A Heuristic Method of Logical Topology Reconfiguration in IP/WDM Optical Networks

Sugang XU [*1], Kaoru SEZAKI [*2], and Yoshiaki TANAKA [*1,*3]
[*1,*3] 29-7 Bldg., 1-3-10 Nishi-Waseda, Shinjuku-ku, Tokyo, 169-0051, Japan.
[*2] Ew-305, Komaba 4-6-1, Meguro-ku, Tokyo, 153-8505, Japan.
xsg@aoni.waseda.jp, sezaki@iis.u-tokyo.ac.jp, ytanaka@waseda.jp

[*1] Global Information and Telecommunication Institute, Waseda University
[*2] Center for Spatial Information Science at the University of Tokyo
[*3] Advanced Research Institute for Science and Engineering, Waseda University

***Abstract*—WDM optical networks represent the direction to the future high capacity wide-area network applications. By creating the optical paths between several nodes in the core networks, logical topology can be created over the physical topology. The network can dynamically change its logical topology corresponding to the changing traffic conditions. In this paper we consider the method of logical topology reconfiguration in wavelength routed optical networks. The exact formulation of the logical topology problem is usually given as a Mixed Integer Linear Programming, but it grows intractable with increasing size of network. Heuristic approaches have been proposed, here we also proposed a different heuristic approach to determine the logical topology reconfiguration.**

***Index Terms*—IP/WDM networks, Logical topology, Reconfiguration, Heuristic method, GMPLS.**

I. Introduction

Wavelength division multiplexing (WDM) makes the huge optical capacity of fiber to be utilized by transmitting multiple signals on different wavelengths on a single fiber. WDM optical networks represent the direction to the future high capacity wide-area network applications. In IP/WDM networks, network nodes employ Wavelength Routing Switches (WRSs) and IP routers. Nodes are connected by fibers to form an arbitrary physical mesh topology. Any two IP routers in this network can be connected together by an all-optical WDM channel, called a light-path. Here a light-path is a point-to-point all-optical wavelength channel that connects a transmitter at a source node to a receiver at a destination node. By using WRSs at intermediate nodes and via appropriate routing and wavelength assignment, a light-path can create logical neighbors out of nodes that are geographically far apart in the network. Thus, a set of light-paths embeds a logical topology on the network. In the logical topology, a light-path carries not only the direct traffic between the nodes it interconnects, but also traffic from nodes upstream of the source (including the source) to nodes downstream of the destination (including the destination). Nodes that are not connected directly in the logical topology can still communicate with one another using the "multi-hop approach", namely, by using electronic packet switching at the intermediate nodes in the logical topology.

Consider N nodes network with an arbitrary but connected physical topology, in the most cases there are main four restrictions to design a logical topology:

1) The number of tunable transmitter and receiver at each node is limited, that is to say the degree of the node is limited;

2) The number of wavelength on each fiber is also limited, W wavelengths on each fiber;

3) The wavelength cannot be used by different light-paths on the same fiber;

4) Without the wavelength converter the light-path has to use the same wavelength along the path.

The motivation of logical topology design is to optimize the network performance, improving the congestion, delays and throughput metrics. And the network can dynamically change its logical topology corresponding to the changing traffic conditions. It is called as logical topology reconfiguration. The general approach to the logical topology reconfiguration problem has been a two-phase operation: first phase being a logical topology design for the new traffic conditions and second phase being a transition period from the old logical topology to the newly designed one, it should achieve the minimal traffic disruption. Normally it is formulated as a mixed integer linear programming (MILP) problem. In fact, this problem and some of its subproblems are known to be *NP-hard* problem. Thus for networks of large sizes it is impractical to attempt to solve this problem exactly. The heuristic algorithms are necessary.

In this paper we focus mainly on the method of logical topology transition with small operations. Instead of searching the exact solution of the logical topology reconfiguration, we try to find an approach to an approximated solution. First we introduce an improved transition approach based on the proposal in [4]. The idea is to find a closer and better logical topology from old one without requirement of the best target logical topology. By this approach we can find a relative good target under certain degree of performance loss, but with minimum transition op-

0-7803-7661-7/03/$17.00©2003 IEEE.

eration, which is more important to the real network operation. Then we propose a heuristic approach of the second phase, say the transition phase. This paper is organized as follows: In the section 2 we describe some related works briefly; In section 3 we describe the improved MILP module; In section 4 we propose a heuristic method; Section 5 provides the numerical results; Section 6 concludes the paper.

II. Related works

Reconfiguration capability has been seen as an outstanding feature of WDM routed optical networks, and several logical topology reconfiguration schemes have been proposed. In [1] authors study the logical topology design problem. It is formulated as a mixed integer linear programming problem with the goal of minimizing the congestions in the network. And a heuristic algorithm is then proposed, which finds an initial connectivity diagram and the corresponding routing, and then performs a local search by applying branch-exchange operations on some least loaded links of the connectivity diagram. The flow deviation method is used to solve the optimal routing problem after getting the connectivity diagram.

In [2], the idea is to achieve the minimal traffic disruption, the authors discuss strategies to transit from the current logical topology to the optimal logical configuration within the smallest steps of branch-exchange operations. This problem of finding the shortest sequence, so as to minimize the reconfiguration duration, can be equivalently reduced to the problem of finding a decomposition of an auxiliary graph into the largest number of vertex-disjoint cycles. Three polynomial-time greedy algorithms are proposed and compared to solve this *NP-hard* graph problem. Both theoretical and simulation results show that the length of a sequence increases at most linearly with the size of the network.

Studies in [1] and [2] present a complete reconfiguration scheme. Given a new traffic pattern, a corresponding optimal logical topology is computed by the algorithm proposed in [1]. The shortest branch-exchange sequence is then figured out by heuristic algorithms in [2].

Hitless reconfiguration is proposed in [3] as a reconfiguration process without the loss of any data. The transition between topologies is achieved by first establishing all new links without removing any link. Authors developed a sufficient theoretical condition for the reconfiguration process to be hitless, including the size of switches and the bound of wavelength numbers.

Study in [4] mainly focuses on the design of optimal logical topologies, for which an exact integer linear programming formulation is presented. For the reconfiguration problem, authors propose a methodology to obtain the new logical topology, based on optimizing a given objective functions, and minimizing the changes required to obtain the new logical topology from the current logical topology. Given a certain traffic matrix, there may be many different logical topologies, each of which has the same optimal value with regard to the objective function. The reconfiguration algorithm searches all possible optimal logical topology corresponding to the certain traffic, and finds the "closest" one.

Study in [5] shows some limitations of the formulation in [4], and proposes different objective functions that can be used for optimization in finding the best logical topology for a given traffic pattern. A modification to the reconfiguration algorithm in [4] is proposed, to include the trade-offs between the amount of reconfiguration steps and the objective that is to be optimized.

In [6], iterative reconfiguration algorithms are developed to track rapid changes in the traffic pattern. At each reconfiguration step, only one small change is made to the logical topology, therefore minimizing the disruption to the network. This approach is shown to provide near optimal reduction in the maximum link load.

A new reconfiguration algorithm [7] is developed to adjust the logical topology by adding or deleting one lightpath at the end of each measurement cycle if it is necessary to do so. The load imbalance is corrected directly by tearing down a lightpath that is lightly loaded or by setting up a new lightpath when congestion occurs. Note that this approach keeps varying number of lightpaths that can grow during the peak traffic hours and fall when traffic decreases.

Study in [8] shows that, the logical topologies created by the new adaptation approach [7] are close to optimal, and the number of lightpath changes is decreased drastically compared to the optimal reconfiguration method that recomputes the logical topology from scratch at every step [4].

III. Improved MILP module

A. Given

Here are some terminologies and notations often used in other studies. Also we will introduce some concepts which will be used in the following sections, and which are common to most formulations to the logical topology problem.

1) Physical Topology: A graph $G_p=(V,E_p)$ in which each node in the network is a vertex, and each fiber optic link between two nodes is an arc. Each fiber link is also called a physical link. The graph is assumed to be undirected;

2) Link indicator: If a physical link exists in the physical topology from a node i to another node j, denoted by p_{ij}, which is 1 if such a link exists in the physical topology and 0 if not;

3) Light-path: A light-path, is a clear optical channel between two nodes without O-E-O conversion, also we assume that there is not any wavelength conversion;

4) Logical Topology: A graph $G_l=(V,E_l)$ in which the set of nodes is the same as that of the physical topology, and each light-path is an arc, it is also called logical link. Here we create the bi-directional light-path between one pair of nodes;

5) Light-path indicator: if a light-path exists in the

logical topology from a node i to another node j, denoted by b_{ij}, which is 1 if such a light-path exists in the physical topology and 0 if not. Let $c_{ij}^{(k)}$ be the light-path wavelength indicator, $c_{ij}^{(k)}$ is 1 if a light-path from node i to node j uses the wavelength k, 0 otherwise. Let $c_{ij}^{(k)}(l,m)$ be the link-light-path wavelength indicator, to indicate whether the light-path from node i to node j uses the wavelength k and passes through the physical link from node l to node m;

6) Physical degree: The physical degree of a node is the number of the physical links that directly connect that node to other nodes;

7) Logical degree: The logical degree of a node is the number of transmitters and receivers (Ts/Rs), and it is also the number of light-paths connecting that node to other nodes. The number of light-paths originating and terminating at a node may be different, and we denote them by logical out-degree and logical in-degree respectively. In this chapter we assume they are equal as they often are and has the limitation of P;

8) Logical hops: The number of light-paths a given traffic packet has to traverse, in order to reach from source to destination node over a particular logical topology, is called the logical hop length of the path from the source to the destination node in that logical topology;

9) Traffic matrix: A matrix, which specifies the traffic between every pair of node in the physical topology. If there are N nodes in a network, the traffic matrix is an $N \times N$ matrix $\Lambda=[\lambda^{(sd)}]$, where $\lambda^{(sd)}$ is the average traffic from source node s to destination node d;

10) Logical traffic load: When a logical topology is established on a physical topology, the traffic from each source node to destination node must be routed over some light-paths. The aggregate traffic resulting over a light-path is the load offered to that logical link. If a light-path exists from node i to j, the load offered to that light-path is denoted by λ_{ij}, the component of this load due to traffic from source node s to destination node d is denoted by $\lambda_{ij}^{(sd)}$;

11) Wavelength amount on each fiber: The number of wavelength on each fiber is denoted by W;

12) Fiber amount on each physical link: The number of fiber on each physical link is denoted by F.

B. Logical topology Design

For the logical topology design, we can use the methods proposed in [4] with the objective (1) of minimum average hop number:

$$Minimize : \sum_{s,d}\sum_{i,j}\lambda_{ij}^{(sd)} \Big/ \sum_{s,d}\lambda^{(sd)} \qquad (1)$$

please see [4] for constraints and details, due to the space limitation we will not describe here.

C. Reconfiguration improvement

The motivation of logical topology design is to optimize the network performance. And the network can dynamically change its logical topology corresponding to the changing traffic conditions. Reconfiguration capability has been seen as an outstanding feature of WDM routed optical networks. Logical topology reconfiguration method proposed in [4] add a new constraint (2) in the MILP module when search the target logical topology configuration, where OPT_{target} stands for the optimized performance under the new traffic condition resolved offline in the first phase—logical topology design phase, without consideration of minimum transition operation. And the new objective function for the target logical topology configuration is to minimize the distance between the old logical topology and the target logical topology, shown as (3), where $b_{ij}(1)$ stands for the old logical topology, b_{ij} stands for the target logical topology.

$$\sum_{s,d}\sum_{i,j}\lambda_{ij}^{(sd)} \Big/ \sum_{s,d}\lambda^{(sd)} = OPT_{target} \qquad (2)$$

$$Minimize : \sum_{i,j}\left|b_{ij}(1)-b_{ij}\right| \qquad (3)$$

Because our aim of transition phase is to minimize the transition operation, the idea of the improvement is to relax the constraint (2) because sometimes with little performance loss, new solution would be closer to the old logical topology than the absolutely exact solution like [4]. Here we introduce into a new constraint (4) to replace that of (2), where $OPT_{endurable}$ stands for a tunable parameter, that is a lower line of performance requirement that is permitted.

$$\sum_{s,d}\sum_{i,j}\lambda_{ij}^{(sd)} \Big/ \sum_{s,d}\lambda^{(sd)} < OPT_{endurable} \qquad (4)$$

IV. Heuristic approach

A. Heuristic Logical Topology Design

For the logical topology design we proposed a heuristic method, which iteratively search the logical link. It is called as MALH(Minimum Average Logical Hop). From the connectivity consideration we start the lightpath establishment from the physical topology, we create one lightpath for each physical link. Then MALH searches one source destination pair s-d, which has biggest contribution to minimize the average logical hop among all possible node pair (we just create one light-path one step) and add $b_{ij}=1$, then update the logical topology, and repeat the process until there is not any logical link can be established. The contribution of one candidate logical link between node s and d is:

$$H_{sd} = \sum_{i=1}^{N}\sum_{j=1}^{N}(\lambda^{(ij)}h_{ij}) \Bigg/ \sum_{i=1}^{N}\sum_{j=1}^{N}\lambda^{(ij)} \quad (5)$$

where H_{sd} denotes the average hop if create lightpath between node pair (s,d), $\lambda^{(ij)}$ denotes the traffic demand between (i,j), h_{ij} denotes the number of hop between node pair (i,j) in current logical topology.

The algorithm is show as follows:

Step1: Make logical topology $G_l = G_p$

Step2: By given a traffic matrix $\Lambda=[\lambda^{(ij)}]$

Make $Q=(q_{ij})=(H_{ij})$; (where the H_{ij} denotes the contribution shown above (5) between node i and j in G_l)

Step3: Select the source destination pair $(imax,jmax)$ that $q_{imax,jmax}=\max_{i,j}(q_{ij})$;

if all source-destination pairs have been tried, then Goto Step5;

Step4: If node *imax* and *jmax* have fewer degree than P then

Find lowest available wavelength on the shortest propagation-delay path between *imax,jmax* in G_p (if there is more than one shortest path, scan them sequentially search the least load path);

If wavelength is available then

Create light-path and the logical link between node *imax* and *jmax;*

$b_{imax,jmax}=1;$

Update G_l;

Goto Step2; .

Else

$b_{imax,jmax}=0;$

Goto Step3;

Else

$b_{imax,jmax}=0;$

Goto Step 3;

Step5: End.

This is an iterative process, it searches the current most necessary logical link for the whole network to improve the current logical topology with small average logical hop, and then update the logical topology to find the next most necessary logical link. It starts from physical topology as the initial topology, and then iteratively solves the topology design problem by slightly changing the logical topology.

B. Traffic assumptions

Before we introduce our reconfiguration method we describe some assumptions on the traffic. Because logical topology reconfiguration will cause the current traffic disruption, generally, logical topology reconfiguration will be implemented after a period of time, for example several weeks or months. The statistic of traffic during a period of time will be used as the input of the logical topology design.

On the other hand, for the efficient network resource utilization, we hope the logical topology to be reconfigured more often to keep up with the traffic changes. Here we divide long-range traffic into relative short-range traffic, for example we divide weekly traffic into days order. By gathering several continual two-days or three-days traffic data $\Lambda_{days}(i)$ (where i=1, 2, 3) we can get the averaged traffic data Λ_{weeks} of one week (6) see Figure.1.

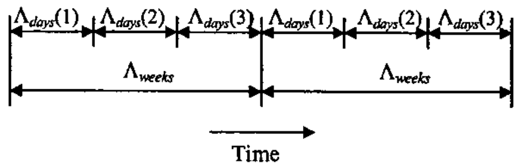

Figure.1 Divide weeks traffic into days traffic.

$$\Lambda_{weeks}=Average(\Lambda_{days}(1), \Lambda_{days}(2), \Lambda_{days}(3)) \quad (6)$$

where *Average*() is a function that calculates the average of days traffic. Each of $\Lambda_{days}(i)$ and Λ_{weeks} is a matrix.

C. Logical Topology Reconfiguration

Different from previous studies, which consider traffic before-and-after reconfiguration independently, we create some relations among the continual traffics. Our idea of logical topology reconfiguration is that we use the long-range part of traffic among several continual two-days or three-days traffic to create relative stable common logical topology with parts of network resources (transmitters/receivers, wavelengths), and then make use of the left resources to create the other part of logical topology for each $\Lambda_{days}(i)$ traffic condition. We also can see the logical topology design is down in several layers. For example, 1) in the first layer, to keep the connectivity of the logical topology we create the base logical topology as same as the physical topology; 2) by given the base layer topology as input topology, in the second layer, we use the long-range part of traffic among several continual two-days or three-days traffic to create relative stable common logical topology with parts of network resources; 3) by given the topology yielded from the second layer design, in the third layer, we make use of the left resources to create the active part of logical topology to optimize the resource utilization for each short-range traffic. During the logical topology transition from a time period to other, the lightpath add/delete operation will just be limited within a small part of logical topology (in the active third layer). So the amount of transition operation may be smaller than that of two layers design (without the common layer). Furthermore we can use the methods talked in section 3 or proposed by other outstanding studies to optimize the performance and minimize the transition within the active third layer. See Figure.2. The logical topology in base+common layer can be seen as the combined basic layer, and the third layer can be seen as the active layer.

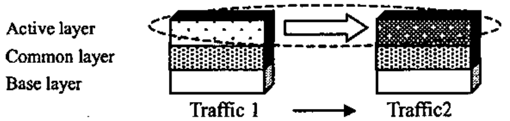

Figure.2 Layered Logical topology design and transition.

D. Consideration of Traffic routing

Hitless reconfiguration is proposed in [3] as a reconfiguration process without the loss of any data. The transition between topologies is achieved by first establishing all new links without removing any link. But if we have not the enough resources to create all the new lightpaths before release the useless lightpaths, how will the current traffic be kept from disrupting? Here based on the layered logical topology design approach, we keep certain amount of lightpaths without changing anymore (within a week unit) they are the lightpaths in the first and second layer. In MPLS(or GMPLS) and nowadays Internet routing, routing information is maintained within each IP router/MPLS switch, generally if one lightpath add/delete operation occurs, the routing table update in packet layer should be triggered immediately. If we reconfigure the lightpaths one bye one like that proposed in [7], the routing table should be updated frequently, the control signal traffic will be increased and cause the network unstable. If we have a smaller transition schedules, we could do the reconfiguration at same time to decrease the bad effect caused be a sequence of network operations. The procedure is:

1) Optimize the traffic routing within packet layer by given the logical topology yielded from the second lay design and the long-range average week traffic assumption;

2) Backup this routing table in each IP router/ MPLS switches;

3) Optimize the traffic routing within packet layer by given the total logical topology yielded from the third lay design with the short-range days traffic assumption, and use this routing table in each IP router/ MPLS switches for current traffic;

4) When reconfiguration event is triggered in two-days or three-days level, compute the target logical topology and optimize the packet traffic routing offline.

5) Switch the routing table of each IP router/ MPLS switches from current routing table to the backup routing table saved in step 2. Then withdraw the traffic back to the base and common layer logical topology;

6) Until the traffics are moved away from the lighpath that should be released, reconfigure the logical topology computed in step4;

7) Switch the routing table of each IP router/MPLS switches to new routing table computed in step 4, and expand the traffic to the new logical topology.

During the switching of the routing table in step 5 and step 7, the synchronization of the routing table update among all the IP router/MPLS switches is very important. Routing table switching should wait for a period of time till the new routing tables in each MPLS switches are consistent.

V. Numerical Results

This section presents numerical examples of the logical topology reconfiguration problem, using the ARPANET-II with 21 nodes as our physical topology (as same as that used in [9]). Network nodes employ Wavelength Routing Switches (WRSs) and IP routers or MPLS switches. We assume there are 10 Ts/Rs in IP router/MPLS switch on each node (*P=10*). For the simplicity we assume there is only single fiber between two neighbor nodes (*F=1*). And wavelength amount on each fiber is 256 (*W=256*). We generated 3 groups of traffic matrices, each with 3 traffic matrices $\Lambda_{days}(1)$, $\Lambda_{days}(2)$, $\Lambda_{days}(3)$. Each traffic matrix corresponds to two-days traffic. Each group of traffic corresponds to Λ_{weeks} that in one week. The traffic matrix is randomly generated, such that 3 nodes are randomly picked up as cluster core nodes with higher traffic to/from other 18 nodes over the range (0, 100); among left 18 nodes, 5 nodes are randomly picked up with uniformly distributed traffic to/from the left 13 nodes over the range (0,50); and among left 13 nodes traffic is uniformly distributed over the range (0,10). And for the simplicity, we create one pair of symmetric lightpaths between any source destination node pair.

In Table 1, 3, 5, we compared the average hop of logical topology within the each traffic group. We reserve from 0 to 9 Ts/Rs to create the basic layer. We can find by reserving several Ts/Rs the average hop performance does not turn down much comparing to that without basic layer approach. With bigger basic layer the performance is poor.

Table 1 Average hop results in traffic group 1

Ts/Rs	$\Lambda_{days}(1)$	$\Lambda_{days}(2)$	$\Lambda_{days}(3)$
0	1.223488	1.294843	1.290880
5	1.228987	1.303064	1.279460
6	1.226237	1.292227	1.283520
7	1.252160	1.297833	1.294980
8	1.254910	1.306054	1.306880
9	1.290652	1.312780	1.323520

Table 2, 4, 6 show us the transition operation results. We can find by reserving more and more Ts/Rs to create the basic layer, when reconfiguration occurs, the path add/delete operation amount decrease smaller and smaller, that is because the unchanged part (the basic layer) grows bigger, the active layer part turns to be smaller and smaller.

Table 2 Transition Operations in traffic group 1

Ts/Rs	$\Lambda_{days}(1)$ to $\Lambda_{days}(2)$	$\Lambda_{days}(2)$ to $\Lambda_{days}(3)$
0	102	110
5	108	116
6	102	100
7	92	84
8	70	58
9	44	38

Table 3 Average hop results in traffic group 2

Ts/Rs	$\Lambda_{days}(1)$	$\Lambda_{days}(2)$	$\Lambda_{days}(3)$
0	1.286986	1.244063	1.267201
5	1.298630	1.285349	1.265531
6	1.321233	1.312751	1.263861
7	1.293836	1.248447	1.274883
8	1.298973	1.258312	1.286239
9	1.287671	1.288272	1.301603

Table 4 Transition Operations in traffic group 2

Ts/Rs	$\Lambda_{days}(1)$ to $\Lambda_{days}(2)$	$\Lambda_{days}(2)$ to $\Lambda_{days}(3)$
0	126	140
5	96	120
6	98	114
7	90	92
8	54	70
9	26	38

Table 5 Average hop results in traffic group 3

Ts/Rs	$\Lambda_{days}(1)$	$\Lambda_{days}(2)$	$\Lambda_{days}(3)$
0	1.232627	1.259769	1.307279
5	1.236434	1.284113	1.301912
6	1.233742	1.265535	1.310362
7	1.234114	1.266464	1.316193
8	1.243776	1.292441	1.323717
9	1.278707	1.3123	1.328749

Table 6 Transition Operations in traffic group 3

Ts/Rs	$\Lambda_{days}(1)$ to $\Lambda_{days}(2)$	$\Lambda_{days}(2)$ to $\Lambda_{days}(3)$
0	126	126
5	110	108
6	96	96
7	94	96
8	78	76
9	46	44

VI. Conclusions

Because our aim is to design a heuristic method of logical topology reconfiguration, the improved MILP model talked in section 3 is a reference module to our heuristic approach. There exist a tradeoff between the need of smallest transition operation and the optimal performance. With more resources to create the bigger stable common layer topology, the smaller transition operation will be expected due to the smaller reconfiguration space in the third layer, the logical topology might not be optimal to dedicated traffic. On the other hand, if we use all resources to optimize the logical topology, the reconfiguration space might be very huge. Furthermore by using heuristic with local optimization the target logical topology might be quite "far" from the old logical topology. So we hope to get certain tradeoff between small transition requirement and the optimal performance.

In our days-unit reconfiguration example, the common layer is created basing on the weeks average traffic assumption, so the smaller transition reconfiguration of logical topology is done between two-days or three-days units. Although the traffic changing from week to week still triggers the whole reconfiguration as that of nowadays, more often logical topology reconfiguration with little transition operation that can improve the network performance is our aim.

In fact the routing table switching approach is not only suitable for our three layered logical topology design and reconfiguration, it also can be used in other reconfiguration approaches for example in weeks reconfiguration, the unchanged part between old and new logical topology can be seen as same as the basic layers (physical topology + common layer) logical topology in our proposal.

REFERENCES:

[1] J. F. P. Labourdette and A. S. Acampora, "Logically rearrangeable multihop lightwave networks," *IEEE Trans. Commun.*, vol. 39, no. 8, pp. 1223-1230, August 1991.

[2] J. F. P. Labourdette, G. W. Hart, and A. S. Acampora, "Branch-exchange sequences for reconfiguration of lightwave networks," *IEEE Trans. Commun.*, vol. 42, no. 10, pp. 2822-2832, October 1994.

[3] K. Bala et al., "Towards hitless reconfiguration in WDM optical networks for ATM transport," in *IEEE GLOBECOM 1996*, pp. 316-320, 1996.

[4] D. Banerjee and B. Mukherjee, "Wavelength-routed optical networks: Linear formulation, resource budgeting tradeoffs, and a reconfiguration study," *IEEE/ACM Trans. Networking*, vol. 8, no. 5, pp. 598-607, October 2000.

[5] B. Ramamurthy and A. Ramakrishnan, "Virtual topology reconfiguration of wavelength routed optical WDM networks," in *IEEE GLOBECOM 2000*, pp. 1269-1275, 2000.

[6] A. Narula-Tam and E. Modiano, "Dynamic load balancing for WDM based packet networks," in *IEEE INFOCOM 2000*, pp. 1010-1019, 2000.

[7] A. Gencata and B. Mukherjee, "Virtual-topology adaptation for WDM mesh networks under dynamic traffic," in *IEEE INFOCOM 2002*. http://www.ieee-infocom.org/2002/papers/333.pdf

[8] A. Gencata, L. Sahasrabuddhe, and B. Mukherjee, "Virtual-topology adaptation with minimal lightpath change for dynamic traffic in WDM mesh networks," in *OFC, 2002*.

[9] A. Mokhtar, M. Azizoglu, "Adaptive Wavelength Routing in All-Optical Networks," *IEEE /ACM Trans. Networking*, vol.6, no.2, pp.197-206, April 1998.