On-Line Routing in WDM-TDM Switched Optical Mesh Networks

Arun Vishwanath and Weifa Liang
Department of Computer Science
The Australian National University
Canberra, ACT-0200, Australia
Email: {arunv,wliang}@cs.anu.edu.au

Abstract

This paper considers the on-line traffic grooming problem in WDM-TDM switched optical mesh networks without wavelength conversion capability. The objective is to efficiently route connection requests with fractional wavelength capacity requirements onto high-capacity wavelengths and balance the load on the links in the network at the same time. To do so, we propose a cost function, which not only encourages grooming new connection requests onto the wavelengths that are being used by existing traffic, but also performs load balancing by intelligently increasing the cost of using wavelengths on links. The performance results obtained by experiments on a representative sized mesh network show that the proposed algorithm outperforms the existing algorithms.

1. Introduction

Wavelength Division Multiplexing (WDM) optical networks provide enormous bandwidth, and are promising candidates for information transmission in high-speed networks [1]. The bandwidth available on each wavelength in commercial WDM systems is in the order of OC-48/OC-192/OC-768 (2.48Gbps/10Gbps/40Gbps). However, as most applications require only sub-wavelength capacities, the available bandwidth on a single wavelength far exceeds the capacity requirement of a typical connection request. For example, HDTV works well with just *20Mbps*.

To overcome the disparity between the bandwidth required by connection requests and the available bandwidth on wavelengths, a technique called *traffic grooming* has been proposed. One approach to provisioning sub-wavelength capacity requests (traffic grooming) is to divide a wavelength into multiple time slots and multiplex different requests onto different time slots [2, 3]. Thus, upto p connection requests can share the same wavelength, assuming each request occupies one time slot and each wavelength is divided into p time slots. The resulting multiwavelength optical time division multiplexed network is referred to as a *WDM-TDM network* or a *WDM grooming network*. A connection between a source node and a destination node is setup by assigning time slots on every link in the routing path. Intermediate nodes along the path then switch time slots from one link to the next [4].

On-line traffic grooming in WDM-TDM switched networks was first investigated in [5]. The work examines the effect of wavelength conversion and time slot interchange on the performance of WDM-TDM networks. Their study concludes that, in networks with small number of wavelengths and large number of time slots per wavelength, significant performance gains can be achieved without the use of wavelength conversion but with the use of time slot interchange alone. In [6], the authors consider on-line traffic grooming in time division multiplexed WDM networks under the assumption that the nodes in the network do not have time slot interchangers. As a result, a connection request between a source node and a destination node must occupy the same time slot(s) along the links in the path. This can lead to a high blocking probability when compared to networks that incorporate time slot interchange functionality. The problem is solved by first partitioning it into three sub-problems: routing, wavelength assignment and time slot allocation. For every new incoming connection request, each of these sub-problems is then solved separately to determine the route, wavelength and time slot on which to route the request. In [2, 3], the authors consider the general problem of on-line routing in WDM-TDM switched networks with Optical Time Slot Interchangers (OTSIs), while in [4], the authors propose a generalized network model called the Trunk Switched Network (TSN) to facilitate the modeling and analysis of WDM-TDM switched networks. An analytical model is developed to evaluate the blocking performance of TSNs.

In this paper, we introduce a novel exponential cost function and propose an on-line routing algorithm for traffic grooming in WDM-TDM switched mesh networks. The algorithm integrates traffic grooming and load balancing with the aim of maximizing the network throughput. We show through experimental results that the proposed heuristic algorithm outperforms the other well-known algorithms discussed and analyzed in [2, 3] for traffic grooming in WDM-TDM switched networks.

The rest of the paper is organized as follows. The node architecture and the problem definition are introduced in Section 2. The proposed heuristic algorithm is presented in Section 3 and the simulation results are discussed and analyzed in Section 4. We conclude the paper in Section 5.

2. Preliminaries

In this section, we first introduce the node architecture used in our study and then formally define the on-line traffic grooming problem. We then provide a brief overview of the existing algorithms for the concerned problem.

2.1. Node Architecture

A WDM-TDM switched mesh network consists of switching nodes with communication fiber links interconnecting the nodes. Each fiber link carries a certain number of wavelengths and each wavelength is further divided into a number of time slots. The node architecture for sub-wavelength demand traffic grooming in such a WDM-TDM switched mesh network is shown in Fig. 1

The figure represents a node supporting three links (A, B, C), two wavelengths per link (λ_1, λ_2) and three time slots per wavelength (t_0, t_1, t_2). S_1, S_2, S_3, S_4, S_5, and S_6 are sessions utilizing 3, 1, 2, 2, 1, and 2 time slots of bandwidth respectively. Session S_1 occupying the full bandwidth on λ_1 is switched from input link A to the same output link. S_2 occupying time slot t_0 is switched to the same time slot from input link A to output link B on λ_2. Session S_3 arriving on λ_2 from input fiber link B is dropped locally

Proceedings of the Sixth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'05)
0-7695-2405-2/05 $20.00 © 2005 IEEE

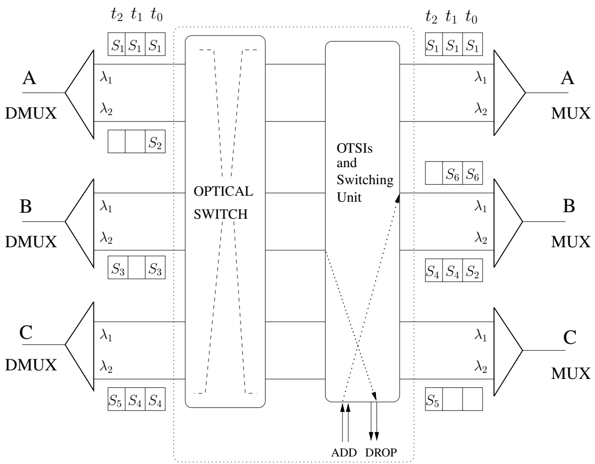

Figure 1. Node architecture

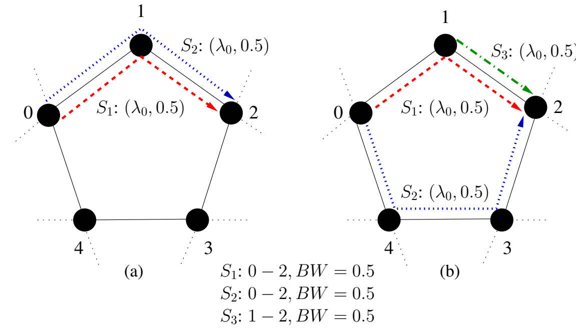

Figure 2. Example illustrating the ASP and the proposed routing schemes

at the node while session S_6 is added at the node and switched to output link B on λ_1. As there are three time slots per wavelength, the optical switch can be set in only three possible settings at any given time. When the switch is set to time slots t_0 and t_1, the signal S_4 occupying these time slots on λ_2 is switched from input link C to output link B. Signals can be delayed using Optical Time Slot Interchangers (OTSIs), therefore, time slots on an incoming signal can be mapped on to different time slots on the outgoing signal. Thus, before signal S_4 is sent on output link B, it undergoes a delay of one time slot duration so that time slots t_0 and t_1 on the incoming signal are mapped on to time slots t_1 and t_2 on the outgoing signal respectively. When the switch is set to time slot t_2, S_5 is switched on λ_2 to the same time slot on the output link C. As wavelength conversion is not incorporated in this architecture, the wavelength of an outgoing signal is the same as its incoming wavelength.

2.2. Problem Definition

The physical topology of a WDM-TDM switched mesh network can be represented by an undirected graph $G = (V, E)$, consisting of $|V| = n$ nodes and $|E| = m$ links interconnecting the nodes. Each link is bidirectional and is modeled as a pair of unidirectional links. $W = \{\lambda_1, \lambda_2, \ldots\ldots, \lambda_w\}$ is the set of available wavelengths in the network. A connection request i is represented by a quadruple $(s_i, d_i, \beta_i, \Delta_i)$, where $s_i \in V$ is the source node, $d_i \in V$ is the destination node, β_i is the required bandwidth and Δ_i is the duration of the request.

Given the current network state (routes and wavelengths being used by existing traffic), the *on-line traffic grooming problem* is to construct a minimum cost bandwidth guaranteed path P_i^λ on wavelength $\lambda \in W$ that connects the source node s_i to the destination node d_i. The aim is to maximize the network throughput. We assume that the established requests cannot be interrupted. The connection requests arrive one after the other and the arrival sequence is not known in advance.

2.3. Routing Algorithms

Based on the information used for establishing a path between the source node and the destination node [7], on-line routing algorithms can be classified into two types, (i) destination-specific routing algorithms, and (ii) request-specific routing algorithms. Destination-specific routing algorithms try to establish the best possible routing path between the source node and the destination node *without* any knowledge of the incoming connection request. On the other hand, request-specific routing algorithms aim to establish the best possible routing path between the two endpoints taking into account the bandwidth requirement of the incoming connection request. In [2, 3], the authors propose and study a new request-specific routing algorithm called *Available Shortest Path* (ASP) and compare its performance with two other destination-specific routing algorithms - *Widest Shortest Path* (WSP) and *Shortest Widest Path* (SWP). Their results indicate the importance of using request-specific routing algorithms for improving the performance of WDM-TDM switched networks. ASP outperforms the two destination-specific routing algorithms not just in terms of blocking probability but also with respect to other metrics such as fairness and utilization. For completeness, we briefly outline these algorithms below. It must be noted that, since wavelength conversion is not allowed, the routing algorithms are iteratively executed for each $\lambda \in W$ to determine the best possible path and wavelength on which to route the request.

(i) Widest Shortest Path (WSP). Dijkstra's algorithm is used to find the widest path between the source node and the destination node. If two or more paths are the same with respect to this metric, the path with the minimum hop count is selected. If two paths have the same hop count, then the tie is broken by choosing the path corresponding to the first-fit wavelength assignment policy.

(ii) Shortest Widest Path (SWP). This is similar to the conventional shortest path routing based on the hop count. If the hop count of two or more paths are the same, then the widest one among them is chosen. In case of a tie, the path corresponding to the first-fit wavelength assignment policy is selected.

(iii) Available Shortest Path (ASP). In this approach, *only* links with sufficient bandwidth capacity to accommodate the request are considered for route computation. Dijkstra's algorithm is then used to determine the shortest path between the source node and the destination node. If two or more paths can accommodate the request, then the path with the minimum hop count is chosen. If there is a tie, then the tie is broken by using the first-fit wavelength assignment policy.

To illustrate how heuristic ASP works, we consider the example shown in Fig. 2(a), which represents a five node subnet - 0, 1, 2, 3, 4, of a large mesh network. Assume that the current network configuration of this subnet is as follows. One wavelength λ_0 is available on links $0-1$ and $1-2$, and two wavelengths λ_0, λ_1 are available on links $0-4$, $4-3$ and $3-2$ respectively. We further assume that the total capacity available on a wavelength is 1, with sessions requiring fractional wavelength capacities. The order of sessions arriving are: S_1: $0 \rightarrow 2$, S_2: $0 \rightarrow 2$ and S_3: $1 \rightarrow 2$. Each session requests bandwidth equivalent to half of the wavelength capacity. The values within the parenthesis along the links in the figure indicate the wavelength and the amount of bandwidth used by the sessions.

S_1 is routed along the links $0-1-2$ as shown in Fig. 2 (a) on λ_0. To facilitate full-duplex communication, the bandwidth requested by sessions is reserved along the links in either direction.

Thus, following the establishment of S_1, S_2 is is also routed along the links $0-1-2$ on λ_0. As a result, node 1 is *logically disconnected* from the network along with the links $0-1$ and $1-2$, since no further wavelengths are available on these links while S_1 and S_2 continue to remain active. By logical disconnection we mean, the links incident to a node *cannot* be used by future connection requests as they do not have any more wavelengths available. In provisioning high speed connections, traffic requests are expected to have long holding times. Thus, logical disconnection of nodes result in fewer routing paths for each subsequent connection request, increasing the number of blocked requests, and in turn leading to significant loss of revenue. Consequently, the new session S_3 is blocked due to the lack of available wavelengths to route it.

3. On-Line Traffic Grooming Algorithm

We propose an On-Line Traffic Grooming Algorithm (OTGA), which (i) encourages grooming new sub-wavelength connection requests onto the wavelengths that are being used by existing traffic, and (ii) incorporates load balancing functionality simultaneously. To do so, we introduce a cost function that takes into consideration the *total load on a link* and the *residual available bandwidth on each wavelength*. For every new incoming connection request, Dijkstra's algorithm is used to establish the routing path between the source node and the destination node. As each connection request can only be routed on a single wavelength, at most w shortest paths can be generated, where w is the total number of wavelengths available in the network. Out of all the resulting wavelengths that can be used to establish the request, the wavelength corresponding to the least cost routing path is selected. Here, the cost of establishing the request is the sum of the cost of all the wavelength-links in the routing path. If two paths have the same cost, then the first-fit wavelength assignment policy is employed to break the tie. The bandwidth required by the connection request is then reserved along the links in the path. In the following, we use an example (see Fig. 2(b)) to explain the idea behind the proposed algorithm.

As in the ASP scheme, S_1 is routed along the links $0-1-2$ as shown in Fig. 2(b) on λ_0. This increases the load on the links $0-1$ and $1-2$. Since links $0-4$, $4-3$, and $3-2$ have more wavelengths, it is desirable to use these links to route future connection requests and prevent the depletion of wavelengths on links $0-1$ and $1-2$. Therefore, higher costs are assigned to λ_0 on links $0-1$ and $1-2$ for subsequent connection requests. As a result, S_2 is routed on λ_0 along the shortest path $0-4-3-2$. Now S_3 can be successfully established on λ_0 along the direct link $1-2$. Therefore, unlike in the Available Shortest Path routing strategy, S_3 is not blocked in the proposed routing scheme. Note that if links $0-1$ and $1-2$ had two wavelengths instead of one, then it might be desirable to groom connection S_2 along with the already established connection S_1.

We define *distance* as the minimum number of hops needed by *any* routing algorithm to route a connection request between the source node and the destination node. In other words, distance is the number of hops in the *shortest path* between the two endpoints in $G=(V,E)$ *without* considering the availability of wavelengths on links. The shortest path between nodes 0 and 2 in the mesh network (Fig. 2) consists of only 2 hops. Therefore, the distance between nodes 0 and 2 is 2. The number of hops used by OTGA to establish connection request S_2 between nodes 0 and 2 is 3 (see Fig. 2(b)). From these two hop counts we note that, in some cases, the number of hops needed by OTGA, and hence the amount of wavelength resources used by it is greater than the corresponding resources needed by the ASP routing scheme. To minimize the utilization of additional wavelength resources, we introduce the following connection admission policy. Let D_i be the distance (computed a priori) between the nodes s_i and d_i, and ε be the *additional* number of hops OTGA can take to establish the connection request between the nodes s_i and d_i. This implies that, even if sufficient bandwidth is available on wavelength $\lambda \in W$ to route request i, the request is *blocked* if the total number of hops in the resulting routing path is greater than $(D_i+\varepsilon)$. Note that ε is independent of the two endpoints of the connection request and the associated bandwidth requirement. Instead, it is an experimental parameter that is tuned depending on the physical topology of the network.

3.1. Cost Function

The cost function used in the algorithm is described as follows. We denote Ω as the total available bandwidth per wavelength. Let $\mu_{u,v}$ represent the total available bandwidth on a link between its two end-points u and v. Therefore, we have

$$\forall (u,v) \in E: \qquad \mu_{u,v} = w \times \Omega. \tag{1}$$

For convenience, we normalize the requested bandwidth to the total available bandwidth on a link. Therefore,

$$\hat{\beta}_i(u,v) = \frac{\beta_i}{\mu_{u,v}}. \tag{2}$$

Let $\mathcal{P} = \{P_1, P_2, P_3, \cdots\cdots\cdots, P_k\}$ be the set of routing paths assigned to connection requests 1 through k. If a request j is rejected, or terminates before the arrival of a new request, then $P_j = \emptyset$, where P_j is the routing path for request j $(1 \le j \le k)$. Therefore, the load on link $(u,v) \in E$ after considering request k is defined as

$$l^k_{u,v} = \sum_{\substack{j=1 \\ (u,v)\in P_j}}^{k} \hat{\beta}_j(u,v). \tag{3}$$

In WDM-TDM switched networks, bandwidth requirements of connection requests are expressed in terms of the number of time slots. In this work, we assume that each wavelength is sub-divided into 16 time slots and the capacity of each time slot is equivalent to 1 OC-3 channel. Therefore, the total capacity of each wavelength is equivalent to 1 OC-48 channel, and we have $\Omega = 16$ OC-3s.

Let $\tau_{j,\lambda'(u,v)}$ be the number of OC-3 channels being used on link (u,v) by request j on wavelength $\lambda' \in W$. Then, after considering request k, the total number of λ' OC-3 channels being used on link (u,v) is

$$U^{k,\lambda'}_{u,v} = \sum_{\substack{j=1 \\ (u,v)\in P_j}}^{k} \tau_{j,\lambda'(u,v)}. \tag{4}$$

When a new connection request i arrives, we assign costs to each wavelength $\lambda' \in W$ on the links in E as follows.

(i) If the capacity of λ' available on link (u,v) is equal to Ω, i.e. λ' is not being used by any existing connection request, then the cost of using λ' on it is

$$\Psi^{\lambda'}_{u,v} = a^{l^k_{u,v}}\left(a^{\hat{\beta}_i(u,v)} - 1\right). \tag{5}$$

(ii) Otherwise, λ' is currently being used by the existing traffic, and two cases arise.

Case 1. If the residual capacity of λ' on link (u,v) is less than Ω, but no less than the requested bandwidth β_i, then the cost of using $\lambda^{'}$ on it is

$$\Psi_{u,v}^{\lambda^{'}} = \frac{a^{l_{u,v}^k}\left(a^{\hat{\beta}_i(u,v)} - 1\right)}{b^{R_{\lambda'}(u,v)}}, \tag{6}$$

where a and b are appropriately chosen constants. Here, $R_{\lambda'}(u,v)$ is the residual capacity of $\lambda^{'}$ on link (u,v) after considering the first k requests, and is given by

$$R_{\lambda'}(u,v) = 1 - \frac{U_{u,v}^{k,\lambda^{'}}}{\Omega}. \tag{7}$$

Importantly, observe that, to realize load balancing and grooming interests, the constants a and b in the cost function must be greater than 1.

Case 2. If the residual capacity of $\lambda^{'}$ on link (u,v) is less than β_i, then $\Psi_{u,v}^{\lambda^{'}} = \infty$, which means it cannot be used to establish the routing path.

Note that, from Equation (5), if $\lambda^{'}$ is not being used by any existing connection request on link (u,v) (i.e. full wavelength capacity of $\lambda^{'}$ is available on link (u,v)), then the cost assigned to it represents the change in its relative load that would occur if it were to be used by the new connection request [8]. From Equation (6), if the residual capacity of $\lambda^{'}$ on link (u,v) is less than Ω, but no less than the requested bandwidth β_i, then the cost of $\lambda^{'}$ on this link is expressed as a function of the change in its relative load and the residual capacity of $\lambda^{'}$. In other words, load balancing is realized by increasing the cost of using wavelengths on heavily loaded links, thus discouraging them from being used by new connection requests. To encourage grooming new connection requests onto the *wavelengths that are already being used* by existing traffic, the costs of these wavelengths are *further decreased* by a factor of their residual capacities. Therefore, among the wavelengths that are currently being used on a link, we encourage grooming on the wavelength that has the highest residual capacity. This minimizes the logical disconnection of nodes from the network and achieves our objective.

3.2. Algorithm

We are now ready to introduce the detailed algorithm as follows. Once a new connection request arrives, the algorithm is executed to determine whether the request should be accepted.

Algorithm OTGA$(s_i, d_i, \beta_i, D_i, \varepsilon)$
begin
$\mathbb{C}_{\text{MAX}} \leftarrow \infty, \lambda \leftarrow nil, P_i^{\lambda} \leftarrow nil,$
/* $\mathbb{C}_{\text{MAX}}$ is the total cost to establish request i, λ is the resulting */
/* wavelength on which to route request i, P_i^{λ} is the routing */
/* path for request i, Ω is the total capacity per wavelength, */
/* Num_Hops (P_i^{λ}) returns the number of hops in P_i^{λ} */

Step 1. Tear down and free the wavelength resources used by all the connection requests that terminate before the arrival of connection request i.
$\forall (u,v) \in E: \ \hat{\beta}_i(u,v) \leftarrow \frac{\beta_i}{\mu_{u,v}}$
Step 2. for each wavelength $\lambda^{'} \in W$ **do**
Step 3. Compute $RC(u,v,\lambda^{'})$, the residual capacity of $\lambda^{'}$ on link $(u,v) \in E$
Step 4. **if** $RC(u,v,\lambda^{'}) = \Omega$
then $\Psi_{u,v}^{\lambda^{'}} \leftarrow$ cost from Equation (5)
else if $\beta_i \le RC(u,v,\lambda^{'}) < \Omega$
then $\Psi_{u,v}^{\lambda^{'}} \leftarrow$ cost from Equation (6)
else $\Psi_{u,v}^{\lambda^{'}} \leftarrow \infty$
endif;
endif;
Step 5. Using Dijkstra's algorithm, find a shortest path $P_{(\lambda^{'},i)}$ from s_i to d_i w.r.t costs $\Psi_{u,v}^{\lambda^{'}}$. Let c_i be the sum of the cost of all the links in $P_{(\lambda^{'},i)}$.
Step 6. **if** $c_i < \mathbb{C}_{\text{MAX}}$ **then**
$\mathbb{C}_{\text{MAX}} \leftarrow c_i; \lambda \leftarrow \lambda^{'}; P_i^{\lambda} \leftarrow P_{(\lambda^{'},i)}$
endif;
endfor;
Step 7. if $\mathbb{C}_{\text{MAX}} \neq \infty$ **then**
if Num_Hops$(P_i^{\lambda}) \le (D_i + \varepsilon)$ **then**
for each link $(u,v) \in P_i^{\lambda}$ **do**
$RC(u,v,\lambda) \leftarrow RC(u,v,\lambda) - \beta_i$
$RC(v,u,\lambda) \leftarrow RC(v,u,\lambda) - \beta_i$
endfor;
return P_i^{λ}
endif;
endif;
Step 8. return "request blocked"
end.

The computational complexity of the proposed algorithm can be analyzed as follows. The shortest path from the source node to the destination node can be found using Dijkstra's algorithm, which takes $O(m + n\log n)$ time using Fibonacci heaps. Therefore, the time complexity of the algorithm is $O(w(m + n\log n))$ as it is run once for each $\lambda^{'} \in W$ with $w = |W|$.

4. Simulation Study

To evaluate the performance of the proposed algorithm, we conducted experiments on a representative sized mesh network shown in Fig. 3, which consists of 24 nodes and 43 fiber links. Each fiber link carries 16 wavelengths. All the nodes in the network have the architecture shown in Fig. 1. We further assume that the wavelength continuity constraint is imposed.

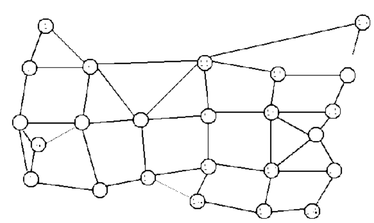

Figure 3. A 24 node telecom network

The bandwidth required by connection requests is uniformly distributed between 1 OC-3 and 16 OC-3s. The request arrival follows a Poisson process with the traffic uniformly distributed between all node pairs. The connection holding time is exponentially distributed. The load (in Erlangs) on the network is varied by increasing the average connection holding time. We simulate 200,000 connection requests to obtain the network performance under a certain network load. The simulations were performed on a Linux PC with a 2.8 GHz Pentium IV processor and 512 MB of memory. The average running time to simulate 200,000 connection requests is about 20 minutes. In all our experiments, the constants a, b and ε are fixed at 4, 2 and 2 respectively. We also experimented with other set of values and found the above combination to give consistently good network performance across all the loads.

4.1. Experimental Results

We compared the performance of OTGA with the existing algorithms - WSP, SWP and ASP. The metrics used to measure the

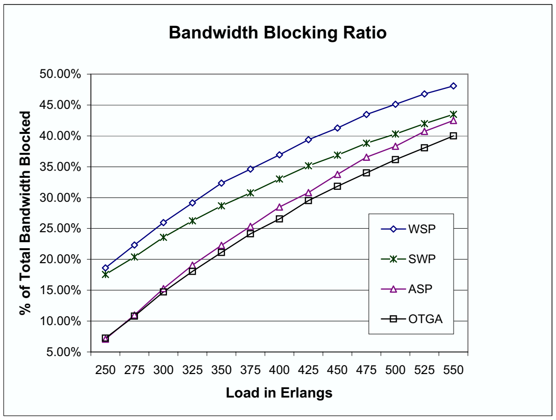

Figure 4. Bandwidth blocking ratio

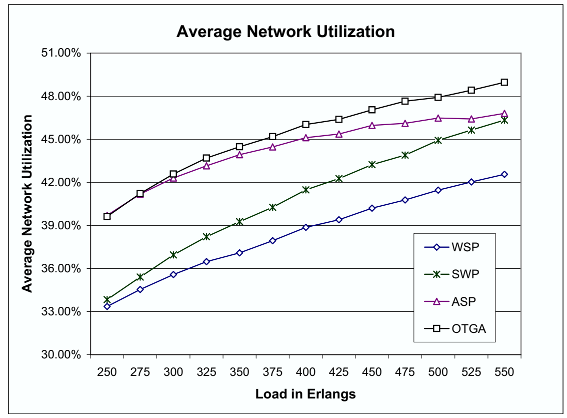

Figure 5. Average network utilization

performance of the algorithms are (i) the bandwidth blocking ratio, and (ii) the average network utilization.

(i) Bandwidth blocking ratio. Fig. 4 compares the bandwidth blocking ratio of different routing algorithms. It represents the percentage of the amount of blocked traffic over the total amount of bandwidth required by all the connection requests during the entire simulation period. As bandwidth requirements of different connection requests are different, just comparing the overall request blocking probability does not reflect the effectiveness of the routing algorithms. Instead, bandwidth blocking ratio is a more suitable metric to compare the network performance and throughput. It can be observed from the figure that, at low network loads (in Erlangs), the percentage of bandwidth blocked by OTGA and ASP is similar. This is because, at low loads, the average connection holding time is less. The costs assigned to all the links derived from the cost function are nearly identical. Therefore, the performance of OTGA and ASP are similar. With the increase in the average connection holding time, the network load also increases. The exponential nature of the cost functions in Equations (5) and (6) prevent the depletion of wavelengths on heavily loaded links by assigning to it, costs, that are significantly higher than the costs assigned to lightly loaded links. This in turn leads to the creation of routing paths that are distributed among the links evenly. From the figure it can be seen that, as the network load increases, the bandwidth blocking ratio increases as well. However, the percentage of total bandwidth blocked by OTGA is lower than that of the other three heuristics. OTGA delivers higher network throughput, and thus offers better performance.

(ii) Average network utilization. The average network utilization is determined as follows. Consider a connection request i between nodes s_i and d_i with the capacity requirement β_i. Let the distance between them be D_i. Now, if connection request i is to be established, then irrespective of the routing algorithm used, the minimum capacity required in the network is $\beta_i \times D_i$. This is called the *effective capacity requirement* of the request. Depending on the routing algorithm employed, the number of hops taken by it to establish the connection request may be greater than D_i. Denote by ENC, the effective network capacity utilized at any instant of time. ENC is defined as the sum of the effective capacity requirement of all the connection requests that are active at that instant. The total network capacity is defined as $m \times |W| \times \Omega$. The network utilization is then determined as the ratio of the effective network capacity utilized to the total network capacity as $\frac{ENC}{m \times |W| \times \Omega}$. We compute the network utilization at intervals of every 250 incoming requests, and average it over $200,000$ connection requests. The resulting curves are plotted in Fig. 5.

WSP achieves the least network utilization because it routes connection requests over longer paths. This results in over usage of wavelength resources. The connection admission policy introduced in OTGA leads to effective utilization of bandwidth, thereby achieving the maximum network utilization.

5. Conclusion

In this paper, we investigated the on-line traffic grooming in a WDM-TDM switched optical mesh network without wavelength conversion capability. Using a novel exponential cost function, we proposed a routing algorithm for the concerned problem. We compared and analyzed the performance of the proposed algorithm with the other known heuristics. The experimental results showed that the proposed algorithm outperforms the existing algorithms - ASP, SWP and WSP with respect to bandwidth blocking ratio and network utilization.

6. Acknowledgment

It is acknowledged that the work by the authors was supported by research grant #DP0449431, funded by the Australian Research Council under its Discovery Schemes.

References

[1] R. Ramaswami and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, Morgan Kaufmann Publishers, 1998

[2] R. Srinivasan and A. K. Somani, Request-specific routing in WDM grooming networks, *Proc. IEEE ICC'02*, vol. 5, pp. 2876 - 2880, April/May 2002

[3] R. Srinivasan and A. K. Somani, Dynamic routing in WDM grooming networks, *Photonic Network Communications*, vol. 5, no. 2, pp. 123-135, March 2003

[4] R. Srinivasan and A. K. Somani, A generalized framework for analyzing time-space switched optical networks, *IEEE Journal on Selected Areas in Communications*, vol. 20, no. 1, pp. 202-215, Jan. 2002

[5] J. Yates, J. Lacey and D. Everitt, Blocking in multiwavelength TDM networks, *Proc. 4th International Conference on Telecommunication, Systems, Modeling and Analysis*, pp. 535-541, March 1996

[6] B. Wen and K. Sivalingam, Routing, wavelength and time-slot assignment in time division multiplexed wavelength routed optical networks, *Proc. IEEE INFOCOM'02*, vol. 3, pp. 1442-1450, June 2002

[7] R. Sriram, G. Manimaran and C. S. R Murthy, Preferred link based delay-constrained least-cost routing in wide area networks, *Computer Communications*, vol. 21, no. 18, pp. 1655-1669, Nov. 1998

[8] J. Aspnes et al., On-line routing of virtual circuits with applications to load balancing and machine scheduling, *Journal of the ACM*, vol. 44, no. 3, pp. 486-504, May 1997