This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

Dynamic Light Trail Routing and Protection Issues in WDM Optical Networks

Weiyi Zhang, Guoliang Xue, *Senior Member, IEEE*, Jian Tang, Krishnaiyan Thulasiraman, *Fellow, IEEE*

***Abstract*— In this paper, we study dynamic light trail routing in a WDM optical network. We present an efficient algorithm for establishing a light trail routing for a new connection request, while using minimum network resources. We also study survivable routing using light trail technology. We present an efficient heuristic for computing a pair of working and protection light trails for a given connection request. Simulation results are presented which demonstrate the advantages of our routing schemes.**

Keywords: **WDM optical networks, light trail routing, protection and restoration.**

I. INTRODUCTION

Wavelength Division Multiplexing (WDM) network effectively increases single-link bandwidth from 10Mbps to over 160Gbps. All optical circuits each on a separate wavelength called lightpaths [1] represent the first major method for optical communication. The granularity provided between a source node and a destination node is that of a complete wavelength. Once a lightpath is set up, the entire wavelength is used exclusively by the connection's source and destination node-pair. No sub-wavelength sharing between nodes along the lightpath is allowed. However, it is often observed that the bandwidth requirement in today's network is often dynamically varying and does not justify the need for allocating an entire wavelength. Therefore, the wavelength capacity may be underutilized.

Multiprotocol Label Switching (MPLS), Optical Burst Switching (OBS) [5], [13], [15], and Grooming technique [12], [16], [17] have been proposed to improve the ability of WDM networks to utilize the fiber resources more efficiently. A new technology termed *light trail* was proposed in [7] to avoid the inability of intermediate nodes to use a connection wavelength, and the constant reconfiguration of switches. The authors in [4] proposed a scheme to avoid the collision occurring between different connections along a light trail. An important issue of light trail routing is to construct light trails to carry traffic in WDM networks. Static light trail routing problem was studied in [3], [10] recently. However, to the best of our knowledge, no work considered light trail routing for the dynamic traffic. Another important issue in light trail routing is to provide a seamless flow of traffic between source and destination node pairs in the event of a link failure. Each working light trail needs a pre-computed link-disjoint light trail as a backup. [11] studied this issue for static traffic. To our best knowledge, there is no previous work addressing this issue for dynamic traffic.

In this paper, we consider the dynamic light trail routing problem. For each dynamically arriving connection request, we find a light trail to carry it with the objective of consuming a minimum number of free wavelengths. Moreover, we consider protection issues in light trail routing in WDM networks with dynamic traffic. Given a connection request, the problem is to find a pair of working and backup light trails.

The rest of the paper is organized as follows. Light trail technique is reviewed in Section II. In Section III, we present the light trail routing problem for dynamic traffic as well as an effective algorithm to solve the problem which aims at accepting more connections by improving wavelength utilization. In Section IV, we present a heuristic to compute a pair of working and protection light trails for each incoming connection request. In Section V, we present simulation results comparing light trail and lightpath schemes. We conclude this paper in Section VI.

II. LIGHT TRAIL TECHNOLOGY

The concept of *light trail*, as well as a hardware platform and a software protocol for implementing it, was proposed by Gumaste and Chlamtac in [7]. Readers can refer [4], [7] for details of the light trail architecture.

Compared to other schemes based on traditional lightpath routing, e.g. traffic grooming, light trail has some unique properties. First of all, light trail technology avoids costly OEO switching at intermediate nodes, which is necessary in traffic grooming. Each node on the light trail supports the *drop and continue* function. Each node taps a sufficient amount of optical power from the signal for local processing. The remaining optical power is sent through the optical shutter where optical routing decisions are made based on the shutter configuration. The node processes the tapped optical signal and determines whether the data is for it. The second property is that each optical connection on a light trail takes up the light trail exclusively. Each node can transmit data to any downstream node as long as it finds the optical bus is free.

Weiyi Zhang, Guoliang Xue and Jian Tang are with the Department of Computer Science and Engineering, Arizona State University, Tempe, AZ 85287-8809. Email: {weiyi.zhang, xue, jian.tang}@asu.edu. Research supported in part by NSF ITR grant ANI-0312635 and ARO grant DAAD19-00-1-0377.

Krishnaiyan Thulasiraman is with the School of Computer Science, University of Oklahoma, Norman, OK 73019. Email: thulasi@ou.edu. Research supported in part by NSF ITR grant ANI-0312435.

 0-7803-9415-1/05/$20.00 © 2005 IEEE

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.

[4], [7] both presented collision avoidance and re-transmission schemes using light trail technology. Thirdly, a light trail can be dimensioned (expanded or contracted) to meet certain applications, which is implemented by the protocol proposed in [7]. The fourth difference between light trail and traditional lightpath is that the length of a light trail can not be very long due to the optical power loss at each node. According to the studies in [2] and [3], the current expected hop-length of a light trail is 5. In this paper, we also follow this limitation.

III. Dynamic Light Trail Design

An important issue in a light trail network is how to construct light trails to carry traffic in the network. The authors of [3] studied the *static* light trail routing problem. In this paper, we study the problem of *dynamic light trail routing* where the connection requests come and leave dynamically.

We are given a directed network $G(V, E)$, where V is the set of n nodes and E is the set of m links. On each link of G, there are W wavelengths. We use WG_i to represent the wavelength plane corresponding to wavelength λ_i, which is composed of all nodes in G, all links in G for which wavelength λ_i is available and all light trails using wavelength λ_i. We assume that there is no wavelength converter at any node. We use $\mathcal{LT}$ to denote the set of existing light trails, and use $\mathcal{LT}_i$ to denote the set of light trails on wavelength plane WG_i. The length (in hop) of a light trail is bounded by L_{max}. Following is the problem statement.

***INPUT:* A directed connected network G, the set of existing light trails $\mathcal{LT}$, the maximum hop length of a light trail L_{max}, and a new connection request (s, t).**
***OUTPUT:* A single-hop or multi-hop light trail for the connection request with the properties:**

1) **The length of each light trail is no longer than L_{max}**
2) **Consumes the minimum number of free wavelengths**

Our dynamic light trail routing algorithm is presented in Algorithm 1. On each wavelength plane WG_i of G, there is a set of existing light trails $\mathcal{LT}_i$. If both s and t are in some existing light trail, and s is an upstream node of t, this request can be satisfied by reusing this light trail. Otherwise, we construct an auxiliary wavelength plane WG'_i from WG_i, and establish a new light trail to carry this connection request. All the free edges in WG_i, which are not in any light trail on WG_i, are copied into WG'_i. For each light trail lt in $\mathcal{LT}_i$, we transform it into an edge in WG'_i. There are 4 cases to be considered:

Case 1: Both s and t are on a light trail lt, but t is an upstream node of s. We simply ignore this light trail because this light trail is not useful for this connection request.

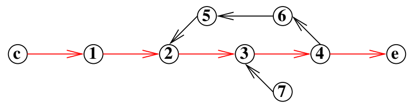
Case 2: Light trail lt does not contain s and t. Suppose the *convener node* of lt is c and the *end node* of lt is e. We add a *shortcut* edge (c, e) in the auxiliary graph WG'_i. The reason to do this is that we will either use the whole light trail or not use it at all. Though in [8], [9] a light trail tree is possible, we follow the light trail architecture in [4], [7] in this paper, and assume a light trail can only be dimensioned linearly. A very small cost ε is assigned to the *shortcut* edge (c, e), and we record the length of this edge to be the number of hops of the light trail lt. This case is illustrated in Fig. 1. On wavelength plane WG_i, we already have a light trail from c to e, marked by red links. We also have some free edges marked by black edges. We shrink light trail $(c, 1, 2, 3, 4, e)$ to a new edge (c, e) in wavelength plane WG'_i, while all the nodes and free edges are kept intact.

Algorithm 1 $LTRouting(G, s, t, \mathcal{LT}, L_{max})$

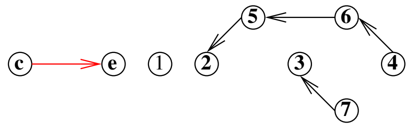
```
step_1 for each wavelength plane WG_i
         Construct a new wavelength plane WG'_i;
         Add all nodes in WG_i into WG'_i;
         for each light trail lt in LT_i
           if (s and t are both in lt) and (s is an upstream
           node of t)
             Return light trail lt to satisfy the request (s,t);
           endif
           if (neither s nor t is in lt)
             Add an edge eg from the convener node of
             lt to the endnode of lt in WG'_i;
             cost(eg) = ε, length(eg) = length of lt;
           endif
           if (only s is in lt) and (s ≠ end node of lt)
             Add an edge e from s to the end node of lt
             in WG'_i;
             cost(eg) = ε, length(eg) = length of lt;
           endif
           if (only t is in lt) and (t ≠ convener node of lt)
             Add an edge e from the convener node of
             lt to t in WG'_i;
             cost(eg) = ε, length(eg) = length of lt;
           endif
         endfor
         for each free edge eg on WG_i
           Add edge eg into WG'_i;
           cost(eg) = M, length(eg) = 1;
         endfor
         FindLightTrail(WG'_i, cost, length, L_max)
         if (there is a feasible path in WG'_i)
           Add the path into set CandidateLightTrail;
         endif
       endfor
step_2 if (CandidateLightTrail is not empty)
         Choose the minimum cost path P, which uses
         wavelength l, from CandidateLightTrail;
         Recover P to be a set of edges MinLT
         on wavelength plane WG_l;
         UpdateLightTrail(WG_l, MinLT);
         Return the found light trail;
       else
         Drop this connection request;
       endif
```

Case 3: Only s in the light trail lt. If s is the end node e, we ignore the light trail because it is not useful for this connection

This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE GLOBECOM 2005 proceedings.



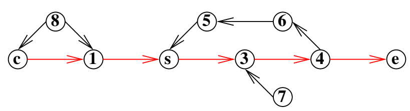
(a) Original Wavelength Plane WG_i



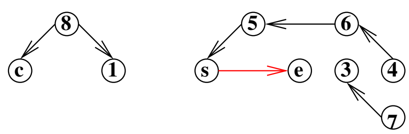
(b) Auxiliary Wavelength Plane WG_i'

Fig. 1. Transformation of Wavelength Plane for Case2.

request. Otherwise, we add a *shortcut* edge in WG_i' from s to node e. If we reuse lt for the new connection, we need to expand the end node to the destination node, and need not to care about the part from the *convener node* c to s on lt. Similarly, we assign a very small cost ε to the new edge (s, e), and record the length of this edge to be the number of hops of lt. We illustrate this case in Fig. 2.



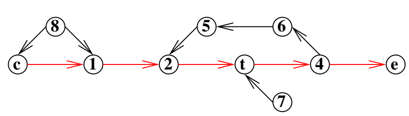
(a) Original Wavelength Plane WG_i



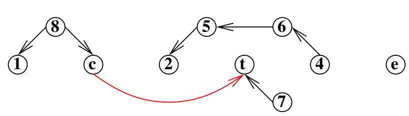
(b) Auxiliary Wavelength Plane WG_i'

Fig. 2. Transformation of Wavelength Plane for Case 3.

Case 4: Only t in the light trail lt. If t is the *convener node* c of lt, we ignore this light trail. Otherwise, we replace lt with a *shortcut* edge (c, t) in WG_i', assign a very small cost ε to the new edge (c, t), and record the length of this edge to be the number of hops of lt. Fig. 3 gives an illustration of this case.



(a) Original Wavelength Plane WG_i



(b) Auxiliary Wavelength Plane WG_i'

Fig. 3. Transformation of Wavelength Plane for Case 4.

Moreover, for each free edge on WG_i, we assign cost M and length 1 to it, where $M \gg \varepsilon$. In this way, we would reuse as many existing light trails as possible.

Algorithm 2 $FindLightTrail(WG_i', cost, length, L_{max})$

step_1 From WG_i', construct a directed graph ${WG_i'}^{L_{max}}$ with node set $V_{L_{max}} = V \times \{0, 1, ..., L_{max}\}$ and edge set $E_{L_{max}}$. If (u, v) is an edge in E, then $E_{L_{max}}$ contains an edge from (u, C_2) to (v, D_2) such that $D_2 = C_2 + length(u, v)$. The cost of all such edges is $cost(u, v)$. In addition, $E_{L_{max}}$ also contains zero cost edges from vertex (t, C_2) to (t, D_2), where $D_2 = C_2 + 1$.

step_2 Compute the shortest paths from $(s, 0)$ to all other nodes in ${WG_i'}^{L_{max}}$.

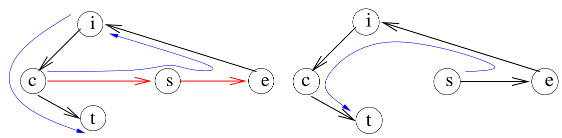
step_3 if there is a path p from $(s, 0)$ to a node in the form (t, L_{max}), return the path;

Algorithm 3 $UpdateLightTrail(WG_l, MinLT)$

step_1 while (*MinLT is not EMPTY*)
    Pop an edge (u, v) from *MinLT*
    Mark node u;
    if (node v has been marked)
        Insert (u, v) back to the head of *MinLT*;
        Set up all edges in *tempLT* to be a new light trail *newlt*;
        Insert *newlt* into set of light trails *foundlighttrails*;
        Remove all light trails in $\mathcal{LT}_i$ which have been expanded into *newlt*;
        Unmark all marked nodes;
        Empty set *tempLT*;
    endif
    else
        insert (u, v) to set *tempLT*;
    endifelse
endwhile

step_2 Return *foundlighttrails*

After finding a path for the connection request (s, t) in graph WG_l', we need to find the corresponding s–t path in the original wavelength plane WG_l by replacing each *shortcut edge* by the segment it represents. The s–t path in WG_l' becomes a set of edges, $MinLT$, from s to t in WG_l, which we call a $walk$ from s to t. A $walk$ may not be a path because it could be cyclic. We use Fig. 4 to illustrate this. The original wavelength plane is shown in Fig. 4(a). There is an existing light trail *(c, s, e)*. Other edges are free edges. The auxiliary graph is in Fig. 4(b). We find a path (s, e, i, c, t) in Fig. 4(b) to satisfy connection (s, t). However, after the path on WG_l obtained by replacing the shortcut edge (s, e) by the light trail (c, s, e) it represents, we obtain the walk (c, s, e, i, c, t) shown in Fig. 4(a), which is not a path. We need to convert this $walk$ into a multi-hop light trail (consisting (c, s, e, i) and (i, c, t) for this example, as shown in Fig. 4(a)). Algorithm 3 finds this light trail and dimensions each hop of it.



(a) Original Wavelength Plane

(b) Find a *walk* in auxiliary wavelength Plane.

Fig. 4. Update a light trail in WG'_l.

Theorem 3.1: Given a directed connected network G, existing light trails $\mathcal{LT}$,and a connection request (s,t), Algorithm 1 can find an optimal solution, in terms of consuming minimum number of free wavelengths, for the dynamic light trail routing problem in linear time.

PROOF. In Step_1 of Algorithm 1, An auxiliary graph WG'_i is constructed for each wavelength plane WG_i. Each edge will be processed at most once since each edge is either free or in just one light trail on WG_i. Therefore it takes $O(m)$ time to construct WG'_i from WG_i. Then we need to find a path to satisfy the connection (s,t). If there exists a path whose hop-length is at most L_{max} that can satisfy the connection request, Algorithm 2 can find a path which has the minimum cost and whose length is no more than L_{max}. The correctness of the algorithm can be proved similarly as in [14]. Note that in Algorithm 2, ${WG'_i}^{L_{max}}$ has $n(L_{max}+1)$ nodes and $O(2mL_{max}+nL_{max})$ edges, where n and m are the number of nodes and edges in G, respectively. Since ${WG'_i}^{L_{max}}$ is an acyclic graph, it takes $O(n(L_{max}+1)+2mL_{max}+nL_{max})$ = $O((m+n)L_{max})$ time to find a shortest path from $(s,0)$ to all other nodes in ${WG'_i}^{L_{max}}$. Therefore, it takes $O(m+(m+n)L_{max})$ time to find a candidate path on each wavelength plane. Consequently Step_1 takes $O(W(m+(m+n)L_{max}))$ to find at most W candidate paths. In Step_2, we can find the optimal path P in $O(W)$ time, and convert it to the corresponding $walk$ $MinLT$ on wavelength plane WG_l in $O(L_{max})$ time. Next, Algorithm 3 will check each edge on $MinLT$ at most twice to find the the final light trail. Its running time is $O(L_{max})$ because there are at most L_{max} edges on $MinLT$. Thus totally Step_2 needs $O(W+L_{max})$ time in the worst case.

Overall, the time complexity of Algorithm 1 is $O(W(m+(m+n)L_{max})+W+L_{max})$. Since W and L_{max} are small constants, the running time of Algorithm 1 is $O(m+n)$. $\square$

IV. Survivable Light Trail Network Design

In this section, we consider the design of a survivable WDM network using light trails to carry traffic. For each incoming connection request, we not only find a working light trail to satisfy it, but also find a link-disjoint backup light trail for the working light trail in case of single link failure. When a single link failure happens, we can switch the traffic from the working light trail to the backup light trail quickly so that we will not loss much information due to the link failure.

Algorithm 4 $Survivable\ LTDesign(G,s,t,LT)$

step_1 Apply Step_1 of $LTRouting(G,s,t,LT)$;
  if set $CandidateLightTrail$ is empty, block this request;
step_2 for first k minimum working light trails found in set $CandidateLightTrail$
    pick up a light trail as the working light trail;
    remove this working light trail in network G;
    Apply Step_1 of $LTRouting(G,s,t,LT)$;
    if set $CandidateLightTrail$ is not empty
      pick up the minimum light trail in it to be the backup light trail;
      Return the working and backup light trails;
    endif
    else
      Recover the working light trail in G;
    endifelse
  endfor
step_3 Block the connection request (s,t);

Algorithm 4 firstly uses Step_1 of Algorithm 1, and has at most W working light trail candidates. If there is no such candidate, we drop the connection request. Otherwise, we choose any one of the k minimum light trail candidates to be the working light trail, and try to find a protection light trail for it. If a protection light trail is found, we return the working and protection light trails, and accept the connection. If there is no protection for this working light trail, we move to next working light trail candidate and try to find protection again. If neither of the k working light trail candidates has protection, we drop this connection.

V. Simulation Results

In order to study the practical effectiveness of light trail scheme, we have performed computational comparisons between light trail and traditional lightpath schemes.

We used three known network topologies with bidirectional links, NSFNET (14 nodes and 42 edges), ARPANET (20 nodes and 64 edges) and ITALIANNET (33 nodes and 134 edges). For each network topology, we consider the cases where the number of wavelength on each link is 4, 8 and 16, respectively. 800 connections are randomly generated for each network topology. Each connection request was generated at each time unit, and has a life time which specifies how many time units it will last. In our simulation, the life time is set to a random integer uniformly distributed in [1,100]. In the tables, the column with label **Network** specifies the topologies, which are represented by the number of nodes N and edges E of the network. The column with label **Wave** specifies the number of wavelength per link. The next 2 columns show the results for **Light Trail** and **LightPath** schemes. Each result contains 2 parts: *Cons* represents the number of connections accepted

 0-7803-9415-1/05/$20.00 © 2005 IEEE

by Light Trail and Lightpath respectively, *Waves* records the number of wavelengths used for setting up those accepted connections.

In the first scenario, we observe how well the light trail perform compared with lightpath scheme for each connection without protection.

TABLE I

COMPARISON WITHOUT PROTECTION

Network	Wave	Light Trail		Lightpath	
$N \times E$	W	*Cons*	*Waves*	*Cons*	*Waves*
14 x 42	4	800	83	316	124
14 x 42	8	800	83	800	141
14 x 42	16	800	83	800	141
20 x 64	4	624	191	278	196
20 x 64	8	800	270	460	328
20 x 64	16	800	270	737	508
33 x 134	4	557	429	269	483
33 x 134	8	671	616	448	854
33 x 134	16	727	721	630	1342

Results for the three network models are presented in Table I. From the table, we can see that light trail scheme always accommodated more connections with less free wavelength consumption. For example, in ARPANET with 8 wavelengths on each link, light trail satisfied 42.5% more connections than lightpath scheme while consumed 11.3% less free wavelengths.

In the second scenario, we compare the light trail and lightpath schemes for the case of survivable WDM routing. The results are presented in Table II. For both the lightpath protection scheme and the light trail protection scheme, we have used shared protection. First we compute a shortest (minimum free wavelength usage) working lightpath for a connection, and then compute shortest backup path which is link-disjoint with the candidate working lightpath but could share wavelength with other protection paths if possible.

TABLE II

COMPARISON WITH PROTECTION

Network	Wave	Light Trail		Lightpath	
$N \times E$	W	*Cons*	*Waves*	*Cons*	*Waves*
14 x 42	4	494	120	170	137
14 x 42	8	800	180	800	185
14 x 42	16	800	179	800	185
20 x 64	4	274	232	203	203
20 x 64	8	419	352	404	368
20 x 64	16	671	483	655	564
33 x 134	4	200	465	209	464
33 x 134	8	445	835	450	901
33 x 134	16	674	1185	709	1456

From the table, due to the limitation of light trail length, we observe that the light trail and lightpath provided similar results. However, basically light trail scheme still works better than lightpath approach, especially when the number of wavelengths per link is very small. For example, in ARPANET, with 4 wavelengths per link, compared to lightpath approach, light trail accommodated 71 more connections as well as used 29 less wavelengths. When the number of wavelength per link increased to 8, light trail accommodated 19 more connections with 16 less wavelengths than lightpath scheme.

VI. CONCLUSIONS

In this paper, we have studied dynamic light trail routing and survivable light trail routing in WDM optical networks. In particular, we have presented an efficient algorithm for computing a light trail for an incoming connection request. We have also presented an efficient heuristic for survivable light trail routing by establishing a pair of working and backup light trails. Simulation results are presented to demonstrate the advantages of light trail scheme over the traditional lightpath schemes.

REFERENCES

[1] I. Chlamtac, A. Ganz, G. Karmi, Lightpath communications: a novel approach to high speed optical WANs, *IEEE Trans. on Commun.*, Vol. 40 No.7 July 1992, pp. 1152.

[2] I. Chlamtac, A. Gumaste, Light-trails: a solution to IP centric communication in the optical domain, Online publication, February 17, 2003, Springer-Verlag Berlin Heidelberg 2003.

[3] Jing Fang, Wensheng He, A.K.Somani, Optimal light trail design in WDM optical networks, *ICC'2004*, pp. 1699-1703.

[4] M.T. Frederick, N.A. VanderHorn, A.K. Somani, Light trails: a sub-wavelength solution for optical networking, *HPSR'2004*, pp. 175-179.

[5] Ge, F. Callegati, L. Tamil, On optical burst switching and self-similar traffic, *IEEE Communication Letters*, vol. 4, March 2000, pp. 98-100.

[6] A. Gumaste, T. Antony, DWDM Networks Design and Engineering Solutions, *McMillan Publishers*.

[7] A. Gumaste, I. Chlamtac, Light-trails: a novel conceptual framework for conducting optical communications, *HPSR'2003*, pp. 251-256.

[8] A. Gumaste, I. Chlamtac, Mesh implementation of light-trails: a solution to IP centric communication, *ICCCN'2003*, pp 178-183.

[9] A. Gumaste, I. Chlamtac, J.P. Jue, Light-frames: A pragmatic framework for optical packet transport, *ICC'2004*, pp. 1537-1542.

[10] A. Gumaste, G. Kuper, I. Chlamtac, Optimizing light-trail assignment to WDM networks for dynamic IP centric traffic, *Local and Metropolitan Area Networks, IEEE LANMAN 2004*, Apr. 2004, pp 113-118.

[11] Wensheng He, J. Fang, A. K. Somani, On survivable design in light trail optical networks, *Proceeding of 8th IFIP Working Conference on Optical Network Design and Modeling*, Feb, 2004.

[12] E. Modiano, P. Lin, Traffic grooming in WDM networks, *Communications Magazine, IEEE*, Vol. 39, July 2001, pp 124-129.

[13] S. Verma, H. Chaskar, R. Ravikanth, Optical burst switching: A viable solution for terabit IP backbone, *IEEE Network*, November/December 2000, pp. 48-53.

[14] Guoliang Xue, A. Sen, R. Banka, Routing with many additive QoS constraints, *Communications, 2003. ICC '03. IEEE International Conference on*, Vol. 1, 11-15 May 2003, pp 223-227.

[15] M. Yoo, C. Qiao, Optical burst switching (OBS) - a new paradigm for and optical Internet, *Journal of High Speed Networks (JHSN)*, vol 8. no. 1, 1999, pp. 69-84.

[16] Hongyue Zhu, Hui Zang, Keyao Zhu, B. Mukherjee, A novel generic graph model for traffic grooming in heterogeneous WDM mesh networks, *Networking, IEEE/ACM Transactions on*, Vol. 11, April 2003, pp 285-299.

[17] Keyao Zhu, B. Mukherjee, Traffic grooming in an optical WDM mesh network, *Selected Areas in Communications, IEEE Journal on*, Vol. 20, Jan. 2002, pp 122-133.

 0-7803-9415-1/05/$20.00 © 2005 IEEE