

Single- and Multi-path Logical Topology Design and Traffic Grooming Algorithm in IP over WDM Networks[#]

Kwangil Lee, Mark Shayman

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

Abstract - In this paper we investigate single- and multi-path logical topology design and traffic grooming algorithms. Since the problem of the optimal logical topology design for all traffic demands is NP-complete, we design a logical topology by sequentially constructing the shortest path for one source-destination pair at a time. In the single path case, the demand for each source-destination pair must be routed on a single path. In the multi-path case, if it is not feasible to route a demand on a single path, the traffic demand is divided into multiple subdemands and a path is provided for each subdemand. Each path is a locally optimized path in the sense that there are no other paths with less hop count that may be constructed from existing links and newly created links. We propose heuristic algorithms for logical topology design and traffic grooming in both the single path and multi-path cases. The performance of these algorithms in terms of weighted hop count and throughput is evaluated using the GLASS/SSF simulator. The results indicate that a proposed single path algorithm outperforms other known algorithms in terms of weighted hop count and throughput. Also, we observed that multi-path algorithms improve network throughput and a multi-path algorithm with widest-shortest paths reduces the weighted hop count compared to single path algorithms.

Keywords- logical topology design, traffic grooming, lightpath, single hop, multi-hop

I. INTRODUCTION

Wavelength-division multiplexing (WDM) networks are considered as a promising technology for the next generation wide area networks because of their 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 have electronic switching over multiple lightpaths for traffic between some source and destination pairs [1,8].

Much research has been done since the early 90's on the logical topology design and traffic grooming problem. That research focused largely on the optimization of objective functions such as weighted hop distance [3,6,8] and maximum link utilization [7,8,10,11]. However, the problem of logical topology design and traffic grooming is known to be NP complete. So, many algorithms deal with direct (single-hop) connection setup between source and destination pairs using heuristic functions. And, traffic grooming for multi-hop traffic is typically left for routing policy at a higher layer such as IP or MPLS [6,8]. Even though there are some approaches to provide multi-hop connection by branch exchanges after logical topology design, branch exchanges are done with only some lightpaths, not all lightpaths [10,11]. Also, the possibility of grooming a traffic demand on multiple paths when there is insufficient capacity on a single path is rarely considered.

In this work, we first investigate single path heuristic algorithms that integrate logical topology design and traffic grooming for multi-hop traffic. The general structure of these algorithms is as follows: The source-destination pair traffic demands are ordered according to some criteria and considered sequentially. When a demand is considered, the algorithm makes the choice that is locally optimal in the sense that the demand is placed on a path that has the minimum possible number of logical hops considering all topologies that refine the partial topology existing when that demand is considered. Second, we propose two multi-path logical topology design algorithms. When a single path has insufficient capacity for a traffic demand, the algorithm partitions it into smaller traffic demands so they may be accommodated by the current network. And, those are considered recursively by the algorithm.

This paper is organized as follows. Section 2 reviews related work for logical topology design and traffic grooming. Section 3 gives notations and objective functions used in this paper. We describe the local optimization problem for logical topology design and traffic grooming, and propose two heuristic algorithms for each of the single-path and multi-path cases in Section 4. And, in Section 5, we analyze the performance of the algorithms using various metrics and compare the performance to that of other proposed schemes. Finally, we conclude the paper in Section 6.

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

II. RELATED WORK

[3,6] propose 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. After designing the logical topology, they map residual multi-hop traffic onto the logical topology. The fundamental distinction between [3] and [6] is in the initialization of the logical topology. Also, these algorithms deal with multiple lightpath setup with multi-path for a source and destination pair. But, they do not deal with the multi-path issue for sub-lightpath traffic demands.

[1,4,8] propose lightpath setup algorithms that consider either physical or logical hop count in the topology design. [1] uses physical hop count value for the computation of link utilization factor. Based on the link utilization factor, the lightpaths are setup based on interface availability in source and destination. And, [4,8] tries to minimize delay by providing direct lightpaths for source and destination pairs that have longer 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, these algorithms only consider the case when interfaces are available in both source and destination. Grooming of multi-hop traffic during logical topology design is not considered.

[8] proposes a lightpath deletion algorithm for logical topology design. The algorithm first builds a fully meshed logical topology and deletes the lightpaths with lowest link utilization until all constraints are satisfied. And, [10] constructs an initial logical topology and assigns flows onto the topology. After that, it re-configures some lightpaths by branch exchanges in order to maximize the objective functions. This algorithm takes an optimization strategy after the logical topology design, but it does not deal with the optimization of the initial logical topology.

III. LOGICAL TOPOLOGY AND TRAFFIC GROOMING PROBLEM

The objective of the problem is to determine the logical topology and path assignments so as to optimize the objective functions for given traffic demands. The general problem is stated in many papers using ILP (Integer Linear Programming) [1,3,5,7,8]. So, we define some basic notations and objective functions used in this paper.

A. Assumptions

The logical topology design describes the lightpath setup problem with constraints in optical networks. In our work we are mainly focused on the problem with one constraint, the number of electronic interfaces (degree), and other constraints are not considered. So, we will assume that sufficient wavelengths and wavelength converters are available so that whenever router interfaces are available at the end nodes, a lightpath can be setup--i.e., the routing and wavelength

assignment problem is always solvable. This assumption has been made elsewhere in the literature [3,6].

B. Notation

In this paper, we will use the following notation.

$G^o=(V,E^o)$	Optical network (physical) topology 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.
$G^l=(V,E^l)$	Logical (virtual) topology. E^l represents the set of logical links, i.e. lightpaths.
T	Traffic matrix T . Each entry t_{sd} represents the aggregated traffic demand from source s to destination d .
LP^z_{sd}	A logical path z connecting from node s to node d consisting of a sequence of optical lightpaths.
$Delay(z)$	Delay for a logical path z . The delay includes propagation delay incurred in the optical network and electronic processing delay at each intermediate router.
$BW(z)$	The bandwidth used (load) for a logical path z .

C. Objective Function

The goal of this paper is to minimize the weighted delay and maximize the network throughput as shown in Equations (1) and (2). Since delay is mainly due to the electronic processing at the intermediate nodes, the delay can be measured in terms of average weighted hop count by replacing delay with hop count. And, the network throughput can be measured by the total traffic amounts accommodated by the logical topology. This is computed by the summation of total bandwidth of each traffic groomed logical path.

$$\text{Minimize: } \frac{\sum_z BW(z) \times delay(z)}{\sum_z BW(z)} \quad \text{and} \quad \text{-----(1)}$$

$$\text{Maximize: } \sum_z BW(z), \forall z \quad \text{-----(2)}$$

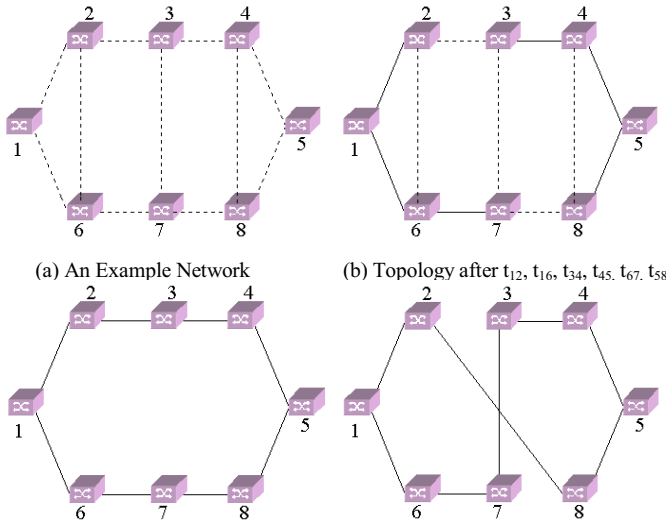
IV. HEURISTIC ALGORITHMS WITH LOCAL OPTIMIZATION

In this section, we propose single- and multi-path logical topology and traffic grooming algorithms so as to optimize our objective functions. The basic idea of the algorithms is to setup multi-hop lightpaths by considering logical topology design and traffic grooming simultaneously using a local optimization. This is enabled by making use of optical and logical topology graphs simultaneously denoted as $G^o = (V, E^o)$ and $G^l = (V, E^l)$ accordingly.

A. Basic Local Optimization Problem

The local optimization problem can be stated as follows: Given a partial logical topology with a set of traffic demands assigned to paths in this topology and traffic matrix, select a

traffic demand t_{sd} with higher priority in the traffic matrix and then find the shortest logical path from s to d with available bandwidth at least t_{sd} using either existing (logical) links (traffic grooming) or a combination of existing links and new links to be created (logical topology design). Our approach is to optimize our objective function by allocating the network resources, i.e., lightpaths and bandwidth, first to higher priority traffic demands. Let us consider a simple example explained in Figure 1. In this example, we assume that the number of interfaces at each node is two and lightpaths are bi-directional. We also assume that each existing link has sufficient residual bandwidth to accommodate the traffic demands being considered. Figure 1 (a) shows an example optical network and the dashed links represent optical links. Now, we consider the situation for providing paths for sorted traffic matrix $T = \{t_{12}, t_{16}, t_{34}, t_{45}, t_{67}, t_{58}, t_{18}, t_{25}, t_{78}, t_{23}\}$, where the demands are ordered according to some measure of priority.



(a) An Example Network (b) Topology after $t_{12}, t_{16}, t_{34}, t_{45}, t_{67}, t_{58}$
(c) Topology without local optimization (d) Topology with local optimization
Figure 1. Example

Let us consider how our algorithm works. If s has an available transmitter and d has an available receiver, then the shortest path is the one hop path obtained by creating a direct link from s to d (since we are assuming the optical network has resources to create such a lightpath). So, we can create direct links between the source node and destination nodes of the traffic demands $t_{12}, t_{16}, t_{34}, t_{45}, t_{67}$, and t_{58} as shown Figure 1 (b). When interfaces are available at both source and destination, there is no difference with other heuristic algorithms.

Now, consider the path provisioning for a traffic demand t_{18} . Since node 1 has no available interface, it is impossible to have a direct link between node 1 and node 8. Thus, the shortest path will be a multi-hop path. In other heuristic algorithms the paths for t_{18}, t_{25} cannot be determined at logical topology design but rather at traffic grooming since there is no available transmitter and receiver respectively. If traffic grooming decides those paths after two lightpaths are set up for t_{78}, t_{23} , then two paths for higher priority traffic demands have four hop paths, but lower priority traffic demands (t_{78}, t_{23}) have only one hop path as shown Figure 1 (c).

Multi-hop path provisioning makes the difference between our algorithm and other heuristic algorithms. Since a lightpath can be setup only between nodes that have available interfaces, the multi-hop lightpath setup problem can be defined as a node search problem as follows: We are given a partial (logical) topology with a set of traffic demands assigned to paths in this topology. We refer to the residual bandwidth of a link as the *available link bandwidth* (ALB). Let t be a given traffic demand. For each node x , let $H(s,x,t)$ denote the minimum hop distance from s to x considering only links with ALB at least t . Given s and t , let $f(s,t)$ denote the node x that minimizes $H(s,x,t)$ among those nodes that have available transmitters. In our example, $f(1,t_{18})$ is node 2 since $H(1,2,t_{18})=1$ and $H(1,7,t_{18})=2$. Similarly, let $f^R(d,t)$ denote the node y that minimizes $H(y,d,t)$ among those nodes that have available receivers. In the case of a tie in the selection of x (y), we choose the node x that maximizes the BW of the shortest path from s to x (y to d). For the demand t_{18} , $f^R(8,t_{18})$ is node 8. So, if we setup a new link between node 2 and 8, and then can provide a shortest path (1-2-8) with two hops.

Given a traffic demand t_{sd} from s to d , let $x = f(s,t_{sd})$ and let $y = f^R(d,t_{sd})$. We claim that the shortest path from s to d that has available bandwidth t_{sd} and includes at least one new link consists of the path from s to x with length $H(s,x,t_{sd})$, the newly created direct link from x to y , and the path from y to d with length $H(y,d,t_{sd})$. We denote this path by $P(s,x,y,d,t_{sd})$. To prove this, first note that this path is at least as short as any other path with available bandwidth t_{sd} that contains exactly one new link. However, if there is a path that contains more than one new link, we can shorten that path by establishing a direct link between the node at the head end of the first new link and the node at the tail end of the last new link in the path. So no path with more than one new link can be optimal. However, there may be a shorter path consisting entirely of existing logical links.

It follows from the preceding arguments that the locally optimal path from s to d for the demand t_{sd} is either $P(s,x,y,d,t_{sd})$ or a path that uses only links that already exist in the partial topology--i.e., a traffic groomed path. Given the partial topology with already assigned traffic demands as described above, let $LTD(s,d,t_{sd})$ denote the length of the "logical topology designed" path $P(s,x,y,d,t_{sd})$ that requires addition of one link and let $TG(s,d,t_{sd})$ denote the length of the shortest "traffic groomed" path from s to d --i.e., the shortest existing path having available bandwidth of at least t_{sd} . We define the Estimated Logical Hop Count (ELH) as

$$ELH(s,d,t_{sd}) = \min[LTD(s,d,t_{sd}), TG(s,d,t_{sd})]$$

It is an estimate for the optimized hop count for the source-destination pair s,d with demand t_{sd} . In above example, $TG(1,8,t_{18})$ is infinite because there is no path between node 1 and node 8 with existing links. So, $ELH(1,8,t_{18})$ is equal to $LTD(1,8,t_{18})$, two hops. Now consider the demand t_{25} . $LTD(2,5,t_{25})$ is six since the only available new lightpath is from node 3 to node 7, giving the path 2-1-6-7-3-4-5. However, $TG(2,5,t_{25})$ is two so $ELH(2,5,t_{25})$ is equal to $TG(2,5,t_{25})$. When $LTD(s,d,t_{sd})$ and $TG(s,d,t_{sd})$ is same, then either one is possible. In this example, we setup a lightpath from node 3 and 7 for the demands t_{78} and

t_{23} . Then, we can get the topology for our example as shown in Figure 1 (d). In this way we always provide more optimal paths to higher priority traffic demands. From the preceding analysis we have the following result.

Theorem 1. Given a partial topology with traffic demands assigned to paths, $ELH(s,d, t_{sd})$ is equal to the number of hops in the locally optimal path from s to d with bandwidth t_{sd} . Any path that may be constructed from existing links and newly created links has length at least $ELH(s,d, t_{sd})$.

B. Single Path Algorithms

To obtain a heuristic algorithm, we couple local path optimization with a rule that specifies the order in which source-destination pairs should be considered. We will see that the ordering of source-destination pairs has a significant impact on the effectiveness of the local optimization. Here, we propose two heuristic algorithms that differ in the way they order the source-destination pairs.

1) Maximum Traffic Demands (ELH-MTD)

A simple approach is to select at each step the source-destination pair with maximum traffic demand that has not yet been considered. In this approach, the traffic matrix is sorted in descending order and locally optimal paths are chosen sequentially. Whenever a path includes a link that does not already exist, that link is added to the partial logical topology.

[Algorithm 1] ELH with Maximum Traffic Demands

- Step 1** Find s' and d' such that $t_{s'd'} = \max[t_{sd}]$ for all s,d
Step 2 Compute a logical path $LP^z_{s'd'}$ for $s'-d'$ pair containing $ELH(s',d', t_{s'd'})$ hops
Step 3.1 $E^1 = E^1 \cup \{E^1_{x'y'}\}$ if $E^1_{x'y'} \in LP^z_{s'd'}$, and $E^1_{x'y'} \notin E^1$
Step 3.2 $T = T - \{t_{s'd'}\}$
Step 4 If T is empty, DONE
 Otherwise, go to Step 1

2) Maximum Resource Efficiency (ELH-REF)

When a traffic demand is assigned to a path, the efficiency with which it uses logical network resources depends on the number of (logical) hops; fewer hops means more efficiency. We propose an algorithm that uses a resource efficiency factor. This value is computed by the division of traffic demand by the ELH. At each step of the algorithm, we select the source and destination pair with the maximum value and either groom the traffic demand or setup a lightpath.

[Algorithm 2] ELH with Resource Efficiency Factor

- Step 1** Calculate $ELH(s,d, t_{sd})$ for all s,d
Step 2 Find s' and d' such that $t_{s'd'} / ELH(s',d', t_{s'd'}) = \max[t_{sd} / ELH(s,d, t_{sd})]$ for all s,d
Step 3 Compute a logical path $LP^z_{s'd'}$ for $s'-d'$ pair containing $ELH(s',d', t_{s'd'})$ hops
Step 4.1 $E^1 = E^1 \cup \{E^1_{x'y'}\}$ if $E^1_{x'y'} \in LP^z_{s'd'}$, and $E^1_{x'y'} \notin E^1$
Step 4.2 $T = T - \{t_{s'd'}\}$
Step 5 If T is empty, DONE
 Otherwise, go to Step 1

C. Multi-path Algorithms

Now we extend our basic topology design algorithm into multi-path. In this paper, we propose two multi-path logical topology design and traffic grooming algorithms that extend ELH-REF algorithm since the performance of ELH-REF is better than that of ELH-MTD as shown in Section 5.

1) Multi-path Algorithm with Widest Shortest Path (MPELH-WS)

The purpose of this algorithm is to provide multiple paths for a source-destination pair in order to minimize the number of hops. When the algorithm routes a traffic demand t_{sd} , it finds the shortest path from s to d with nonzero residual BW that can be constructed by adding one new lightpath. It also finds the shortest path with nonzero residual BW from s to d using existing lightpaths (the traffic groomed path). The shorter of these is chosen to route a portion of the demand. (Ties are broken by choosing the widest of the shortest paths--i.e., the one whose minimum link BW is maximum.) Let $ELH(s,d,*)$ denote the number of hops in the selected path. If the selected path requires adding a new lightpath, that lightpath is added to the logical topology. Let $B(s,d)$ be the BW of the selected path. If $B(s,d)$ is at least as great as t_{sd} , then the entire demand is routed on the selected path and t_{sd} is removed from the traffic matrix. Otherwise, if $B(s,d)$ is smaller than t_{sd} , then the portion $B(s,d)$ of the demand is routed on the path and t_{sd} is replaced by $t_{sd} - B(s,d)$ in the traffic matrix.

The algorithm routes traffic demands in decreasing order of a resource efficiency factor. The definition of REF given earlier is modified to be $B(s,d)/ELH(s,d,*)$ if $B(s,d)$ is less than t_{sd} . Otherwise, REF is given by $t_{sd}/ELH(s,d, t_{sd})$ as before. Note that $ELH(s,d,*)$ is the length of the shortest path with nonzero BW while $ELH(s,d, t_{sd})$ is the length of the shortest path with BW at least t_{sd} . If $B(s,d)$ is at least t_{sd} then the shortest path with nonzero BW has at least t_{sd} BW, so $ELH(s,d,*) = ELH(s,d, t_{sd})$. Thus, the definition of REF in both cases is given by $\min(B(s,d), t_{sd})/ELH(s,d,*)$. The multi-path ELH algorithm with widest shortest path is summarized in Algorithm 3.

[Algorithm 3] Multi-path Algorithm using ELH-REF

- Step 1** Compute REF value for all s,d
Step 2 Find s' and d' such that $REF(s',d') = \max[REF(s,d)]$ for all s,d
Step 3 Compute a logical path $LP^z_{s'd'}$ for $s'-d'$ pair containing ELH hops for s',d'
Step 4.1 $E^1 = E^1 \cup \{E^1_{x'y'}\}$ if $E^1_{x'y'} \in LP^z_{s'd'}$, and $E^1_{x'y'} \notin E^1$
Step 4.2.1 $T = T - \{t_{s'd'}\}$ if $t_{sd} \leq B(s,d)$
Step 4.2.2 $T = T - \{t_{s'd'}\} + \{t^{i+1}_{s'd'}\}$, where $t^{i+1}_{s'd'} = t_{s'd'} - B(s,d)$, otherwise
Step 5 If T is empty, DONE
 Otherwise, go to Step 1

2) Multi-path Algorithm with Shortest Widest Path (MPELH-SW)

The preceding algorithm (MPELH-WS) seeks to route each traffic demand over the shortest possible paths. Alternatively, we may seek to route each traffic demand over the fewest possible paths. To accomplish this, we should use the widest paths between the source and destination. When the algorithm

routes a traffic demand t_{sd} , it finds the path from s to d with maximum BW that can be constructed by adding one new lightpath. It also finds the maximum BW path from s to d using existing lightpaths (the traffic groomed path). The wider of the two paths is chosen to route a portion of the demand. (Ties are broken by choosing the shortest path.) The value of $ELH(s,d,*)$ is the number of hops in the selected path. $B(s,d)$ is the bandwidth of the selected path. The remainder of the description and statement of the algorithm is the same as that for MPELH-WS.

The algorithm requires the determination of the widest path from s to d that includes a new lightpath. Such a path can be constructed as follows: For each node x , let $AB(s,x)$ be the available BW of the widest path from s to x . Pick a node x that maximizes $AB(s,x)$ among those nodes that have a free transmitter. Also, pick a node y that maximizes $AB(y,d)$ among those nodes that have a free receiver. Construct a maximum BW path from s to d by choosing a maximum BW path from s to x , creating a new lightpath from x to y , and choosing a maximum BW path from y to d . Note that given any path from s to d that contains at least one new lightpath, there is a path from s to d containing exactly one new lightpath with at least as much BW. So the widest path from s to d containing at least one new lightpath contains exactly one new lightpath.

V. PERFORMANCE ANALYSIS

A. Simulation Environment

We analyze the proposed integrated logical topology design and traffic grooming schemes through simulations using GLASS/SSF simulator[12,13]. We consider a 16-node NSFNet network topology. We assume that all nodes have both OXC and router functionality. Also, each link has an unlimited number of wavelengths. So, lightpaths can always be set up if the degree constraints are not violated. In our simulations, each node has five transmitters and receivers. The capacity of each wavelength is normalized to one bandwidth unit (BU) in our model.

Each entry in the traffic matrix represents the aggregated traffic demand of a source-destination pair. It is generated independently using the uniform distribution between 0 and 0.5 BU. We used 15 randomly generated traffic matrices in our experiments. We compare our single path algorithms with other logical topology design algorithms such as MRU[1], HLDA [3]and DLPA [8]. DLPA is a logical topology algorithm which deletes low utilized lightpaths from an initial fully-meshed topology[8].

B. Analysis

1) Single Path Algorithm

We measured the weighted hop distance value and network throughput as performance metrics as shown in Figures 2 and 3. In the figures, ELH-REF algorithm works better than any other algorithm as measured by either weighted hop distance or network throughput. ELH-REF reduces the weighted hop distance 8 to 19% and average 13%. Also, it increases the network throughput 9 to 16.7% and average 12% compared to

other algorithms. This confirms that the resource efficiency factor in ELH-REF helps lightpaths be setup in order to maximize the network throughput as well as to provide shorter paths between nodes.

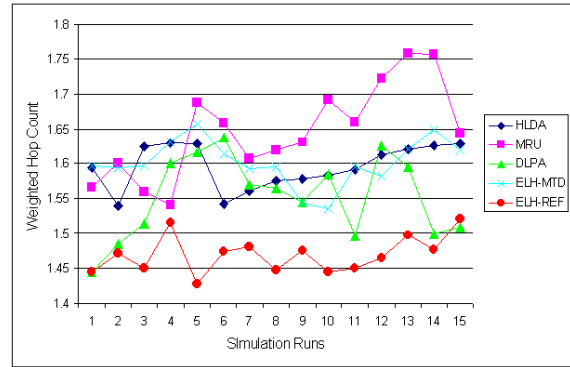


Figure 2. Weighted Hop Count – Single Path

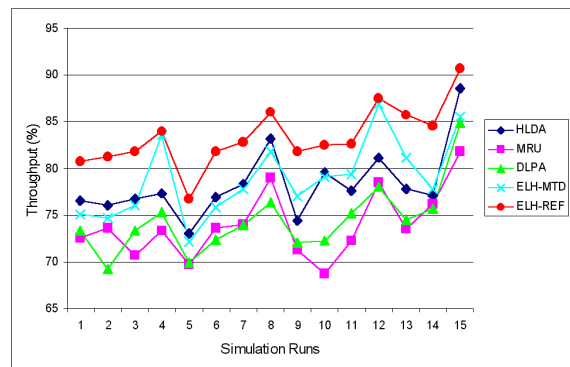


Figure 3. Network Throughput – Single Path

ELH-MTD does not show as good results as ELH-REF. Our observation is that some lightpaths established for multi-hop traffic in ELH-MTD are underutilized and make some logical paths longer. So the performance of the algorithm depends on the utilization of the multi-hop lightpaths. This confirms that the optimization of the multi-hop traffic is critical for the performance. Also, HLDA that maximizes single hop traffic shows similar behavior to ELH-MTD.

In our experiments, we found that the MRU algorithm showed poorer performance. While ELH-REF divides traffic demands by logical hop distance, MRU divides traffic demands by *physical* (optical) hop distance. This makes MRU effective at optimizing the use of optical layer resources but not especially effective at optimizing the performance metrics we considered.

Lightpath deletion approach such as DLPA shows lower weighted hop distance and network throughput. DLPA deletes lower utilized links until degree constraints are satisfied. During the deletion, lower traffic flows passing through deleted links are remapped into other links. Because the lower utilized links are deleted and remapped first, the higher traffic flows that are remapped later may be forced to take relatively longer

paths or be blocked if enough network resources are not available.

2) Multi-path Algorithm

First, we compare the performance of the multi-path algorithms with that of the best-performing single path algorithm, ELH-REF. As shown in Figure 5, network throughput is always increased about 0.5-5%. In the single path algorithm, a traffic demand that cannot be accommodated on any single path is blocked. In contrast, in the multi-path algorithms, it may be possible to route some or all of the demand by splitting it over several paths. Thus, increased throughput is to be expected.

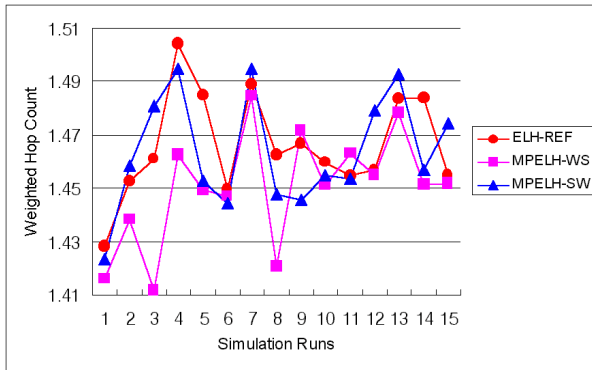


Figure 4. Weighted Hop Count – Multi-Path

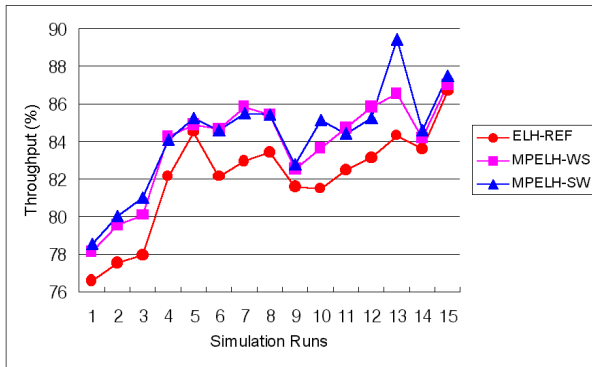


Figure 5. Network Throughput – Multi-Path

The results also show that MPELH-WS displays lower weighted hop count values compared to the single path algorithm. This is explained by noting that the multi-path algorithm can utilize short paths that do not have enough residual bandwidth to accommodate the entire traffic demand for a source-destination pair, and hence cannot be utilized by the single path algorithm.

Next, we compare the two multi-path algorithms. As shown in Figure 4, MPELH-SW shows lower hop count in most cases. This confirms that the shortest path algorithm contributes to minimizing the hop count. As for network throughput, MPELH-SW is slightly better than MPELH-WS. Since MPELH-SW is superior in the hop count metric and there is very little difference in the throughput metric, we conclude that MPELH-SW shows better performance than MPELH-SW.

VI. CONCLUSION

In this paper, we describe the local optimization problem for logical topology design and traffic grooming. Because the consideration of all traffic demands in the logical topology design is NP-complete, we design the logical topology so as to provide an optimal path for one source and destination pair at a time. The optimal path is computed by considering logical topology and traffic grooming together. The length of the locally optimal path is called Estimated Logical Hop Count (ELH). And, we propose two heuristic algorithms using ELH: ELH-MTD and ELH-REF. Also, we propose multi-path algorithms using widest-shortest path and shortest-widest path by extending ELH-REF algorithm. We perform simulation analysis using GLASS/SSF simulator. By the simulations, we observed that ELH-REF shows better performance in terms of delay and network throughput than other known single path algorithms. And, multi-path algorithms improve the network throughput and MPELH-SW reduces weighted hop count compared to single path algorithms. Finally, we conclude that the multi-path algorithm using widest-shortest path outperforms that of using shortest-widest path.

REFERENCE

- [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.
- [12] GLASS simulator, <http://dns.antd.nist.gov/glass/>
- [13] SSFnet, <http://www.ssfnet.org>