 times that at the OC-12 level [1] . On the whole, this clearly makes the cost of full-scale grooming at the OC-12 level negligible compared to that at the OC-3 level.

Let us now look at the advantages of the mixed groomer architecture over that of an OC-3 groomer or OC-12 groomer alone. If an OC-12 groomer alone is employed, it does not have the capability to switch OC-3s among OC-12s. So, the add/drop traffic in the form of OC-3s padded up to OC-12s cannot be multiplexed together. Each of these OC-12s will have to be assigned as they are to OC-48s on the link. Hence, as each OC-12 can only have one OC-3, the maximum traffic that can be supported is $\frac{1}{4}$, *i.e.*, 1/(groom factor) of the total bandwidth. On the other hand, using the OC-3 groomer alone suffers from two disadvantages. Firstly, due to the absence of the OC-12 groomer, if OC-12s among two OC-48s need to be swapped

[1] Though the number of ports on the OC-3 level is *exactly* 4 times that at the OC-12 level, we say switching cost is "more than" 16 times because the intrinsic cost of switching traffic streams increases as we go to finer levels of granularity.

(switching at the OC-12 level), this has to be done by swapping each of the OC-3s on these OC-12s. This is costlier as switching needs to be done at a finer granularity. Also, since there is no OC-12 groomer to pick out the specific OC-12s, all the OC-12s on these OC-48s will have to be fed as input to the OC-3 groomer. More importantly, the OC-3 groomer directly receives OC-3s padded up to OC-12s as input. This implies that the number of ports consumed due to the add/drop traffic is equal to the number of OC-3s in the add/drop traffic. Note that in the mixed groomer, this number was $(\frac{1}{4})^{th}$ of the add/drop traffic because multiplexing/demultiplexing is performed by the mapper. So, the mixed groomer architecture derives its efficiency by the combination of the OC-3 and OC-12 groomers and also, maintains practical feasibility and cost-effectiveness by the constraint on the number of ports on the OC-3 groomer.

2.1. *Example*

The following example clearly brings to the fore the advantages of using a combination of groomers in place of having an OC-3 or OC-12 groomer alone. Consider the 6-node network shown in Fig.2 with demands of 3, 1 and 4 OC-3s between the (source, destination) pairs *(1, 4)*, *(3, 5)* and *(2, 6)*, respectively. As outlined above switching cost at the OC-12 level is negligible to that at the OC-3 level which only depends on the number of OC-12s on the link. So, from here on, we consider OC-12 as a wavelength. The state of the network in each of the three cases explained below is as shown in Fig. 3.

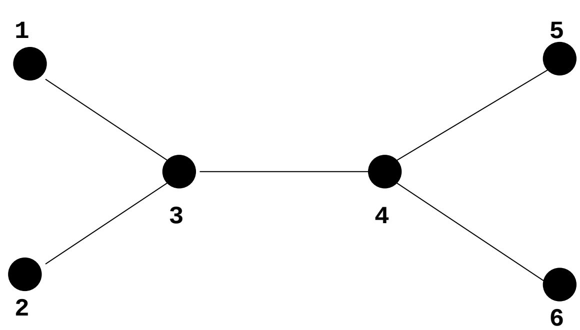


Fig. 2. Example 6-node Physical Topology

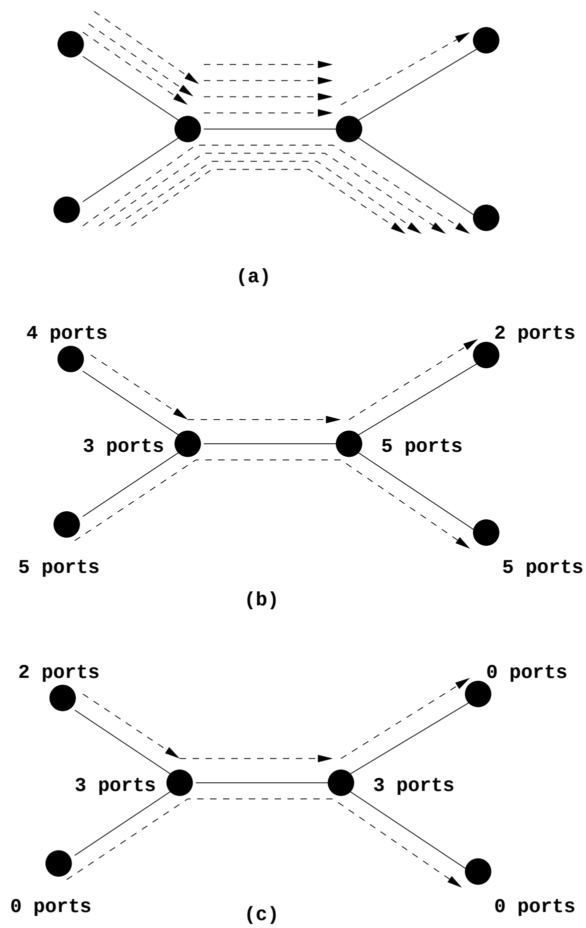


Fig. 3. Network state with (a) OC-12 groomer, (b) OC-3 groomer, (c) Mixed groomer

- OC-12 groomer alone
  When only an OC-12 groomer is available at each node, there is no grooming capability at the OC-3 level at any node. So, multiple OC-3s cannot be groomed onto the same OC-12, which implies that each OC-3 has to be carried on a new OC-12. This in turn implies that the number of wavelengths required on a link is equal to the total number of OC-3s transmitted along the link. As shown in Fig. 3(a), though there is no switching cost at any of the nodes, the overall network cost is high due to the large number of wavelengths required to satisfy the traffic demand. In this example, atleast 8 wavelengths are required on link *(3, 4)*.
- OC-3 groomer alone

If each node has an OC-3 groomer with full-scale grooming capability, then optimal grooming can be performed as shown in Fig. 3(b). But, the downside of this scheme is the high switching cost borne due to the large number of grooming ports required at each node, as shown in Fig. 3(b). As explained before, each add/drop OC-3 consumes a port on the OC-3 groomer and hence, the add/drop traffic itself consumes 3, 4, 1, 3, 1 and 4 ports at nodes 1, 2, 3, 4, 5 and 6, respectively. Also, at every node, among all the OC-12s on the links incident at that node, every OC-12 that needs to be groomed consumes an OC-3 grooming port at that node. An OC-12 needs to be groomed if some OC-3s on it need to be either dropped or switched to other OC-12s. All these properties together necessitate as many as 5 OC-3 grooming ports at nodes 2, 4 and 6. So, though the number of wavelengths required is reduced from 8 to 2 in comparison with the previous case, the grooming cost introduced keeps the network cost high.

– Mixed groomer architecture (OC-12 groomer + OC-3 groomer + Mapper)
The network state achieved with the mixed groomer node architecture (shown in Fig. 3(c)) clearly highlights its merits because as in the case with the OC-3 groomer alone, the number of wavelengths required is 2 but with much lower switching cost. The maximum number of ports needed at any of the nodes is 3 and three of the nodes do not even require an OC-3 groomer. The 4 OC-3s from node 2 to node 6 can be routed on the same OC-12 without consuming any OC-3 grooming ports as an OC-12 which is completely packed with OC-3s between the same (source, destination) pair directly goes from the mapper to the OC-12 groomer. Also, no ports are required for the OC-12 from node 4 to node 5 as a single OC-3 is put onto it. Lesser number of ports are also taken up at nodes 1, 3 and 4 because the mapper multiplexes/demultiplexes the add/drop traffic and hence, the number of ports consumed on the OC-3 groomer by the add/drop traffic is only $(\frac{1}{4})^{th}$ the number of add/drop OC-3s, which in this example translates into only one port at each of these nodes.
This example shows that our mixed groomer node architecture brings together the beneficial features of both a coarse granularity and a fine granularity groomer, *i.e.*, lower switching cost and lesser number of wavelengths required, respectively.

3. Problem Statement and Heuristic Solution

We consider the dynamic grooming problem in a service-differentiated scenario, wherein every newly arriving connection, may have an arbitrary traffic demand and also, each connection may or may not require protection. All connections are assumed to be bidirectional, *i.e.*, whenever some bandwidth is reserved on link (x, y) for some connection, equal amount of bandwidth has to be reserved for that connection on link (y, x) as well. Since the dynamic grooming problem is NP-hard, we present a heuristic algorithm to solve it. Whenever a request to setup a connection arrives, we need to determine a route for it and assign wavelengths to it on all links along this route. The secondary route for any connection is decided upon based on the single-link failure assumption. The routing and wavelength assignment for both the primary as well as the backup has to be carried out given the constraints on network resources. The resource constraints we consider are :

(i) The maximum number of distinct wavelengths on which traffic can be routed on each physical link - W_{max}. In our problem setting, this should ideally be the maximum number of OC-48s that can be carried on any link. However, since the grooming capability on the OC-12 groomer is practically unlimited, what matters is the number of ports consumed on the OC-3 groomer which depends on the number of OC-12s. Hence, we work with OC-12 as a wavelength and use W_{max} as the maximum number of OC-12s on each link. We assume the same W_{max} to hold over all physical links.
(ii) The number of ports on the OC-3 groomer at each node - P_{max}. This places a limit on the number of OC-12 streams that can

be groomed at each node, *i.e.*, the number of OC-12s which require OC-3s on them to be either added/dropped at that node or switched to other OC-12s.

(iii) The groom factor, *i.e.*, the ratio between the bandwidths of the higher and lower level streams - G. In our case where we consider OC-12 and OC-3, groom factor is 4.

The current network state, in terms of existing lightpaths and the traffic routed on each of them, has to be also taken into account while deciding the route and wavelength assignment for the new connection. In the existing literature [7] [8] on dynamic grooming, it has been assumed that no lightpath can be split or rerouted as that would lead to traffic disruption. We too stand by the policy that no lightpath can be split but we believe that dynamic rerouting is possible. Rerouting can be performed whenever a new connection cannot be established given the existing network state. A new connection cannot be established either if a route cannot be found for the primary or even if the primary can be established, if it requires a backup and a route cannot be determined for the secondary. Routes for both the primary and the secondary are to be found under the constraint that the existing lightpaths cannot be split but new lightpaths can be added given the spare resources available in the existing network state. The rerouting can be carried out by trying to reroute either the backup or the primary path of some connection in the following sequence where the next step is considered only if the new connection cannot be established by the rerouting in the preceding steps.

– Firstly, consider all connections which have been allotted a backup and whose secondary path is a end-to-end backup, not a segmented one. Consider these connections in some order and for each one, determine a new secondary which is link-disjoint with the current primary as well as the current backup. If such a route is found, check if rerouting the backup along the new secondary will facilitate the setting up of the new connection.
– Now, consider all connections for which a backup has been allocated, independent of whether it is end-to-end or segmented. Consider these connections in some order and for each one, determine the "best" route for the secondary in the current network state and check if the new connection can be satisfied by rerouting the backup along this route.
– Finally, consider all connections which either do not require protection or have a end-to-end backup. Consider these too in any order and try to determine a new route for the primary which is
  · link-disjoint with the current primary if it does not have a backup
  · link-disjoint with its secondary, if it exists, and not necessarily link-disjoint with the current primary

  Check if rerouting the primary along the new route will help establish the new connection.

At any stage in the above series of steps if it is determined that the new connection can be established by some rerouting, then that rerouting is performed and the new connection is setup. One point to be noted is that to establish a connection, not only must there exist a route on which the primary can be setup but there must also exist a secondary route if the connection requires protection. In our rerouting mechanism, the secondary of some connection is being rerouted in the first two steps and this does not disrupt the traffic. In the third step, two cases are considered. In the case wherein the route considered for rerouting does not require a backup, the new route must be link-disjoint with the current route. On the other hand, if it does have a backup, the new route need be link-disjoint only with its secondary, not necessarily link-disjoint with its current primary route. So, while the lightpaths which need to be established to route the traffic along the new primary route are setup, traffic can be routed either along the existing primary (in the case of no backup) or along the existing backup. Once the lightpaths are setup, traffic can be routed on the new primary as well as its link-disjoint route (the old primary or the secondary) for some amount of time and then the traffic flow along the link-disjoint route can be stopped.

The working of this is similar to that in Unidirectional Path Switched Ring (UPSR), wherein traffic is routed along both sides of the ring and the receiver chooses the data from that side on which higher power is detected. This process does not involve too much delay as overhead since the total time required to setup the new connection is still in the same order as the time required to establish a lightpath. In the case when a link can be expected to fail at the instant when a new connection arrives (the probability of which is almost negligible), the second step cannot be employed because the new secondary is not link-disjoint with the existing one. Whereas rerouting the secondary using the first step still ensures no disruption because rerouting is done along a path link-disjoint with the existing secondary and so, even if some link on the primary route fails while the new secondary is being setup, traffic can be routed along the existing secondary. Also note that rerouting in the third step ensures that the existing secondary works as the secondary even after rerouting, as it is link-disjoint with the new primary as well.

The routing mechanism employed is a minor modification of the fixed-alternate path routing. Instead of considering the k-shortest paths between two nodes, the set of paths which have lesser than k physical hops more than the shortest path are considered. This not only ensures fairness in setting up a connection independent of the shortest distance between the source and destination, but also helps in saving of network resources by avoiding the possibility of routing along extremely long paths. We determine the "best" combination of route and wavelength assignment as that which leads to the least increase in number of ports consumed. We call this the "least-port-increase" route. If there exists more than one such route, then the one with least number of physical hops is chosen. For provisioning of the secondary, we use the segmented backup approach with backup-backup multiplexing. The advantages of segmented backup are outlined in [10] while the algorithm to calculate the bandwidth to be reserved on any link when backup-backup multiplexing is employed is given in [7].

4. Illustrative Example

To provide a deeper insight into the heuristic outlined in the previous section, we consider an example scenario and step through the working of our heuristic. The physical topology considered is a 4×4 mesh with $G = 4$ and one wavelength available ($W_{max} = 1$). The connection requests (with their corresponding demands) arrive in the sequence *(7, 3)* (3 OC-3s with protection), *(13, 7)* (2 OC-3s), *(4, 8)* (4 OC-3s), *(1, 7)* (3 OC-3s with protection), *(1, 3)* (1 OC-3 with protection) and *(9, 7)* (1 OC-3). We now follow the changes in the network state (Fig. 4) as these connections arrive in the given order.

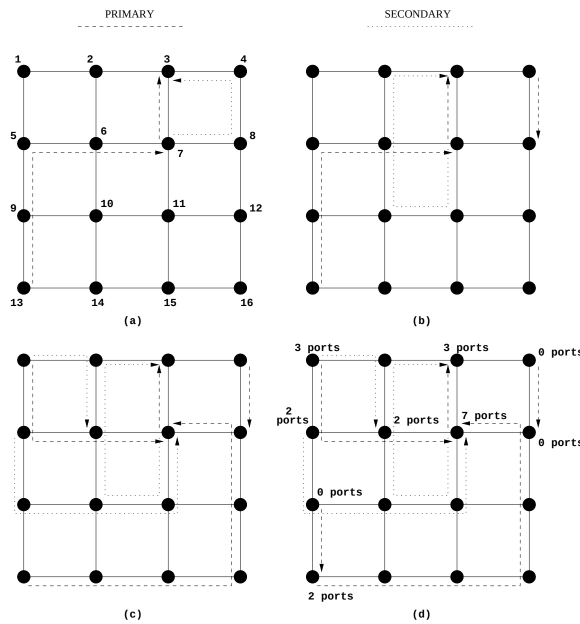


Fig. 4. Development of our heuristic

(a) As no connection currently exists, the primary for *(7, 3)* is assigned to the shortest path (which is a single link) and the secondary is chosen to be the next shortest path. The connection *(13, 7)*, which does not require backup, is also routed on the shortest path from node *13* to node *7*.
(b) The next connection to arrive is *(4, 8)* and given the current state of the network, this cannot be setup as 3 OC-3s have already been reserved on both the links adjacent with node *4* for the secondary of *(7, 3)*. This demand can be satisfied only if the backup path for *(7, 3)* is rerouted. Hence, the route for the secondary of *(7, 3)* is changed to the next

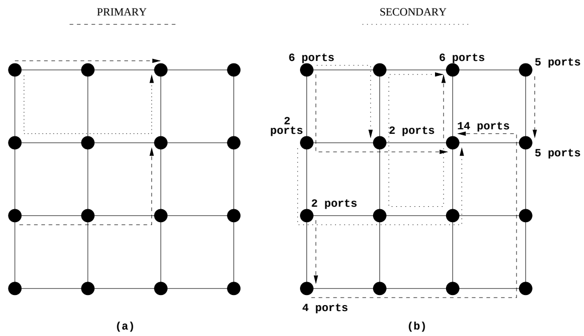


Fig. 5. Result with (a) coarse groomer (b) fine groomer

shortest path available from node *7* to node *3*. The connection *(4, 8)* can now be established on the link between the two nodes.

(c) Now, the request for connection *(1, 7)* arrives and it can be observed that no rerouting of the secondary for *(7, 3)* can facilitate the setting up of this new connection. Hence, we check whether rerouting the primary path for one of the connections will help in accommodating the new demand. Clearly, the routes assigned to *(7, 3)* and *(4, 8)* cannot be changed. That leaves us with connection *(13, 7)* and if it is rerouted along the alternative path available from node *13* to node *7*, the current request can be established. So, we perform the rerouting and setup the primary path for connection *(1, 7)*. Note that the secondary path for this connection can now be determined only if a segmented backup approach employing backup-backup multiplexing is used. The next request *(1, 3)*, which requires protection, can now be routed on the lightpaths $1 \rightarrow 7$ and $7 \rightarrow 3$ as both these have been allotted backups.

(d) Finally, connection request *(9, 7)* arrives and it is routed by setting up the lightpath $9 \rightarrow 13$ and then routing the traffic through the already established lightpath $13 \rightarrow 7$.

In contrast with the network state achieved by using our mixed groomer in combination with our heuristic (Fig. 4(d)), the network states resulting by using a coarse groomer or a fine groomer alone are shown in Fig. 5(a) and Fig. 5(b), respectively. While using the coarse groomer, only connections with 1 OC-3 demand can be setup as only one OC-12 is available and, in the absence of the mapper and the fine groomer, the coarse groomer cannot groom 2 OC-3s onto the same OC-12. In the case of the fine groomer, all the connection requests can be satisfied (if our heuristic is employed) but with significantly high network cost compared to that required with the mixed groomer. This can be attributed to two reasons. As explained in Section 2, the number of ports consumed by the add/drop traffic at any node is equal to the number of OC-3s added/dropped in the case of a fine groomer, whereas with the mixed groomer, only $(1/G)$ of these many ports are required because of the mapper. For example. at node *1*, the total add/drop traffic is 4 OC-3s which requires 4 ports on the fine groomer but just 1 port on the mixed groomer. As 2 OC-12s need to be groomed as well at node *1*, a total of 6 ports are required with the fine groomer while 3 ports are sufficient with the mixed groomer. Also, OC-12s which are either completely packed with OC-3s between the same node pair or have only one OC-3 assigned to them do not require to be groomed at the OC-3 level due to which the OC-3 groomer is not even required at some of the nodes (*4*, *8* and *9*) when the mixed groomer is used. Note that ports are consumed not only by the OC-12s on primary lightpaths but by those on the secondary lightpaths as well. On the whole, while a maximum of 7 grooming ports at the OC-3 level suffice when the mixed groomer is used, using the fine groomer requires as many as 14 ports. This example clearly demonstrates the advantages of our mixed groomer node architecture as well as the strengths of our heuristic. The performance of our heuristic is further highlighted in this example by the fact that all the connection requests could not have been satisfied by employing either rerouting (with end-to-end backup) or segmented backup in isolation.

5. Simulation Results

We conducted several simulations to not only demonstrate the efficiency of our heuristic but to also show its better performance compared to existing heuristics. It can be observed that without

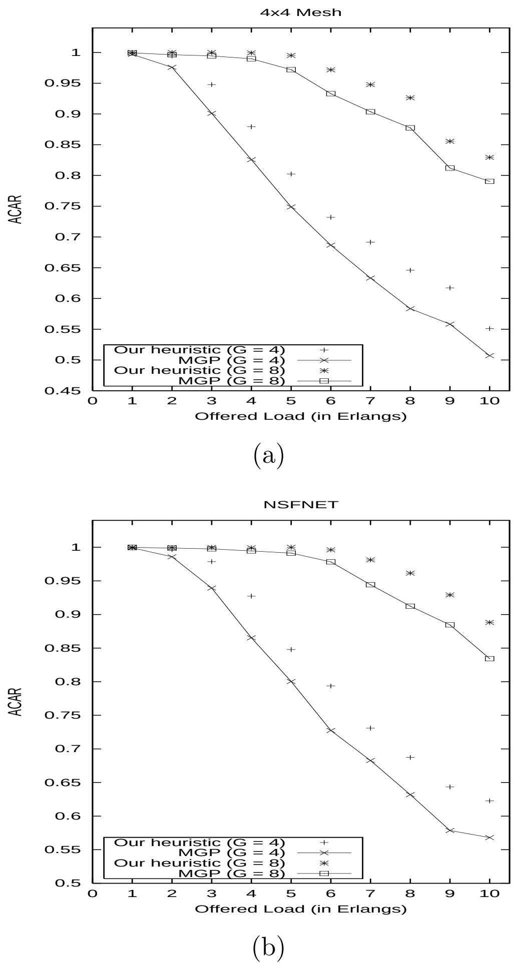


Fig. 6. Increasing ACAR with increasing groom factor ($P_{max} = 15$, $B = 0.5$)

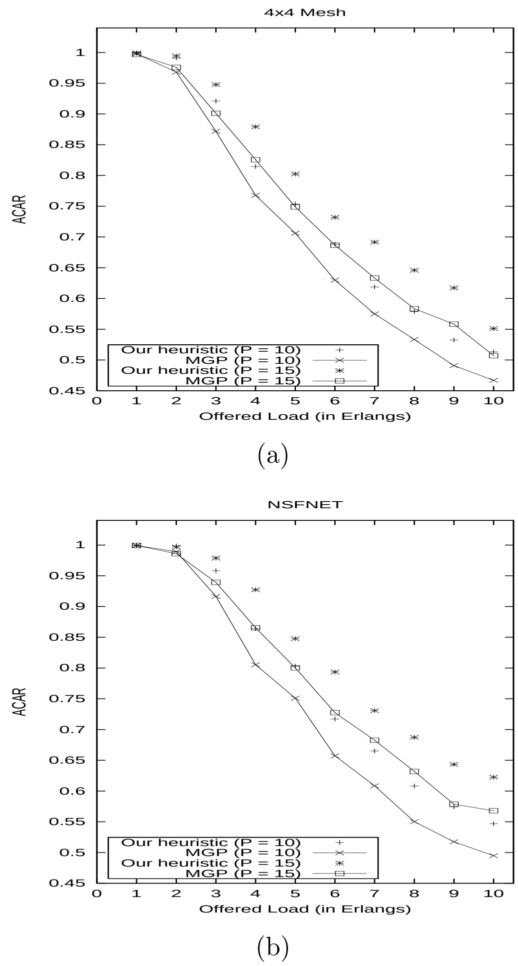
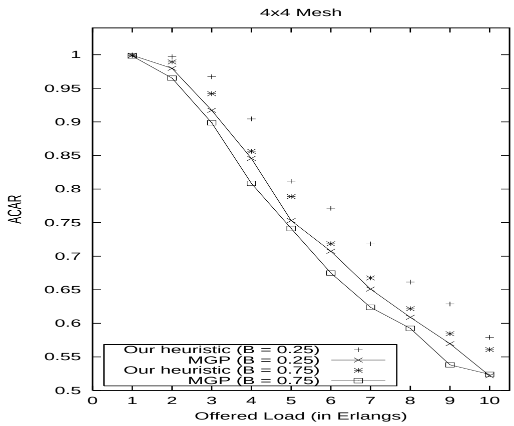


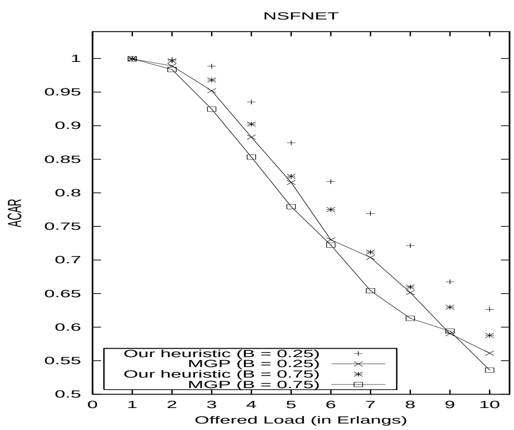
Fig. 7. Increasing ACAR with increasing number of ports ($G = 4$, $B = 0.5$)

using our rerouting mechanism and employing end-to-end backup instead of segmented backup, our heuristic reduces to the adaptation of the *Mixed Grooming Policy* (MGP) [7] to the scenario we are considering, *i.e.*, service-differentiated mesh with mixed groomer node architecture. Since there is no previous work in this scenario, we compare our heuristic with the modification of *MGP*. Moreover, we also compare the performance with the versions of our heuristic wherein only rerouting (with end-to-end backup) or only segmented backup is employed, to determine which of these features contributes more to the improved performance of our heuristic.

All our simulations were conducted on two physical topologies - a 4×4 mesh network and the NSFNET network - with the number of wavelengths (W_{max}) fixed at 5. The call generation mechanism followed is the same as that in [7], except for the introduction of a new parameter B, which is the fraction of the total number of calls which require protection. We used the Average Call Acceptance Ratio (ACAR) as the measure of performance. Firstly, we studied the increase in ACAR with increasing groom factor, the results of which are shown in Fig. 6(a) and Fig. 6(b). Fig. 7(a) and Fig. 7(b) display similar increase in ACAR with greater number of ports available. These results demonstrate the efficiency of our heuristic as the ACAR clearly increases with

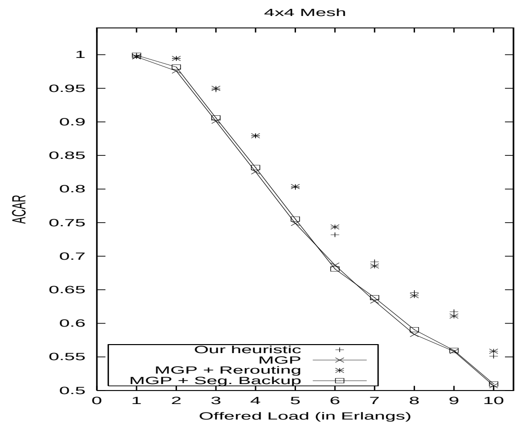


(a)

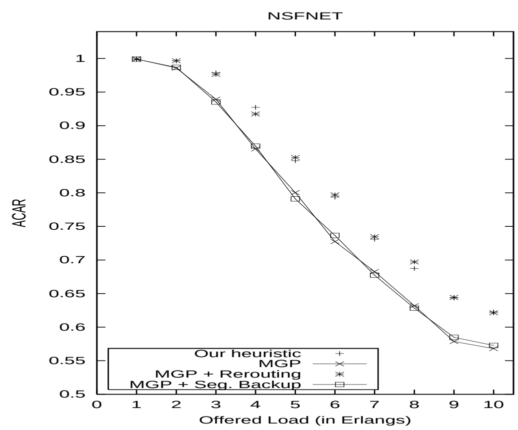


(b)

Fig. 8. Increasing ACAR with decreasing percentage of survivable traffic ($G = 4$, $P_{max} = 15$)



(a)



(b)

Fig. 9. Comparison of increased performance due to segmented backup and rerouting ($P_{max} = 15$, $B = 0.5$, $G = 4$)

increase in available network resources. This efficiency is derived from our policy of assigning a connection to its *"least-port-increase"* route and from the modified form of *k*- shortest paths we consider. The efficiency of our heuristic is further substantiated by Fig. 8(a) and Fig. 8(b), which show increase in ACAR with decrease in the fraction of demands requiring protection. In all these results, the ACAR obtained with our heuristic is consistently higher than that achieved by *MGP*.

The attributes in which *MGP* differs from our heuristic, the dynamic rerouting strategy we introduced and segmented backup in place of end-to-end backup, are clearly the factors behind this better performance. However, to determine which of these features contributes more towards the better performance, we compared the ACAR yielded by *MGP* and our heuristic with that obtained by employing only rerouting (with end-to-end backup) or only segmented backup. The results of these simulations, shown in Fig. 9(a) and Fig. 9(b), clearly indicate that the scheme of dynamic rerouting, never considered before in the literature, which we have proposed in this paper is the main cause for better performance.

However, these results do not bring out the improvement in performance obtained due to the employment of the mixed groomer node architecture instead of an OC-3 or OC-12 groomer alone. To

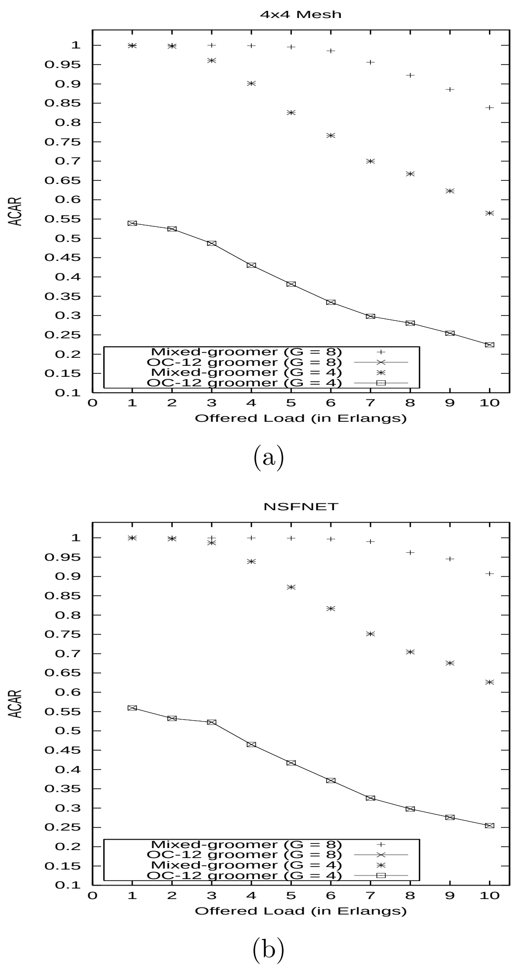


Fig. 10. Higher ACAR obtained by using mixed groomer instead of OC-12 groomer ($P_{max} = 15,\ B = 0.5$)

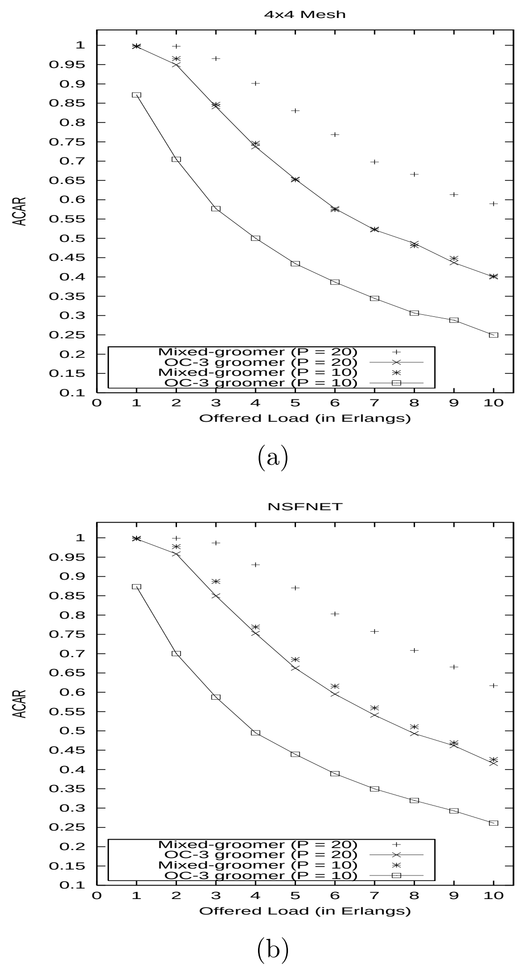


Fig. 11. Higher ACAR obtained by using mixed groomer instead of OC-3 groomer ($G = 4,\ B = 0.5$)

ascertain this, we conducted a couple more simulations, in which we compared the ACAR obtained with the mixed groomer with that obtained with an OC-12 or OC-3 groomer alone. The results of these simulations, in which the grooming heuristic we propose in this paper was employed, are shown in Fig. 10 and Fig. 11, respectively. As explained in Section 2, the OC-12 groomer cannot groom several OC-3s onto a single OC-12. This implies that an OC-12 groomer can support only $\frac{1}{G}$ of the available capacity, which results in the low ACAR as seen in Fig. 10(a) and Fig. 10(b). This also implies that the performance obtained with an OC-12 groomer is independent of G, which also shows up in these results. On the other hand, the reason for the low ACAR obtained when an OC-3 groomer alone is employed (Fig. 11(a) and Fig. 11(b)) is due to the absence of the mapper. In the mixed groomer, the mapper multiplexes/demultiplexes the add/drop traffic, and so, the number of ports consumed on the OC-3 groomer by the add/drop traffic is $\lceil$ (Total add/drop traffic in terms of OC-3s)$/G\ \rceil$. Whereas, when using an OC-3 groomer alone, every add/drop OC-3 consumes one port on the OC-3 groomer. With the limited number of grooming ports available, this translates into lower ACAR. Employing an OC-3 or OC-12 groomer alone, one could get the same performance as that obtained with the mixed-groomer, but at the cost of greater number of groomer ports required or greater number of wavelengths required, respectively.

On the whole, the results in this section substantiated the efficiency and improved performance of both our groomer architecture as well as our heuristic in comparison with existing groomer architectures and grooming heuristics.

6. Conclusion

In this paper, we presented a novel node architecture for traffic grooming in WDM optical networks. The use of groomers at multiple granularities in our mixed groomer node architecture makes it efficient while the constraint on number of ports on the lower-level groomer ensures its practical feasibility and cost-effectiveness. We addressed the problem of dynamic grooming with our node architecture in a service-differentiated scenario. We suggested a scheme for rerouting of connections dynamically and showed that this when used in combination with segmented backup employing backup-backup multiplexing helps to increase the average call acceptance ratio significantly. Extensive simulation results were provided to justify our claims.

Acknowledgments

The mixed-groomer node architecture was first conceived when the first author undertook a summer internship at Tejas Networks, Bangalore during the period May to July 2002. The authors thank Dr. Sarath Kumar and Dr. Kumar N. Sivarajan at Tejas Networks for their invaluable contributions through insightful discussions. This work was supported by the Department of Science and Technology, New Delhi, India. The authors wish to thank the anonymous reviewers for their comments and suggestions that helped improve the quality of this paper.

References

[1] E. Modiano, A. Chiu, Traffic grooming algorithms for reducing electronic multiplexing costs in wdm ring networks, IEEE/OSA Journal of Lightwave Technology 18 (1) (2000) 2–12.

[2] X. Zhang, C. Qiao, An effective and comprehensive approach to traffic grooming and wavelength assignment in sonet/wdm rings, IEEE/ACM Transactions on Networking 8 (5) (2000) 608–617.

[3] X. Zhang, C. Qiao, On scheduling all-to-all personalized connections and cost-effective designs in wdm rings, IEEE/ACM Transactions on Networking 7 (3) (1999) 435–445.

[4] P. J. Wan, G. Calinescu, L. Liu, O. Frieder, Grooming of arbitrary traffic in sonet/wdm rings, IEEE Journal on Selected Areas in Communications 18 (10) (2000) 1995–2003.

[5] V. R. Konda, T. Chow, Algorithm for traffic grooming in optical networks, in: Proc. IEEE HPSR'01, 2001, pp. 218–221.

[6] K. Zhu, B. Mukherjee, Traffic grooming in an optical wdm mesh network, IEEE Journal on Selected Areas in Communications 20 (1) (2002) 122–133.

[7] S. Thiagarajan, A. K. Somani, Traffic grooming for survivable wdm mesh networks, in: Proc. SPIE OptiComm'01, 2001, pp. 54–65.

[8] H. Zhu, H. Zang, K. Zhu, B. Mukherjee, Dynamic traffic grooming in wdm mesh networks using a novel graph model, in: Proc. IEEE Globecom'02, 2002.

[9] G. Hiuban, S. Pérennes, M. Syska, Traffic grooming in wdm networks with multi-layer-switches, in: Proc. IEEE ICC'02, 2002.

[10] C. V. Saradhi, C. S. R. Murthy, Dynamic establishment of segmented protection paths in single and multifiber wdm mesh networks, in: Proc. SPIE OptiComm'02, 2002, pp. 211–222.

[11] H. V. Madhyastha, C. S. R. Murthy, A practical and cost-effective approach to efficient traffic grooming in wdm mesh networks, Tech. rep., Department of Computer Science and Engineering, Indian Institute of Technology, Madras, India (Dec. 2002).