PAPER

Optical Network Design with Optical Constraints in IP/WDM Networks**

Kwangil LEE†* *and* Mark A. SHAYMAN†, *Nonmembers*

SUMMARY In this paper we consider algorithms for the logical topology design and traffic grooming problem in WDM networks with router interface constraints as well as optical constraints. The optical constraints include restricted transmission range due to optical impairments as well as limits on the number of available wavelengths. We formulate this problem as an integer linear program which is NP-complete. We then introduce heuristic algorithms which use a graphical modeling tool called the Virtual Neighbor Graph and add lightpaths sequentially. The best performing heuristic uses a so-called Resource Efficiency Factor to determine the order in which paths are provisioned for the traffic demands. By giving priority to demands that can be routed over paths that make efficient use of network resources, it is able to achieve good performance both in terms of weighted hop count and network throughput. For finding optimal multi-hop paths sequentially, we introduce interface constraint shortest path problem and solve it using minimum weight perfect matching.
key words: *logical topology, traffic grooming, optical impairments, Interface constraints, Resource Efficiency Factor*

1. Introduction

Wavelength-division multiplexing (WDM) networks are considered to be 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, there are several limitations precluding the establishment of lightpaths between every pair of nodes. First, there are a limited number of router interfaces. Second, lightpaths have a limitation on their physical extent due to various transmission impairments, e.g., attenuation, cross-talk, dispersion, and nonlinearities. Third, not all optical network elements are able to perform wavelength conversion. Consequently, it is necessary to have electronic switching over multiple lightpaths for traffic between some source and destination pairs [1], [10], [11].

Much research has been done on the logical topology design and traffic grooming problem. That research has focused largely on the formulation of the problem using Integer Linear programming (ILP) [1], [3], [5], [10] but the problem is known to be NP-complete. Consequently, many heuristic algorithms are proposed which typically relax some optical constraints so as to reduce complexity of the problem. However, most of these algorithms deal with only direct (single-hop) connection setup between source and destination pairs using heuristic functions, considering interface availability at the source and destination [1], [3], [4], [6], [7]. And, traffic grooming for multi-hop traffic is typically left for routing policy at a higher layer such as IP or MPLS [6], [7]. So, at the logical topology design stage it is very difficult to guarantee that traffic requiring more than one logical hop will be delivered efficiently to the destination. This becomes a greater challenge in optical network design when optical constraints are explicitly modeled since the optical constraints generally require that most traffic be multi-hop. Recently, some algorithms have been proposed considering optical constraints. But, those algorithms have relaxed other constraints in the logical topology design such as interface constraints [2], [10], [11]. However, for the optical design algorithms to be useful for real networks, they must take into account both optical constraints and interface constraints.

In this paper we investigate the optical network design problem with both optical constraints and interface constraints. Our contributions in this paper are as follows: We 1) formulate the problem using ILP(Integer Linear Programming) with optical impairments, 2) propose a simple and efficient graphical modeling method, called Virtual Neighbor Graph (VNG) to solve optical network design problems, especially for multi-hop optical networks, 3) propose an integrated approach to solve the logical topology design and traffic grooming problem simultaneously, 4) propose interface constraint shortest path algorithm using minimum weight perfect matching and 5) propose several heuristic algorithms for multi-hop optical network design using VNG.

This paper is organized as follows. Section 2 describes the optical network design problem using ILP. This includes logical topology design, traffic grooming, and routing and wavelength assignment problem. We introduce the graphical modeling method of Virtual

†The author is with the Department of Electrical and Computer Engineering at University of Maryland, College Park, MD 20742, USA
*Presently,the author is with the Department of Computer Science at Chunagnam National University, Daejeon, Korea
**This paper was presented at the Thirteen International Conference on Computer Communication and Networks, Oct. 2004

Neighbor Graph in Section 3. Also, we introduce interface constraint shortest path problem and propose the perfect matching algorithm in Section 4. And, we propose three logical topology design and traffic grooming algorithms using VNG in Section 5. In Section 6, 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 7.

2. Problem Definition

2.1 Network and Traffic Model

We consider a network connected by bidirectional optical links forming an arbitrary physical topology. However, lightpaths are not feasible between all nodes due to the optical impairments. For this, we assume the optical impairments information can be represented as the length of the linghtpath. And, each link has one fiber consisting of W wavelengths. Each optical node is equipped with an optical cross connect (OXC). We assume each OXC has no capability of wavelength conversion. So, when wavelength conversion as well as optical signal re-generation is required, two lightpaths are necessary. But our algorithms can be easily extended to include networks that include a mixture of nodes with and without wavelength conversion. Each OXC is connected to an edge device, e.g., IP router. For simplicity, we refer to the combination of OXC and IP router as a 'node' in this paper. Each node has a limited number of transmitters and receivers. And, traffic between nodes is symmetric and each traffic demand is less than the capacity of a wavelength. We are considering only a single path for each traffic demand. If there is not a single path that can accommodate a traffic demand, that traffic demand is considered blocked.

2.2 Problem Statement

The problems of optical network design are formulated as integer linear programs in this paper. The following will be given as input:

V represents a set of integrated router-OXC nodes.
E^o_{ij} Optical network topology variable. $E^o_{ij} = 1$ if an optical (physical) link connecting node i and node j exists in the network.
TR_i, RR_i The number of transmitters and receivers at node i respectively.
W The number of wavelengths on each optical link.
C The channel capacity. It indicates the bandwidth associated with each wavelength.
r Transmission range. The maximum distance that optical signals can propagate without any electrical conversion. May be specified as a distance or as a number of optical hops.
d_{ij} The length of an optical link E^o_{ij}. When the constraint on lightpath length is given in optical hops, then d_{ij} is 1 if $E^o_{ij} = 1$; ∞ otherwise.
T Traffic matrix. Each entry t_{sd} represents the aggregated traffic demand from source s to destination d.

The algorithm produces the following output:

E^l_{xy} Logical (virtual) topology binary variable. E^l_{xy} is 1 if a lightpath is set up between node x and node y; 0 otherwise.
Y^{xy}_{sd} Routing binary variable. Y^{xy}_{sd} is 1 if the path for t_{sd} passes through E^l_{xy}; 0 otherwise.
β_{sd} Blocking binary variable. β_{sd} is 1 if t_{sd} gets a route (with sufficient bandwidth) in the logical topology; 0 otherwise.
$w^{\lambda,xy}_{sd}$ Optical routing binary variable. $w^{\lambda,ij}_{xy}$ is 1 if E^l_{xy} passes through optical link E^o_{ij} using wavelength λ; 0 otherwise.

The constraints for the optical network design are given in Equation (1) through Equation (13).

$$\sum_{i,j} w^{\lambda,ij}_{xy} d_{ij} \le r, \forall \lambda, x, y \tag{1}$$

$$\sum_{i} w^{\lambda,ix}_{xy} = 0, \forall \lambda, x, y \tag{2}$$

$$\sum_{i} w^{\lambda,yi}_{xy} = 0, \forall \lambda, x, y \tag{3}$$

$$\sum_{i} w^{\lambda,iv}_{xy} = \sum_{j} w^{\lambda,vj}_{xy}, \forall \lambda, v \ne x, y \tag{4}$$

$$\sum_{\lambda} w^{\lambda,ij}_{xy} \le 1, \forall i, j, x, y \tag{5}$$

$$\sum_{y} E^l_{xy} \le TR_x, \forall x \tag{6}$$

$$\sum_{x} E^l_{xy} \le RR_y, \forall y \tag{7}$$

$$\sum_{s,d} t_{sd} Y^{xy}_{sd} \le C E^l_{xy}, \forall x, y \tag{8}$$

$$\sum_{v} Y^{sv}_{sd} = \beta_{sd}, \forall s, d \tag{9}$$

$$\sum_{v} Y^{vs}_{sd} = 0, \forall s, d \tag{10}$$

$$\sum_{v} Y^{dv}_{sd} = 0, \forall s, d \tag{11}$$

$$\sum_{v} Y^{vd}_{sd} = \beta_{sd}, \forall s, d \tag{12}$$

$$\sum_{i} Y^{iv}_{sd} = \sum_{j} Y^{vj}_{sd}, \forall v \ne s, d \tag{13}$$

- Equations (1) through (4) are concerned with optical network constraints. Equation (1) states that the length of a lightpath cannot exceed the transmission range. Equations (2)-(4) represent the wavelength continuity constraint, that the same wavelength is used on each optical link in a lightpath.
- Equation (5) states that each lightpath uses only a single wavelength.
- Equations (6) and (7) express the degree constraints for logical topology design. The number of lightpaths starting from or terminating at node x cannot exceed the number of transmitter/receiver interfaces at node x.
- Equations (8) give the network capacity constraints. It states that the total traffic passing through a logical link (lightpath) does not exceed the link capacity (a wavelength).
- Equation (9)-13) state that a logical path starts and ends with a logical link, and the number of incoming and outgoing logical links for a path at an intermediate node.

2.3 Objective Function

Let hop(s,d) denote the (logical) hop distance of the provisioned path for tsd. Then, we consider two metrics as our objective functions: the weighted hop count and the network throughput as shown in Equations (14) and (15), respectively.

$$\text{Minimize}: \sum_{s,d} t_{sd} hop(s,d) / \sum_{s,d} t_{sd} \tag{14}$$

$$\text{Maximize}: \sum_{s,d} t_{sd} \beta_{sd} \tag{15}$$

3. INTEGRATED TOPOLOGY CONTROL AND TRAFFIC GROOMING WITH VIRTUAL NEIGHBOR GRAPH

Since the ILP problem as described above is NP-complete, we turn to heuristic algorithms. In this section, we propose a simple graphical method for multi-hop optical network design, especially with optical constraints.

3.1 Virtual Neighbor Graph(VNG)

The nodes in the virtual neighbor graph correspond to the network nodes. Given the current state of the logical topology design and grooming process, the VNG contains the information needed to determine what logical paths and (bidirectional) lightpaths can be formed. Two nodes x,y in the VNG are connected by a link if either (1) there is already a lightpath between x and y, or (2) it is possible to construct a lightpath between them. This means that x and y both have available interfaces, are within the transmission range of each other, and the routing and wavelength assignment problem can be solved–i.e., there is a path from x to y and a wavelength that is available on all the optical links (the same wavelength). If x and y are connected by a link in the VNG, we refer to them as (virtual) neighbors. Even though multiple lightpaths are available, we allow only one potential logical link between two nodes at each time.

From the viewpoint of computational complexity, it is not practical to determine the solvability of the RWA problem for each pair of nodes in order to determine if they are neighbors. Consequently, we modify the definition of neighbors slightly. We choose a particular RWA algorithm and then say that x and y are neighbors if, in addition to the other conditions, the RWA algorithm produces a wavelength continuous path from x to y. With this change, a logical topology design and traffic grooming solution derived from the VNG will always be a feasible solution but it is possible that some feasible solutions may be missed. It should also be noted that the neighbor relationships will generally depend on the particular RWA algorithm selected. In this paper, we make use of the well-known RWA algorithm: shortest path and first-fit. Itfollows that finding neighbors while considering optical constraints takes $O(WN^2)$ time at each node. Usually, W is given as a constant, the complexity becomes $O(N^2)$.

We illustrate the virtual neighbor graph with an example. Figure 1 depicts a sample optical network. The nodes are connected with bidirectional optical links and each optical link has two wavelengths, λ_1 and λ_2. In this example, we assume the degree constraint is two. Also, the distance constraint is two (physical) hops. Then, we can construct the virtual neighbor graph as shown in Figure 2. As shown in Figure 2, the number of virtual (potential) neighbors of each node is four because each node is directly connected to two nodes and the distance constraint is two. If all nodes can be reached without any electrical conversion, the initial VNG will be full mesh graph. On the other hand, if the distance constraint is just one, then the VNG will be the same as the optical (physical) topology.

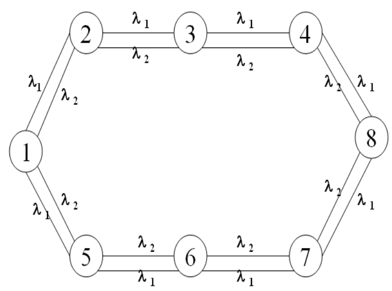


Fig. 1 Example Optical Network

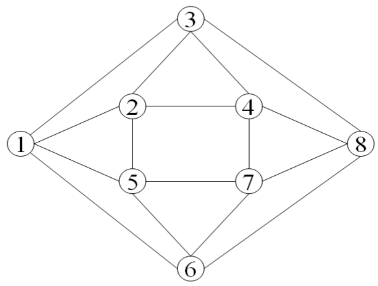


Fig. 2 Illustrated Virtual Neighbor Graph

3.2 Integrated Topology Design and Traffic Grooming

We use the VNG to do integrated topology design and traffic grooming. When a path is determined for a traffic demand, any virtual (potential) links are converted to actual links–i.e., the lightpaths are set up–and virtual links that violate the constraints are deleted. Each link in the VNG is weighted with available link bandwidth on each wavelength. If the available link bandwidth is not enough for the demand, then we replace an existing lightpath with a potential lightpath if it is available. Otherwise, it is eliminated. Our algorithm takes the following steps.

1. Initially, construct Virtual Neighbor Graph considering all optical constraints, (1) through (4).
2. A traffic demand is chosen based on some criteria and a shortest path is computed for the demand satisfying interface and bandwidth constraints. If none exists, then the demand is rejected.
3. If the path includes virtual links, then those links are marked as actual links.
4. The capacity of each link on the path in the virtual graph is updated (decreased) to incorporate the bandwidth allocated to the demand routed.
5. The VNG is updated by eliminating all the potential links that lead to the violation of the interface constraints (6) and (7) as well as of wavelength continuity constraints (2) through (4). (I.e., a virtual link is eliminated if the RWA algorithm does not find a wavelength continuous path between the pair of nodes.)
6. Steps 2,3,4 and 5 are repeated until all traffic demands are considered. This way, a logical topology is created from the VNG and all the routes for the traffic demands are computed.

Let us consider an example. We use the example network shown in Figure 1. In this example, we consider bidirectional traffic demands and each traffic demand is normalized by setting the capacity of each wavelength equal to one. And, we assume that there are four traffic demands sorted as follows: $T = \{(t_{48} = 0.45), (t_{25} = 0.4), (t_{34} = 0.35), (t_{18} = 0.3)\}$. We use constrained shortest path routing as a routing algorithm. Also, we assume that the wavelength for a lightpath is assigned randomly in this example. The shortest paths for three traffic demands (t_{48}, t_{25}, t_{34}) are easily computed using Figure 2. After three traffic demands are provisioned with direct links, we get the optical network and VNG as shown in Figures 3 and 4 accordingly. In Figure 3, the thick lines represent actual logical links (existing lightpaths) and the thin lines are optical links available for future lightpaths. In Figure 4, the weight of each link represents available bandwidth on each wavelength. Also, the available bandwidth on λ_2 between 3 and 4 and on λ_1 between 4 and 8 is 0 because of interface constraint at node 4.

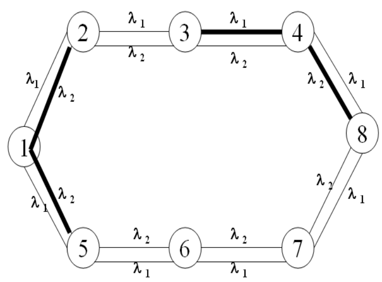


Fig. 3 Optical Network - t_{48}, t_{25}, t_{34}

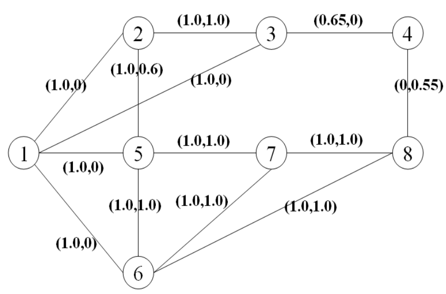


Fig. 4 Virtual Neighbor Graph - t_{48}, t_{25}, t_{34}

Also, the lightpaths setup for the three traffic demands require the virtual graph to be updated. Compared to Figure 2, we find that some nodes no longer have four neighbors: Node 4 has two neighbors, and nodes 2,3,7,8 each have three neighbors. Node 4 has only two neighbors because there are no free interfaces available so no additional lightpaths can be set up. Thus, nodes 2 and 7 are no longer virtual neighbors of node 4. As a result, nodes 2 and 7 now have only three neighbors. For nodes 3 and 8, they cannot be neighbors even though they have free interfaces and wavelengths are available. This is due to the fact that there is no single wavelength available along the path within distance constraints(two optical hops) and thus a lightpath from 3 to 8 would require wavelength conversion.

Now, we consider lightpath setup between nodes 1 and 8 using the updated VNG (Figure 4). We can find a shortest path (1-6-8) with two logical hops. And, we easily allocate resources along this path. This requires four optical hops and four wavelengths as shown in Figure 5. When we allocate wavelengths along the path, we get a VNG as shown in Figure 6. This process is continued until all traffic demands are either provisioned or blocked.

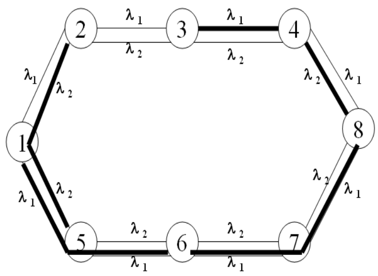


Fig. 5 Optical Network - $t_{48}, t_{25}, t_{34}, t_{18}$

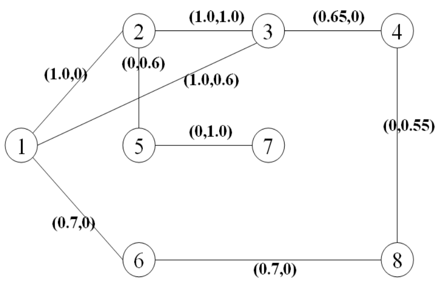
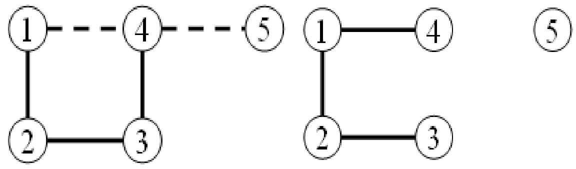


Fig. 6 Virtual Neighbor Graph - $t_{48}, t_{25}, t_{34}, t_{18}$

4. INTERFACE CONSTRAINT SHORTEST PATH PROBLEM

4.1 Interface constraint Shortest Path Problem

As explained in the previous example, the logical topology design and traffic grooming problem is converted into the constraint (interface and bandwidth) based shortest path finding problem for the traffic demand by using VNG. We call this problem an interface constraint shortest path problem. Usually, a constraint based shortest path algorithm can be obtained by modifying Dijkstra or Bellman-Ford algorithm. However, the interface constraint shortest path problem cannot be solved in this way. Let us consider the simple example in Figure 7. In this example, the interface constraint is two. And, solid links are actual links and dotted links are virtual links. All links are bi-directional and unweighted. We wish to find a shortest path in the graph consisting of actual links and virtual links. If the computed path includes virtual links, all virtual links are converted into actual links. This means that if the path contains a virtual link from node i to node j, i and j must each have a free interface. Let's compute a shortest path from node 1 to node 5 using Dijkstra's algorithm. Using Dijkstra's algorithm we can construct all shortest paths from node 1 other than node 5 as shown in Figure 7(b). As shown in the figure, the interface constraint rules out a path $1-4-5$ since node 4 has only one free interface but the path contains two virtual links. So, a path from node 1 to node 5 cannot be added to the shortest path tree. However, there exists a shortest path, $1-2-3-4-5$, in the graph. The problem is that the shortest path algorithms are based on the optimality principle, which states that each subpath of the optimal path is itself optimal [14]. However, an optimal sub-path from 1 to 4 is $1-4$, not $1-2-3-4$. Since this violates the optimality principle, using either the Dijkstra or Bellman-Ford algorithm cannot find the shortest path from node 1 to 5.



(a) Example Network (b) Shortest Path Tree

Fig. 7 Interface Constraint Shortest Path Problem

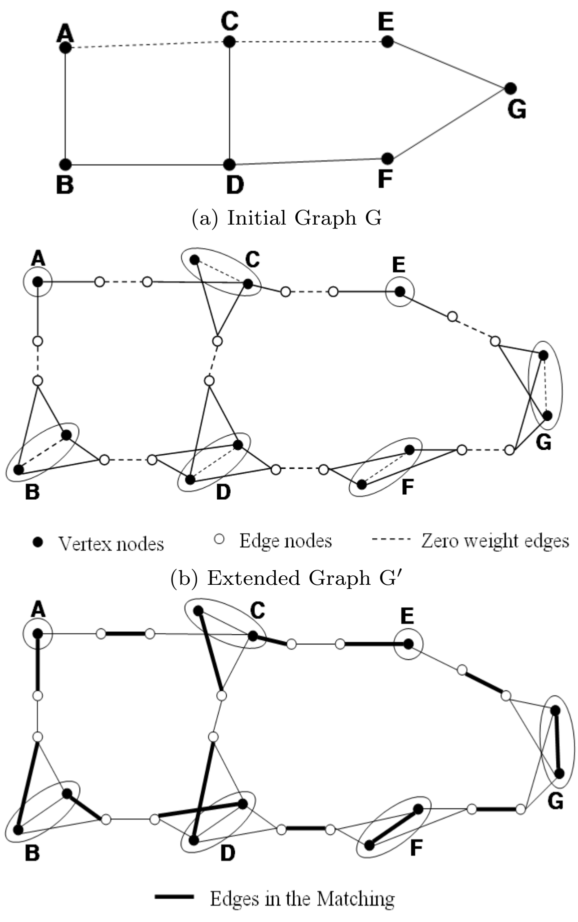
4.2 Minimum Weight Perfect Matching Algorithm

The interface constraint shortest path problem can be reduced as a minimum weight perfect matching problem. Then, the problem is formally described as follows: We have a network with two kinds of edges (links), 'actual' and 'virtual'. We refer to these edges as *green* and *red* edges respectively. Each node v has a constraint that at most p_v red edges can be incident on it from the path. If node v has $D(= TR_i, RR_i)$ green edges incident on it, then $p_v = 0$. When node v has $D-1$ green edges, then $p_v = 1$; otherwise, $p_v = 2$. We reduce the problem to the problem of computing a minimum weight perfect matching in a new graph G'. For each vertex $x \in V - \{v_s, v_t\}$ we create two nodes x_1 and x_2, and add a zero weight edge between them. We retain v_s and v_t as they are.

For each edge $e_{xy} \in E$ we create two new vertices $v_{e_{xy}}$ and $v_{e'_{xy}}$ (called edge nodes) and add a zero weight edge between them. We also add edges from $v_{e_{xy}}$ to x_1 and x_2, each of these has weight $w_{xy}/2$. We add edges from $v_{e'_{xy}}$ to y_1 and y_2, with each such edge having weight $w_{xy}/2$. Finally, for any red edges (u, x) and (x, y) with $p_x = 1$ we delete the edges from $v_{e'_{ux}}$ to x_1

and from $v_{e_{xy}}$ to x_1.

We illustrate an example for the proposed perfect matching algorithm. In the Figure 8, we wish to compute the shortest interface constraint path from A to E. All edges have unit weight. The interface constraint for node C is assumed to be two. Then, we can get an extended graph(Figure 8(b)) from the initial graph (Figure 8(a)) as explained. Notice that the shortest unconstrained path goes from A to C to E. However, this path uses two red edges in sequence. Adding both edges to the graph will cause node C to have interface three. The shortest constrained path starts at node A and goes node B, D, C and finally goes at E. The minimum weight perfect matching is also shown corresponding to this path in the Figure 8 (c). (We suggest to refer [15] for the details of the perfect matching algorithm.)



(a) Initial Graph G

(b) Extended Graph G′

(c) G′ after the perfect matching is applied

Fig. 8 Example : Perfect Matching Algorithm

Theorem 1: A minimum weight perfect matching in G' will yield a shortest path in G connecting v_s and v_t with the property that for any vertex v on the path, no more than p_v red edges are incident on v.

Proof Suppose we have a valid path p_{st} in G. There is a minimum weight perfect matching in G' of the same weight. For all vertices v that are not on the path, we can match v_1 and v_2 with zero weight. For all edges not on the path, we match the corresponding edge nodes with zero weight. For edges e_{xy} on the path, we do not match $v_{e_{xy}}$ and $v_{e'_{xy}}$ but instead match these nodes with x_i and y_j. The key point is that if $p_x = 1$ then only one red edge on the path may be incident to x. As a result, we can match the red edge with x_2 and the (adjacent) green edge with x_1.

To prove the converse, consider a minimum weight perfect matching in G' . Note that in this matching, nodes v_1 and v_2 that match together are not on the optimal path. Similarly edges e_{xy} and e'_{xy} that match together are not on the path. Construct a subgraph of G by "mapping" the minimum weight matching in G' to G, by merging the vertices v_1 and v_2. Note that each node in the subgraph, other than v_s and v_t, has degree exactly zero or two. Note that if $p_v = 1$ then only one red edge incident to v may be chosen in the subgraph, as only v_2 can match to a "red edge node" and not v_1. Thus we get a path that satisfies the interface constraints.

The complexity of the minimum weight perfect matching algorithm is $O(N^3)$ [15]. So, the logical topology design and traffic grooming for each traffic demand is the sum of the complexity for VNG and for CSPF computation, thus $O(N^3)$.

5. HEURISTIC ALGORITHMS

It is easily seen that the order in which the traffic demands are provisioned can have a significant effect on the result of the logical topology design and traffic grooming process. So, the decision regarding the sequencing of traffic demands is very important for the performance of the algorithm. We introduce and analyze three heuristic algorithms for this purpose.

5.1 Maximum Traffic Demands(MTD)

This is one of the simplest ways to order the traffic demands. The traffic demands are ordered in decreasing order of magnitude and the s-d pair with maximum traffic demand is considered first for logical topology design and traffic grooming. The complexity for each demand in this algorithm is only $O(N^3)$.

5.2 Maximum Network Resource Usage(MNR)

In this algorithm, at each step we find the traffic demand that requires maximum network resources and provision for that demand next. The network resources required for each demand is computed by multiplying the traffic demand by the number of logical hops that

Table 1 The Summary for the Complexity of the Algorithms

Algorithm	Complexity
HLDA	$O(MN^2)$
MRU	$O(M^2N^2)$
DLPA	$O(M^2N^2)$
MTD	$O(MN^3)$
MNR	$O(M^2N^3)$
REF	$O(M^2N^3)$

would be required if the demand were provisioned now. We refer to this quantity as the expected number of logical hops (ELH); it can be easily computed by the constrained shortest path algorithm in the VNG. Since an updated ELH must be computed for each traffic demand at each step, the complexity of one step of this algorithm -i.e., for provisioning one demand–is $O(MN^3)$. where M is the number of s-d pairs.

5.3 Resource Efficiency Factor(REF)

By allocating network resources efficiently, we can achieve higher network performance. The resource efficiency in topology design and traffic grooming is related with how many links are used by a flow. We define the resource efficiency factor (REF) to be the quotient obtained by dividing each traffic demand by the expected hop count, ELH. At each step we update the REF values of the remaining entries in the traffic matrix. Then we route the demand with the largest REF value. The complexity of this algorithm for each step is $O(MN^3)$.

5.4 Complexity Comparison

We compare the complexity of our algorithms with other well-known algorithms, HLDA (Heuristic Logical topology Design Algorithm) [1], [3], MRU (Maximum Resource Utilization) [1], DLPA (Deletition Light Path Algorithm) [7]. Our algorithms determine logical topology and traffic grooming simultaneously. But, these tasks are independent in the other algorithms. For comparison, we assume that shortest path routing policy is used for the purpose of traffic grooming for residual traffic demands after the logical topology design. Table 1 summarizes the complexity of the algorithms for the logical topology design and traffic grooming. As shown in Table 1, MNR and REF have higher complexity than the other algorithms.

6. Performance Analysis

6.1 Simulation Environment

We analyze the proposed integrated logical topology design and traffic grooming algorithms through simulations using the GLASS/SSF simulator [12], [13]. We consider a 16-node NSFNet network topology. We assume that each link has 20 wavelengths and there is only one optical link between nodes. And, each node has no wavelength conversion capability. So, if wavelength conversion is required, then setup of two lightpaths is necessary. In our simulations, each node has five transmitters and receivers. The capacity of each wavelength is normalized to one bandwidth unit (BU). 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. For the analysis, we used 10 randomly generated traffic matrices.

6.2 Analysis

For analysis, we run simulations with two different scenarios varying lightpath distance constraints. In the first scenario, we assume that the distance for a lightpath is limited to three optical hops. So, multiple lightpaths are required to reach nodes that are more than three optical hops away. However, other algorithms were proposed under the assumption that a lightpath can be always setup between any arbitrary two nodes. This means when a distance constraint is enforced and multi-hop lightpaths are required between two nodes, other algorithms cannot be directly applicable. So, we compared the performance of only three heuristic algorithms we proposed for this scenario. We measured the performance metrics of weighted hop count and network throughput as shown in Figures 9 and 10. In the figures, REF algorithm works better than any other algorithm as measured by both weighted hop count and network throughput. REF reduces the weighted hop count 10 to 25 % compared to MTD algorithm and 100% compared to MNR. Also, it increases the network throughput about 10% and 20-30% compared to MTD and MNR respectively. This confirms that the resource efficiency factor helps lightpaths be setup in order to maximize the network throughput as well as to provide shorter paths for more traffic.

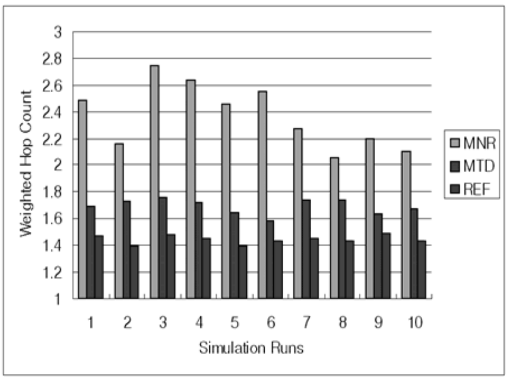


Fig. 9 Scenario 1 : Weighted Hop Count

In our simulations, the MTD algorithm shows

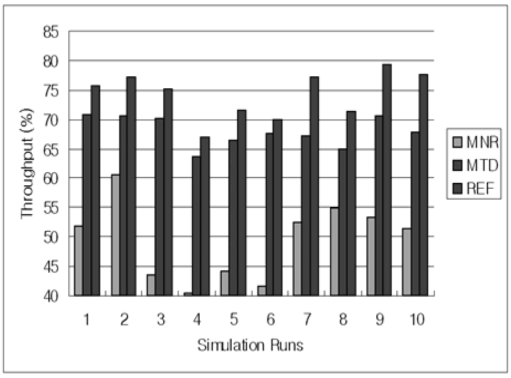


Fig. 10 Scenario 1 : Throughput

pretty good performance in the metrics even though it has poorer throughput than REF. Considering its lower complexity, MTD algorithm has reasonable performance. However, MNR algorithm shows worst performance. By giving higher priority to demands that use more network resources, the number of blocked demands is increased. Also, we run more simulations with different transmission range. We observed that the results are similar to those presented in this section.

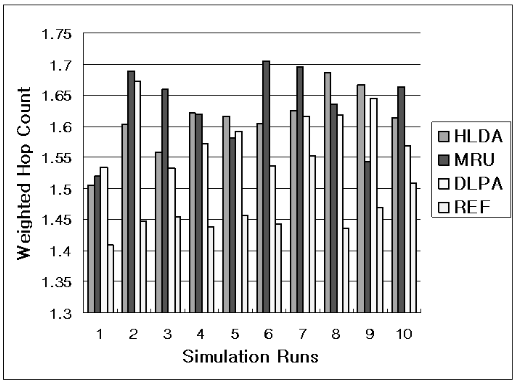


Fig. 11 Scenario 2 : Weighted Hop Count

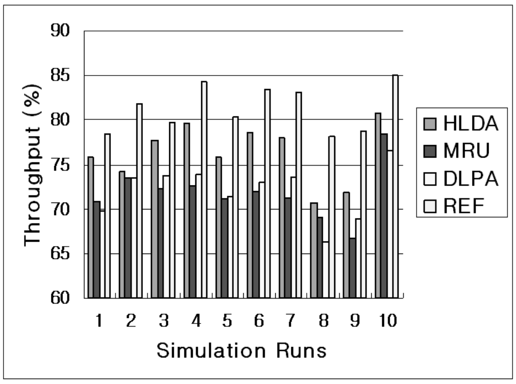


Fig. 12 Scenario 2 : Throughput

In the second scenario, we compare REF algorithm with other algorithms without lightpath distance constraints. This means that any node can be a virtual neighbor if there is any available interface and wavelength continuous path. The results are shown in Figure 11 and 12. As shown in the figures, REF algorithm reduces the weighted hop count from 8 to 15% compared to the other algorithms. Also, it increases the network throughput between 3 to 10%. This confirms that REF algorithm makes use of network resources efficiently and thus accommodates more traffic in the network. In our experiments, we found that the MRU algorithm showed poorer performance. While 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.

7. CONCLUSION

In this paper, we describe algorithms for integrated topology design and traffic grooming for optical networks that have both router interface constraints and optical constraints. The optical constraints include limitation on transmission range due to impairments as well as limitations on available wavelengths. We introduce a graphical modeling tool, the virtual neighbor graph, that permits the constrained shortest path algorithm to be used to determine paths for traffic demands as well as to determine which lightpaths to set up. Three algorithms are introduced which differ in the order in which traffic demands are considered. Of these, the REF algorithm shows the best performance in the simulations. At each step, this algorithm provisions a path for the demand that has the maximum resource efficiency factor.

Acknowledgements

We thank to Professor Samir Khuller at University of Maryland for his comments and proof on the perfect matching algorithm.

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] E. Leonardi, M. Mellia, M. A. Marsan, "Algorithms for the Logical Topology Design in WDM All-Optical Networks", Optical Networks Magazie, pp. 35- 46. Jan. 2000.
[8] J-F, P. Labourdette, A.S. Acampora, "Logically Rearrangeable Lightwave Networks", IEEE Transactions on Communications, pp. 1223-1230, Aug. 1991.
[9] 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.
[10] J. Q. Hu, B. Leida, "Traffic Grooming, Routing, and Wavelength Assignment in Optical Mesh Networks", Apr. 2002.
[11] L. Gouveia, P. Patricio, A. F. Sousa, R. Valadas, "MPLS over WDM Network Design with Packet Level QoS Constraints based on ILP Models", Infocom '2003, Apr. 2003
[12] GLASS simulator, http://dns.antd.nist.gov/glass/
[13] SSFnet, http://www.ssfnet.org
[14] Ravindra K. Ahuja , Thomas L. Magnanti, James B. Orlin, "Network Flows: Theory, Algorithms and Applications",Prentice Hall, 1993.
[15] W. Cook, A. Rohe, "Computing Minimum Weight Perfect Matchings", INFORMS Juornal of Computing, 1998.

Kwangil Lee Kwangil Lee received B.A., M.S, and Ph. D in Computer Science from Chungnam National University, Daejeon, Korea in 1993, 1996 and 2001 respectively. He joined National Institute of Standards and Technology as a guest researcher from 2000-2002. Since 2002, he worked as a research associate at Institute for System Research and Department of Electrical and Computer Engineering in University of Maryland. Currently, he is a Lecturer at Department of Computer Science in Chungnam National University.
He's research interests are in traffic engineering, QoS, routing, multicasting, optical networks, topology control, call admission control, MANET, mobile IP and network mobility.

Mark A. Shayman Mark A. Shayman received the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1981. He served on the Faculty of Washington University, St. Louis, MO, and the University of Maryland, College Park, where he is currently Professor of Electrical and Computer Engineering. His research interests are in communication networks, including MPLS, optical and wireless networks.
Dr. Shayman received the Donald P. Eckman Award from the American Automatic Control Council and the Presidential Young Investigator Award from the National Science Foundation. He has served as Associate Editor of the IEEE Transactions on Automatic Control.