

Integrated Logical Topology Design and Traffic Grooming in Re-configurable WDM Networks[#]

Kwang-II Lee, Lakshmi Sudarsan, Mark Shayman*

* Department of Electrical and Computer Engineering and Institute for Systems Research
University of Maryland, College Park, MD 20742
{kilee, sharmila, shayman}@glue.umd.edu

Abstract : Wavelength-division multiplexing (WDM) networks provide reconfigurability and the possibility of creating single hop communication channels between end nodes. This eliminates the electronic processing at intermediate nodes and reduces delay significantly. However, because of the limited number of network interfaces and other scalability issues, single hop channels do not support all traffic demands. Some of the traffic requires routing or switching over multi-hop lightpaths to reach the destination. Such traffic is referred to as multi-hop traffic. The optimization of multi-hop traffic has significant impact on the performance of optical networks.

In this paper, we propose an integrated logical topology design and traffic-grooming algorithm for the optimization of the multi-hop traffic. This is enabled by the multi-hop lighpath setup mechanism, which is to establish lightpaths from/to intermediate nodes to/from end nodes for multi-hop traffic. The proposed algorithm considers traffic grooming for multi-hop traffic when the logical topology is designed and always gives high priority to high traffic demands when allocating network resources. Consequently, it leads to low delay and to high network throughput. Simulations indicate that the proposed algorithm has substantial improvement in terms of the average weighted hop distance and network throughput in comparison with known algorithms such as HLDA, MMHA and MRU.

1. Introduction

Wavelength-division multiplexing (WDM) networks are considered as a promising technology for the next generation wide area networks because of its reconfigurability and plentiful bandwidth [3]. WDM networks set up lightpaths dynamically by reconfiguring the optical switches and can provide single hop communication channels between end nodes. This eliminates the electronic processing at intermediate nodes along the path and significantly reduces delay. However, it is generally impossible to provide single hop connectivity between each pair of end nodes due to limited number of router interfaces and other scalability issues. Consequently, it is necessary to create multi-hop paths and have electronic switching over multiple lightpaths for traffic between some source and destination pairs [1,8].

The logical topology describes the lightpaths between the electronic routers and is determined by the configuration of the OXCs, and transmitters and receivers at each node. Traffic grooming maps the traffic demands onto the logical topology [1,9]. Many heuristic algorithms for logical topology design suggest minimizing costs such as weighted hop distance [3,6,8] and maximum link utilization [7,8,10,11]. These algorithms deal with single-hop connection setup between source and destination pairs. Also there were some approaches to provide multi-hop connection by branch exchanges after logical topology design [10,11]. The problem of setting up optimal multi-

hop paths during the logical topology design is NP complete. Instead of considering logical topology design and traffic grooming together, these algorithms create a logical topology and leave grooming of traffic onto multi-hop paths to routing policy at a higher layer such as IP or MPLS [6,8].

However, isolation of the traffic-grooming problem from the logical topology design problem has some significant drawbacks. First, it is more optimal to assign high traffic demands to shorter multi-hop paths rather than assigning low traffic demands to single hop paths. This observation is not taken into account in algorithms that focus only on the creation of single hop paths. Second, lightpaths are established considering the end-to-end source-destination pair. So, there is no way to establish lightpaths for multi-hop traffic from/to intermediate nodes even though some interfaces and resources are available [6]. This results in inefficient network resource utilization. Third, some traffic demands could be blocked due to bandwidth shortage even though some optical resources are available, resulting in low network throughput. This phenomenon occurs since the logical topology design does not consider establishing multi-hop logical paths if needed, hence does not guarantee the accommodation of all traffic demands even though enough resources may be available.

In this work, we propose an integrated logical topology design and traffic-grooming algorithm. The algorithm supports lightpath setup for multi-hop as well as single hop paths. This is enabled by the consideration of the traffic grooming for multi-hop paths together with the logical topology design. We have used simulations with randomly generated traffic demand matrices to evaluate the performance of our algorithm in comparison with well-known algorithms such as HLDA, MMHA and MRU. The results indicate that our algorithm performs better in terms

[#] Research partially supported by the Laboratory for Telecommunications Sciences under contract MDA 90499C2521.

of average weighted (logical) hop count and network throughput. In terms of average number of wavelengths per lightpath, its optical resource usage is similar to MMHA, more efficient than HLDA, but not as efficient as MRU.

This paper is organized as follows. Section 2 reviews related work for logical topology design and traffic grooming. Section 3 formulates the problem as objective functions for logical topology design and traffic grooming. We propose an integrated logical topology design and traffic-grooming algorithm in Section 4. And, in Section 5, we analyze the performance of the algorithm using various metrics and compare the performance to that of other proposed schemes. Finally, we conclude the paper in Section 6.

2. Related Work

The logical topology design and traffic-grooming problem has been studied extensively so far.

[3] proposes several heuristic logical topology design algorithms for optical networks. The primary goal of these algorithms is to construct logical topologies in order to maximize the single hop traffic. The algorithms are categorized as traffic dependent and traffic independent algorithms. The traffic dependent algorithm establishes lightpaths between nodes in the descending order of traffic demands. The traffic independent algorithm establishes lightpaths either randomly or by exploiting the physical (optical) connectivity.

[4] proposes Maximize Multi-Hop Traffic Algorithm (MMHA), a traffic dependent logical topology design algorithm which aims at optimizing the logical hop count for traffic requiring multi-hop path setup. Since the delay is mainly affected by the electronic processing in the intermediate routers, it tries to minimize delay by reducing the logical hop count. Traffic demands weighted by the logical hop count (relative to the incomplete logical topology) are sorted in descending order and lightpaths are established in that order. However, the algorithm establishes end-to-end single hop lightpaths between source and destination pairs and not multi-hop paths.

[6] proposes three traffic grooming methods. It constructs an initial logical topology with minimum spanning tree and then maps the traffic onto the incomplete logical topology. The strategy for the traffic grooming is to use SPF (Shortest Path First) or OMP (Optimized Multipath). The choice is left for IP routing or MPLS path setup policy. After mapping the traffic onto the initial topology, it establishes additional lightpaths using weighted residual traffic demands and cost functions. The design strategies are very similar to the ones proposed in [3]. But, the fundamental distinction among the algorithms is in the initialization of the logical topology.

[1] proposes traffic grooming methods after the logical topology design. The algorithms use the same cost function, involving hop count (MST) and resource utilization factor (MRU), for the logical topology design and traffic grooming. Using these cost functions, the algorithms sort the weighted traffic demands in descending

order and set up lightpaths. Then, the residual traffic demands are mapped onto the current logical topology in the descending order of their bandwidth requirement.

[8,10] propose different way to design logical topology. [8] first builds full-meshed logical topology design and delete lightpaths with lowest link utilization until all constraints are satisfied. For [10], it constructs initial logical topology design and assigns flows onto the topology. After then, it selects m -lightpaths among n lightpaths and performs all possible branch exchange with it. Then, the traffics are re-mapped into the new topologies. And, the algorithm chooses a topology that has maximum objective function value.

3. General Problem Statement

The general problem we address is stated as follows [1,3,5,7,8]. We are given the following inputs to the problem.

- 1) Optical network (Physical) Topology: $G_o = (V, E_o)$ consisting of a weighted unidirectional graph, where V represents the set of integrated router-OXC nodes, and E_o represents the set of optical (physical) links.
- 2) Number of wavelength carried by each fiber.
- 3) The traffic demand is given by an $N \times N$ matrix, where $N=|V_o|$. Each entry (i,j) of the traffic matrix represents aggregated traffic amounts from source 'i' to destination 'j'.
- 4) The number of transmitter/receivers at each node 'i'.

The objective of the problem is to determine the following.

- 1) Logical (virtual) topology: $G_l=(V, E_l)$
- 2) A set of traffic groomed logical paths to minimize the delay and/or maximize total network throughput.

3.1 Logical Topology Design

In this section, we formulate the problem of logical topology design given the optical (physical) network topology. The logical topology is assumed to be a directed graph, hence potentially asymmetric. For the definition of the problem, we will use the following notations.

1) Notations

T_{sd}	traffic demand from source s to destination d .
OLP_{sd}^n	optical light path n connecting source s to destination d and provides routing information of optical path n .
D_{i}^T, D_{i}^R	the number of used transmitter and receiver interfaces at node i , respectively.
D_i^C	the degree constraints at node i .
C	channel capacity. It indicates the bandwidth associated with each wavelength.

2) Constraint

Equations (1) and (2) express the degree constraints for logical topology design. The number of lightpaths starting from or terminating at node i cannot exceed the number of transmitter/receiver interfaces at node i .

$$\sum_j OLP_{ji}^k = D_i^R \leq D_i^C, \forall i \quad \text{-----}(1)$$

$$\sum_j OLP_{ij}^k = D_i^T \leq D_i^C, \forall i \quad \text{-----}(2)$$

3.2 Traffic Grooming Problem

Traffic grooming deals with the issue of the traffic mapping onto the logical topology. So, it finds optimal path to reach the destination satisfying the constraints. This can be achieved in several ways. In this paper, we assume there is a single path satisfying the traffic demands and hence multi-path issues are not covered. The problem of the traffic grooming is stated as following.

1) Notations

- LP_{sd}^k a logical path k connecting from node s to node d consisting of a set of optical light paths.
 $= \{OLP_{sd_1}^{n_1}, OLP_{d_1d_2}^{n_2}, \dots, OLP_{d_m d}^{n_m}\}$
- D_{sd}^k delay for a logical path k . The delay includes propagation delay incurred in the optical network and electronic processing delay at each intermediate router.
- F_{sd}^k the bandwidth used (load) for a logical path k .

If the source and destination are clear, the subscript may be omitted.

2) Constraint

In this paper, we consider only one constraint related to logical topology. Equation (3) explains the constraint that the total traffic flowing through a lightpath (or a logical link) does not exceed the wavelength capacity C .

$$\sum_k F_{sd}^k \leq C \quad \text{-----}(3)$$

$$\forall k \ni OLP_{ij}^n \in LP_{sd}^k, \text{ for any } i, j \text{ and } \forall s, d$$

3.3 Objective Function

The goal of traffic grooming in this paper is to minimize the weighted delay and maximize the network throughput. The network throughput can be measured by

the blocked connection requests due to the lack of available bandwidth.

$$\text{Minimize: } \frac{\sum_k F^k \times D^k}{\sum_k F^k} \text{ and} \quad \text{-----}(4)$$

$$\sum_{s,d} T_{sd} - \sum_k F^k \quad \text{-----}(5)$$

Since the problem of finding the optimal solution for logical topology design and traffic grooming together is NP-complete, heuristic approaches are suggested for the solution. In the next section, we introduce a heuristic algorithm to optimize the objective functions.

4. Integrated Logical Topology Design and Traffic Grooming

In this section, we propose a heuristic logical topology and traffic-grooming algorithm for the optimization of multi-hop traffic. The basic idea of the algorithm is to set up multi-hop lightpaths by considering logical topology design and traffic grooming simultaneously. This is enabled by making use of optical and logical topology graphs respectively denoted as: $G_o = (V_o, E_o)$ and $G_l = (V_l, E_l)$. Although the logical topology graph is an output of the algorithm, the partial (incomplete) logical topology graph generated at each iteration is used for subsequent iterations in the algorithm. The pseudocode for the path computation is given in Algorithm 1. The algorithm consists of two steps. The first step is to set up lightpaths for single-hop traffic whenever possible, which is similar to that explained in [4]. The second part is to set up lightpaths for multi-hop traffic using the two graphs G_o and G_l . And, multi-hop lightpath setup is divided into two types: Forward and Backward. The two types are similar to each other except for the direction of light path setup.

1) Single hop lightpath setup

The first stage is to setup lightpaths and place logical edges between source and destination nodes. So, the traffic matrix is first sorted in descending order of traffic demand. Then, we choose the source-destination pair with maximum traffic demand and try to establish a single hop lightpath connecting that source-destination pair. The policy for lightpath setup is very similar to HLDA [3] except that HLDA establishes multiple connections with multiple paths for the same source-destination pair. If the degree constraints at the source and destination are not violated and wavelengths are available in the path to the destination, a lightpath is set up and the information is inserted into the logical topology graph. The bandwidth in the logical link is reserved to accommodate the traffic

demand.

2) Forward multi-hop lightpath setup

This situation arises when there are no free transmitter ports available at the source but free receiver ports available at the destination node. This setup methodology uses already created logical edges from the source node and establishes a multi-hop lightpath by creating a lightpath from one of the intermediate nodes to the destination. This is illustrated in Figure 1 (a). In this example, we assume that the degree constraints are two at each node. Consider a situation where the algorithm is trying to set up a lightpath from A to D, after setting up a lightpath from A to B. Since all transmitters are used for lightpaths from A to B and A to E, there is no way to set up additional lightpaths at node A using the physical topology. So, the algorithm looks at the logical edges to its neighbors and selects the one with maximum available bandwidth that can accommodate the current demand. In this example, the available bandwidth on the link AB is larger than the traffic demand from A to D, hence node B is selected and a lightpath setup procedure is applied from node B to node D. Since node B has one available transmitter, a lightpath is established from node B to D thus generating a two-hop path for the traffic from A to D. If the chosen neighbor does not have any free transmitter interfaces, the algorithm looks at other neighbors with enough available bandwidth and continues to try to setup logical edges to the destination. In the situation that none of the one-hop neighbors of the source have free transmitter ports, the algorithm continues this lightpath setup procedure by considering two-hop neighbors, three-hop neighbors, and so on until it finds a node from which it can set up a lightpath to the destination.

In the worst case, if no lightpaths could be set up for the current traffic demand, the traffic groomed logical path is provided by the algorithm using the final logical topology graph. In other words, accommodation of such a traffic demand is deferred until logical topology design is completed. Then the routing algorithm is used to determine if there is a (single) path in the logical topology that has sufficient residual capacity to accommodate the demand. Otherwise the demand is regarded as blocked.

3) Backward multi-hop lightpath setup

Backward multi-hop lightpath setup deals with the situation when free receiver ports at the destination node are not available but a free transmitter is available at the source. Here, a multi-hop lightpath is set up by establishing a lightpath from the source to the intermediate node. Let us consider a similar example presented earlier with a degree constraint of two per node. This is illustrated in Figure 1 (b). If a lightpath from A to D has to be set up, node D has no available receivers but node A has a free transmitter. In this case, the algorithm looks at the logical incoming neighbors of node D and

selects the one with the maximum available bandwidth that can accommodate the current demand. Nodes C and E are logical neighbors of node D. Since the available bandwidth on link C to D is larger than the current traffic demand, node C is selected for the lightpath setup procedure. And, a lightpath is set up from node A to C as there are free receivers available at node C. If the chosen neighbor does not have any free receiver interfaces, the algorithm continues as in the forward path setup procedure. In the worst case, if no lightpath can be setup for the current traffic demand, the traffic groomed logical path is provided by the algorithm using the final logical topology graph.

Algorithm 1. Integrated Logical Topology and Traffic Grooming Algorithm

```

Find  $s'$  and  $d'$  such that  $t_{s'd'} = \text{MAX } t_{ij}$  for all  $i, j$ 
If  $((D_{s'}^T < D_{s'}^C) \text{ AND } (D_{d'}^R < D_{d'}^C))$  then
    Find a path from  $s'$  to  $d'$  in  $G_o(V_o, E_o)$ 
    If (path found)
        Insert  $E_l'(s', d')$  into  $E_l'$ 
    endif
else if  $((D_{s'}^T = D_{s'}^C) \text{ AND } (D_{d'}^R < D_{d'}^C))$  then
    Put all  $V_x$  into neighbor list, where
         $t_{s'd'} \leq \text{AvailBW}(E_l'(s', x)), E_l'(s', x) \in E_l'$ 
    While (not path found OR neighbor list is not empty)
        Get a node  $y$  from neighbor list with minimum hop
        count and maximum available bw
        Check if a lightpath could be setup from  $y$  to  $d'$ 
        If (found) then
            Setup lightpath and Insert  $E_l'(y, d')$  into  $E_l'$ 
        else
            Put all  $V_x$  into neighbor list where
                 $t_{s'd'} \leq \text{AvailBW}(E_l'(y, x)), E_l'(y, x) \in E_l'$ 
        endif
    endwhile
else if  $((D_{s'}^T < D_{s'}^C) \text{ AND } (D_{d'}^R = D_{d'}^C))$  then
    Put  $V_x$  into neighbor list, where
         $t_{s'd'} \leq \text{AvailBW}(E_l'(x, d')), E_l'(x, d') \in E_l'$ 
    While (not path found OR neighbor list is not empty)
        Get a node  $y$  from neighbor list with minimum hop
        count and maximum available bw
        Check if a lightpath could be setup from  $s'$  to  $y$ 
        If (found) then
            Setup lightpath and Insert  $E_l'(s', y)$  into  $E_l'$ 
        else
            Put all all  $V_x$  into neighbor list where
                 $t_{s'd'} \leq \text{AvailBW}(E_l'(x, y)), E_l'(x, y) \in E_l'$ 
        endif
    endwhile
endif

```

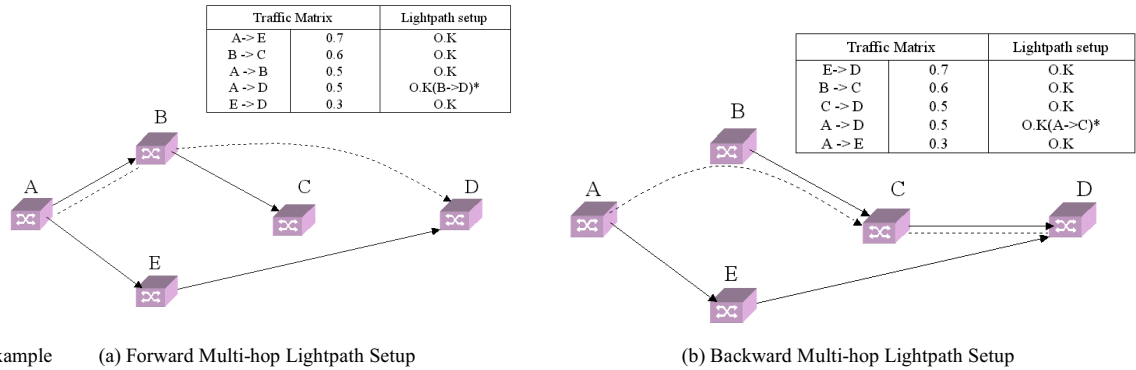


Figure 1. Example (a) Forward Multi-hop Lightpath Setup

(b) Backward Multi-hop Lightpath Setup

5. Experimental Analysis

5.1 Simulation Environment

We analyze the proposed integrated logical topology design and traffic-grooming scheme through simulations implemented in C language. We use 14-node NSFNet network topology for the simulation [6]. We assume that all nodes are working as OXC as well as router. Also, each node has capabilities to perform the wavelength conversion and there is no wavelength continuity problem. Each node has same number of transmitters and receivers. In our simulations, each node has three transmitters and receivers. And, we assume that each link has unlimited number of wavelengths. So, lightpaths can always be set up if the degree constraints are not violated. Each wavelength corresponds to one bandwidth unit (BU) in our model.

Each entry in the traffic matrix represents the aggregated traffic demands of a source-destination pair. It is generated randomly using uniform distribution between 0 and 0.5 BU for each source and destination pair. For the analysis, we used 10 different traffic matrices in our experiments.

We compare the performance of the proposed scheme with other logical topology design and traffic grooming algorithms. The algorithms used for comparison in our simulations are HLDA [3], MRU [1], and MMHA [4]. And, the shortest path routing is applied to all algorithms in optical networks and constraint-based shortest path routing (CSPF) in the virtual (IP/MPLS) networks. In other words, each lightpath is routed on a path of minimum optical hops that has a wavelength available on each optical link. Also, each traffic demand that cannot be accommodated via multi-hop lightpath set up is routed on a path of minimum logical hops that has sufficient residual capacity. If no such path is available, that demand is blocked.

5.2 Analysis

To evaluate the performance of each algorithm, we used average weighted (logical) hop count and network throughput as metrics. We present the performance of each algorithm with respect to these metrics.

1) Average Weighted Hop Count

Assuming that delay is mainly due to the electronic processing at the intermediate nodes, we measure the delay in terms of average weighted hop count. This can be easily computed by replacing delay with hop count in Equation 4. Figure 2 shows the average hop count value in the case of each algorithm for different traffic demand matrices. As shown in the figure, the proposed integrated algorithm has lower delay in every instance. This confirms that providing shorter multi-hop paths to higher traffic demand is better than providing single hop paths to lower traffic demands.

In our simulations, MRU and HLDA show behavior that is very sensitive to variation in the traffic matrix. Also, we observed that MMHA has a higher value (poorer performance) than other algorithms in this metric. The cost function of MMHA is dependent on the product of hop count and traffic demand. If there is a low demand that has a very high current hop count, its ingress-egress pair will be selected for creation of a one-hop path. This may be at the expense of a much higher demand that has a moderate current hop count and apparently leads to the poorer behavior of MMHA for this metric. In contrast, our algorithm would seek to find a shorter path for the higher demand. If a one-hop path was not possible, then our algorithm might create a two-hop path.

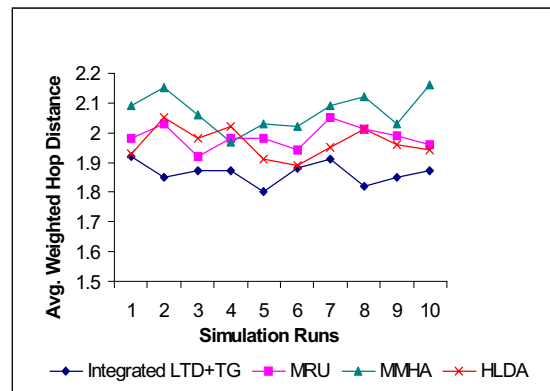


Figure 2. Average Weighted Hop Count

2) Network Throughput

We also measure network throughput in terms of the traffic rate accommodated by each logical topology and traffic-grooming scheme. As shown in Figure 3, our proposed algorithm accommodates more traffic demands than other algorithms. Since other algorithms are focused on the optimization of single hop traffic, they do not consider the traffic grooming effects of multi-hop traffic in conjunction with the logical topology design. So, some high traffic demands that require multi-hop paths cannot get sufficient bandwidth and are therefore blocked. However, the proposed algorithm takes account of traffic grooming effects all the time and assigns network resources by setting up (typically short) multi-hop paths if necessary, thus servicing more traffic requests to the network.

One interesting observation is that MMHA shows next best throughput in our experiments even though it has larger weighted hop distance. The reason is that other algorithms reduce average number of hops by providing single hop paths to higher traffic demands. And then, shortest path algorithm is used for traffic grooming. This might cause high link utilization on some links and prevent the network from being able to provide enough bandwidth for multi-hop traffic. However, The fact that MMHA considers not only traffic demands but also hop count in its cost function plays a beneficial role of distributing traffic among logical links. So, multi-hop traffic demands could go through underutilized links.

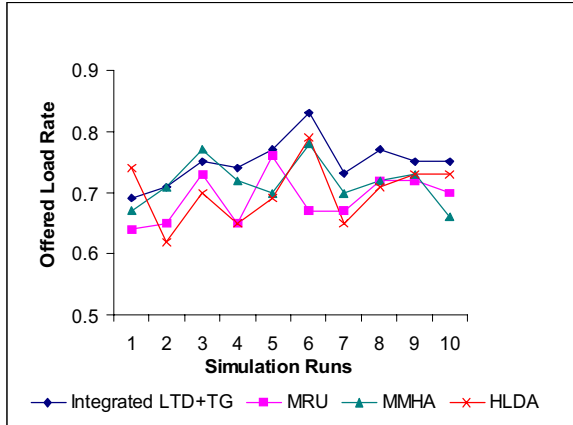


Figure 3. Network Throughput

6. Conclusion

In this paper, we propose an integrated logical topology design and traffic-grooming algorithm. Our approach is to set up lightpaths for multi-hop traffic as well as for single hop traffic, and to determine logical paths during the logical topology design. By creating relatively short paths for high demand multi-hop traffic, the algorithm provides very good performance in terms of the delay and network throughput. This was illustrated by

simulation analysis.

The experimental results are mainly focused on the analysis of the logical topology design and traffic grooming scheme with unlimited optical network resources--i.e. the only constraint is on the number of router interfaces. However, there are many constraints in real networks such as number of wavelengths and wavelength conversion. So, simulation analysis should be done so as to understand the behavior of the proposed scheme in real networks. Also, we assume that all traffic demands are satisfied with one path, or the demands are blocked even though there is some fraction of bandwidth available. This residual bandwidth can be utilized by traffic grooming with multi-path. Our future work will address these important issues.

References

- [1] K. Zhu, B. Mukherjee, "Traffic Grooming in an Optical WDM Mesh Networks", IEEE JSAC, pp. 122-133, Jan. 2002.
- [2] M. Kodialam, T. V. Lakshman, "Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks", IEEE Infocom, pp. 358-366, 2001.
- [3] R. Ramaswami, K. N. Sivarajan, "Design of Logical Topologies for Wavelength-Routed optical Networks", IEEE JSAC, pp. 840-851, Jun. 1996.
- [4] B. Mukherjee, Optical Communication Networks, New York, McGraw-Hill, 1997.
- [5] D. Banerjee, B. Mukherjee, "Wavelength-routed Optical Networks: Linear Formulation, Resource Budgeting Tradeoffs and a Reconfiguration Study", IEEE/ACM Trans. Networking, vol. 8, pp. 598-607, Oct. 2000.
- [6] K. H. Liu, C. Liu, J. L. Pastor, A. Roy, J. Y. Wei, "Performance and Testbed Study of Topology Reconfiguration in IP over WDM", IEEE Transactions on Communications, in press, 2002.
- [7] R. Dutta, G. N. Rouskas, "A Survey of Virtual Topology Design Algorithms for Wavelength Routed optical Networks", Optical Network Magazine, Jan. 2000.
- [8] E. Leonardi, M. Mellia, M. A. Marsan, "Algorithms for the Logical Topology Design in WDM All-Optical Networks", Optical Networks Magazine, pp. 35- 46. Jan. 2000.
- [9] A. L. Chiu, E. H. Modiano, "Traffic Grooming Algorithms for Reducing Electronic Multiplexing Costs in WDM Ring Networks", Journal of lightwave Technology, vol. 18, no. 1, pp. 2-12, Jan. 2000.
- [10] J-F, P. Labourdette, A.S. Acampora, "Logically Rearrangeable Lightwave Networks", IEEE Transactions on Communications, pp. 1223-1230, Aug. 1991.
- [11] A. S. Acampora, G. W. Hart, J-F. P. Labourdette, "Branch-Exchange Sequences for Reconfiguration of Lightwave Networks", IEEE Transactions on Communications, Vol. 42, No. 10, pp. 2822 -2832, Oct. 1994.