Network Topology Control and Routing under Interface Constraints by Link Evaluation

Mehdi Kalantari
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Phone: 301 405 8841, Email: mehkalan@eng.umd.edu

Abhishek Kashyap
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Phone: 301 405 8843, Email: kashyap@eng.umd.edu

Kwangil Lee
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Phone: 301 405 8841, Email: kilee@eng.umd.edu

Mark Shayman
Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Phone: 301 405 3667, Email: shayman@eng.umd.edu

Abstract

In this paper we introduce a new approach for controlling the topology of networks under interface constraints, along with routing the given traffic demands. Our approach is applicable to the situations in which each network node can form links with several potential neighbors. We formulate the routing problem as a multi-commodity flow problem for maximizing the throughput. We extend the multi-commodity flow problem by introducing some non-linear constraints which represent the non-feasibility of having links which can violate interface constraints. The additional constraints lead to a kind of competition among the links which cannot co-exist. The links are chosen so as to maximize the network throughput. We provide a polynomial time algorithm by doing linear approximation of the non-linear constraints. We propose topology control algorithms which use the initial multi-commodity flow formulation of routing as well as algorithms which use the modified linear version of multi-commodity flow formulation. Our simulation results show the efficiency of our approach in choosing a proper set of links from the potential links that give a feasible and good network topology.

Keywords: Topology Control, Routing, Multi-Commodity Flow Problem, Linear Programming, and Linear Approximation.

This research was partially supported by AFOSR under grant F496200210217.

I. Introduction

Topology control is a key problem for both wireless and optical networks. In an ad hoc network with omnidirectional antennas, the problem is to determine how much the transmission range can be reduced to conserve energy and reduce interference while maintaining adequate connectivity. In a wireless network with point-to-point links (either free space optical or highly directional RF), the problem is somewhat different. Each node has a limited number of interfaces (transmitters and receivers), and the problem is to determine which links with potential neighbors to establish to obtain a topology that can best accommodate the traffic demands. In a reconfigurable wavelength division multiplexing (WDM) network, the problem is similar: given a limited number of interfaces on each router, determine which logical links (lightpaths) to establish that best meet the traffic requirements. In this paper, we focus on the topology control problem for wireless networks with point-to-point links and WDM networks. In both of these applications, the problem is to select an optimal topology from among the large number of feasible topologies that satisfy the interface constraints at each node.

There is a wealth of literature on the logical topology design and routing (or traffic grooming) in WDM networks [1], [2], [10]. A significant amount of research has been done on the formulation of the problem as a Mixed-Integer Linear Programming (MILP), but the problem is known to be NP-Hard [1], [3], [6], [11]. Consequently, many heuristic algorithms are proposed, which typically relax some constraints to reduce complexity [1], [3], [5], [7], [8], [9]. Authors of [9] propose a logical topology design algorithm by using a link deletion heuristic, which starts with a fully meshed logical topology and performs routing on it, and then it starts to delete the links with low utilization in order to find a feasible topology. In most of the proposed approaches, topology control problem and routing have been done independently, and the multihop traffic is routed after the topology is found [7], [8]. Heuristic Topology Design Algorithm (HLDA) is introduced in [2]. This heuristic tries to establish logical links (lightpaths) between pairs of nodes with the highest traffic demand.

In this paper we investigate the topology control and routing problem using an approach based on multi-commodity flow formulation. MILP is a simple scheme to formulate the problem, but is computationally prohibitive. As an alternative to MILP, we define the problem as a nonlinear optimization problem by converting integer constraints into nonlinear constraints. Then we convert the resulting nonlinear programming problem into a linear programming problem by linear approximation of the nonlinear constraints. Our approach gives a joint solution for the routing and topology control problems.

The rest of this paper is organized as follows: Section II describes the typical multi-commodity flow problem in the linear programming framework. Later in this section we give generalization of the linear multi-commodity flow problem to a nonlinear framework that gives optimal solution for the topology control. The nonlinear framework is achieved by adding a set of nonlinear constraints that take into account the fact that all potential links cannot co-exist in a feasible topology. Then we make approximations of the nonlinear constraints and make them linear to avoid the complexities of solving the nonlinear problem. In Section III we propose algorithms that use the generalized multi-commodity flow problem of Section II to extract proper feasible network topologies. Section IV illustrates the efficiency of the proposed algorithms through some examples and simulation results. The results show considerable increase in the network throughput by using the algorithms in Section III. Section V concludes the paper.

II. Multi-Commodity Flow Problem

In this section, we review the basic structure of the multi-commodity flow problem, and its extensions for making it suitable for our purpose. In Subsection A we give the standard form of the multi-commodity flow problem in a network with fixed topology. In this approach optimization of network performance is written as a linear programming problem. The performance measure is defined as achievable throughput for a given set of traffic demands. In Subsection B we modify the standard formulation by adding some nonlinear constraints to it for the purpose of taking into account interface constraints. The nonlinear programming problem can be solved by using some numerical methods, but in general it is hard to find its global optimum. In Subsection C, we introduce some techniques to make linear approximations of the nonlinear programming problem to fit it in the linear programming framework, which is much easier to solve.

A. Standard Multi-Commodity Flow Problem

Multi-commodity flow problem is a standard formulation of routing for maximizing the throughput of a network with fixed topology. Assume we have a network of N nodes and L unidirectional links. Furthermore, assume there are M sources, each corresponding to a destination. Let $S = \{S_1, S_2...S_M\}$ and $D = \{D_1, D_2...D_M\}$ denote the set of sources and destinations. The flows of sources can be routed in a multi-path scheme to their destinations. The multi-commodity flow problem deals with maximizing the sum of the flows of the different commodities, where each commodity is defined as the traffic generated by one source-destination pair. For formulating the problem, we use the following notations:

x_m^l: The amount of flow of commodity m that flows through link l.
I_n: The set of incoming links to node n.
O_n: The set of outgoing links from node n.
u_l: The total amount of flow on link l

Then we define a set of constraints known as flow conservation law. These constraints state that if a node is not the source or the destination of a commodity, then the amount of

incoming flow of the commodity to the node should be equal to the amount of outgoing flow of that commodity from the node. In other words, for every node n, and commodity m,

$$\sum_{l \in I_n} x_m^l = \sum_{l \in O_n} x_m^l, \; n \neq S_m, \text{ and } n \neq D_m. \tag{1}$$

Limitation of the link capacities incurs the following set of constraints:

$$u_l = \sum_{m=1}^{M} x_m^l \leq c_l \tag{2}$$

in which c_l is the capacity of link l. The objective of the problem is maximizing the sum of flows of commodities:

$$\text{Maximize } J = \sum_{m=1}^{M} f_m \tag{3}$$

in which f_m is the total flow of commodity m; this value can be written in terms of x_m^l variables in the following way:

$$f_m = \sum_{l \in O_{S_m}} x_m^l = \sum_{l \in I_{D_m}} x_m^l \tag{4}$$

The multicommodity flow problem can be summarized as:

$$\begin{array}{l} \text{Maximize } J = \sum_{m=1}^{M} f_m \\ \quad \text{Subject to:} \\ \sum_{l \in I_n} x_m^l = \sum_{l \in O_n} x_m^l, \; n \neq S_m, \text{ and } n \neq D_m \\ u_l = \sum_{m=1}^{M} x_m^l \\ u_l \leq c_l \\ x_m^l \geq 0 \end{array} \tag{5}$$

Since all the constraints and the objective function are linear in terms of variables, the above optimization problem is a linear programming problem which is essentially easy to solve. There are a variety of numerical methods that solve problems of this kind in polynomial time [12], [13]. The solution of the problem gives the optimal routes from the sources to their corresponding destinations, as well as the amount of flow that should be sent along each route. It should be noted that the above formulation of multi-commodity flow problem can handle the case in which a source sends its flow to several destinations (or a destination receives the flow of multiple sources) with minor modifications.

B. Multi-Commodity Flow Problem With Interface Constraints

The optimization problem given in the previous section is for the case in which the topology of the network is fixed. Now we consider the case in which the network topology is not fixed, and each node has a limited number of interfaces (transmitters or receivers). Each unidirectional link in the network uses one transmitter of its starting node, and one receiver of its ending node. The problem in this case is to establish a set of links from the set of potential links that gives the best network throughput. We define the potential links as the set of links that can be established or potentially exist in a feasible network topology. It can be shown that searching through all possible topologies needs an exhaustive search in a set of exponential cardinality, and it is numerically infeasible. The problem was proved NP-Hard in [4].

To do the formulation for this case, we start with a virtual network topology. In this virtual topology, a link exists between every pair of nodes that can potentially have a link. In other words, in the virtual topology all links are potential links. We use all the notations of the previous subsection for this virtual topology (i.e., L is now number of potential links, and c_l is the capacity of a potential link l). Additionally, for a potential link l we define a nonnegative slack variable and denote it by y_l; for link l we connect y_l to u_l by the following inequality:

$$u_l \leq c_l e^{-y_l} \tag{6}$$

The above inequality forces the link utilization u_l to be close to zero when the value of y_l is high enough, and when the value of y_l is close to zero, u_l can take the full capacity of the link. Now let R_n and T_n denote the number of transmitters and the number of receivers of node n respectively.

In order to satisfy the receiver constraint for node n we write the following set of inequalities.

$$\sum_{l \in \theta} y_l > Q \quad \forall \theta \subset I_n \quad C(\theta) = R_n + 1 \tag{7}$$

in which $C(\theta)$ is the cardinality of set θ, and Q is a sufficiently large positive real number. By inspecting equations (6) and (7), it can be seen that if Q is large enough, it is impossible that more than R_n incoming links of node n to have high utilizations. Equation (7) leads to competition among the links in the virtual topology that cannot co-exist. To make this point more clear, consider a case in which all $R_n + 1$ links that belong to a given $\theta \subset I_n$ have high utilizations. Then inequality (6) implies that the corresponding y_l values for all of these links are small, which results in a contradiction with inequality (7) since this inequality states that the sum of the values of y_l for the links in θ should be greater than a large number Q.

To take care of all the receiver constraints, we need to write inequalities of kind (7) for all possible $\theta \subset I_n$ with $C(\theta) = R_n + 1$, and for all nodes.

We take the transmitter constraints of the nodes into account in a similar way. In this case we write inequalities of the following form:

$$\sum_{l \in \theta} y_l > Q \quad \forall \theta \subset O_n \quad C(\theta) = T_n + 1 \tag{8}$$

The transmitter constraints at each node are written for all $\theta \subset O_n$ for which $C(\theta) = T_n + 1$.

Adding inequalities of (6), (7) and (8) to the linear programming problem of equation (5) gives the following non-linear

programming problem:

$$
\begin{aligned}
&\text{Maximize } J = \textstyle\sum_{m=1}^{M} f_m \\
&\quad \text{Subject to:} \\
&\textstyle\sum_{l\in I_n} x_m^l = \sum_{l\in O_n} x_m^l, \; n \neq S_m, \text{ and } n \neq D_m \\
&u_l = \textstyle\sum_{m=1}^{M} x_m^l \\
&u_l \leq c_l e^{-y_l} \\
&\textstyle\sum_{l\in\theta} y_l > Q \quad \forall\theta \subset I_n \quad C(\theta) = R_n + 1 \\
&\textstyle\sum_{l\in\theta} y_l > Q \quad \forall\theta \subset O_n \quad C(\theta) = T_n + 1 \\
&x_m^l \geq 0 \\
&y_l \geq 0
\end{aligned}
\tag{9}
$$

Mathematically, if the value of Q is infinity, the above optimization problem gives the optimal feasible topology; the optimal topology can be extracted by forming a network in which potential links are only formed if their corresponding value of u_l is nonzero. Obviously, such a topology is feasible, and it does not violate transmitter or receiver constraints.

One important fact about the optimization problem of (9) is that every feasible network topology corresponds to a point in the feasible set of the above nonlinear optimization problem. This can be shown by letting $y_l = Q$, $u_l = 0$ and $x_m^l = 0$ for every potential link that is not in that feasible topology, and letting $y_l = 0$, and u_l and x_m^l to take their corresponding values of the solution of optimization problem of (5) for the feasible topology. It can be verified that these values satisfy all constraints of optimization problem (9). Therefore, for a large enough Q this optimization problem gives the optimal routes and set of links for the feasible network.

To extract the optimal feasible topology from the solution of optimization problem (9), we need to eliminate links with low values of u_l. Generally, Q does not need to take a very large value. For example, if for a node with $R_n = 2$, then $Q = 10$ works well. In this case, for every three potential links l_1, l_2 and l_3 that arrive at node n, we have $y_{l_1} + y_{l_2} + y_{l_3} \geq 10$. This implies that at least for one of these links the value of y_l is greater that 3.3 which means the corresponding link utilization u_l is less that 4%. If we use such values for Q, some infeasible links will have non-zero utilization. To avoid this, after solving the nonlinear optimization problem, at every node where the degree constraints are violated, we tear down the potential links with lower utilizations.

C. Linear Approximation of Multi-Commodity Flow Formulation with Transmitter and Receiver Constraints

The problem of finding the optimal topology under interface constraints can be solved based on the solution of the nonlinear optimization problem proposed in the previous subsection. This nonlinear optimization problem can be solved by using numerical techniques. However, in general it is hard and expensive to find its global optimum. In this subsection we give a linear approximation of the nonlinear problem, considering the fact that linear optimization problems are very easy to solve.

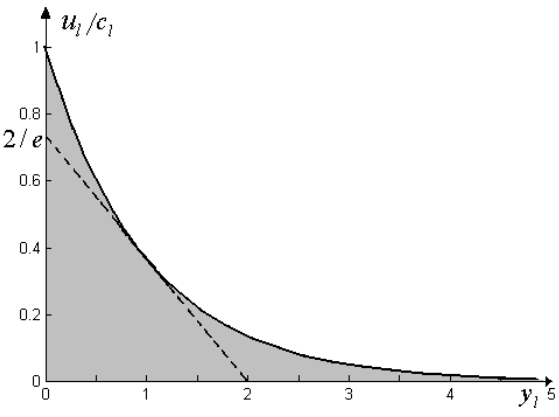

Fig. 1. Shaded area: the feasible set of $u_l \leq e^{-y_l}$. This feasible set is approximated by the area under the dashed line that can be expressed by a linear inequality.

The only source of nonlinearity comes from inequalities defined by (6). Our idea to define a linear approximation is to find a subset of the feasible set of these constraints that can be written in terms of linear constraints. For this purpose, we replace the feasible set defined by inequality $u_l \leq e^{-y_l}$ by its largest subset that can be expressed by a linear constraint. It can be shown that the biggest linear subset of the feasible set of this equation is the area under line defined by equation $eu_l/c_l + y_l = 2$. This approximation has been illustrated in Fig. 1. The shaded area in this figure shows the feasible set of $u_l \leq e^{-y_l}$. This area is approximated by the triangular area under the dashed line which is the feasible set of inequality $eu_l/c_l + y_l \leq 2$. By making this approximation, the feasible set becomes smaller, but the new set can be written by the following linear inequality:

$$eu_l/c_l + y_l \leq 2 \tag{10}$$

Now we recall constraints imposed by the receiver constraint at a node:

$$\sum_{l\in\theta} y_i > Q \quad \forall\theta \subset I_n \quad C(\theta) = R_n + 1$$

The inequalities of this kind cannot be satisfied for an arbitrarily large value of Q. This is because from inequality (10) the maximum value of y_l is 2, and so the maximum possible value for Q is $2(R_n + 1)$. However, if we let $Q = 2(R_n + 1)$, then $y_l = 2$ for all $l \in \theta$ which implies $u_l = 0$ for all $l \in \theta$ from inequality (10). In order to give some span to both y_l and u_l values, we suggest $Q = R_n$ for node n. So we will have the following set of constraints for the n^{th} node:

$$\sum_{l\in\theta} y_l > R_n \quad \forall\theta \subset I_n \quad C(\theta) = R_n + 1 \tag{11}$$

Having this limitation on the value of Q implies that we cannot have a strong competition among the links that cannot co-exist, but as a matter of fact, even lower values of Q lead to some competition among links that will be advantageous in finding

a good feasible topology.

Similarly, we define the following set of inequalities for taking into consideration the transmitter constraints of node n:

$$\sum_{l\in\theta} y_l > T_n \quad \forall\theta \subset I_n \quad C(\theta) = T_n + 1 \tag{12}$$

.

To illustrate the efficiency of the above competition scheme, assume for node n the receiver constraint is 2. Then for every three potential links such as links i, j and k we have $y_i + y_j + y_k > 2$. If one of these links, say link i, has zero or very low utilization (i.e., $u_i \approx 0$), then inequality $eu_i/c_i + y_i < 2$ implies that y_i can take a value close to 2, and inequality $y_i + y_j + y_k > 2$ is almost satisfied, which means y_j and y_k can take values close to zero. This implies that u_j and u_k are free to take larger values. On the other hand, if u_i takes a large value, then from inequality (10), y_i cannot take a large value, and it will be close to zero; this means that at least one of y_j or y_k should take a large enough value to satisfy $y_i+y_j+y_k > 2$. The large value of y_j or y_k pushes the corresponding value of u_j or u_k to zero. Therefore this scheme leads to competition among the links that cannot co-exist in the feasible topology.

The new optimization problem can be written as:

$$\begin{array}{l}
\text{Maximize } J = \sum_{m=1}^{M} f_m \\
\quad \text{Subject to:} \\
\sum_{l\in I_n} x_m^l = \sum_{l\in O_n} x_m^l,\ n \neq S_m, \text{ and } n \neq D_m \\
u_l = \sum_{m=1}^{M} x_m^l \\
eu_l/c_l + y_l \leq 2 \\
\sum_{l\in\theta} y_l > R_n \quad \forall\theta \subset I_n \quad C(\theta) = R_n + 1 \\
\sum_{l\in\theta} y_l > T_n \quad \forall\theta \subset O_n \quad C(\theta) = T_n + 1 \\
x_m^l \geq 0 \\
y_l \geq 0
\end{array} \tag{13}$$

III. Topology Control

In this section, we use the linear programming schemes in the previous section in order to find feasible network topologies. We introduce three algorithms for finding the feasible topology.

Algorithm 1: Link Deletion and Multi-Commodity Flow Problem

Our first algorithm is based on using the linear programming problem given by (5). We solve this problem for the virtual network of all potential links. Then we try to delete the links that violate degree constraints one by one. The links with lower utilization are deleted first. This algorithm can be written in the following way:

1- Solve the optimization problem of (5) on a network of all potential links. The output is the set of link utilization u_l, $1 \leq l \leq L$
2- Set Λ=all potential links
3- Repeat

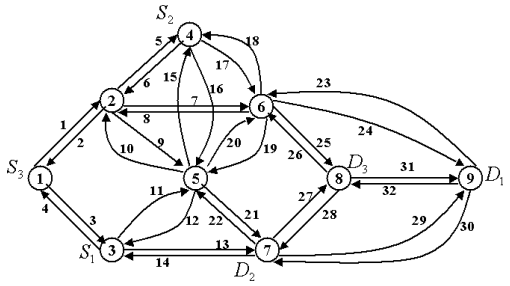

Fig. 2. The initial virtual graph.

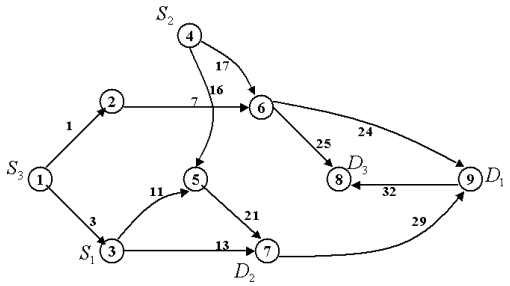

Fig. 3. The resulting topology of algorithm 1

4- V=Set of all links in Λ that violate interface constraints.
5- Let $l_1 = Argmin_l\{u_l \quad l \in V\}$
6- $\Lambda = \Lambda - \{l_1\}$
7- Until V =Empty Set.
After running this algorithm, Λ returns the set of links in the feasible topology.

Algorithm 2: Link Deletion Using Multi Commodity Flow Problem with Link Competition

In this algorithm, we use a similar algorithm as in the previous case. The only difference is that for this case we use the optimization problem given by (13) on the virtual network of all potential links at the start of algorithm.

Algorithm 3: Link Addition by Using Multi Commodity Flow Problem with Link Competition

In this algorithm, again we make use of the optimization problem given by (13), to find the values of u_l. In order to find a feasible topology, we start from a network with no links, and start adding links from the virtual topology to it. The links with the higher utilizations are added first. Let Ω denote the set of all potential links; then the algorithm can be written in the following way:

1- Solve the optimization problem of (13) on a network of all potential links. The output is the set of link utilization u_l, $1 \leq l \leq L$
2- Set Λ=Empty set
3- Repeat

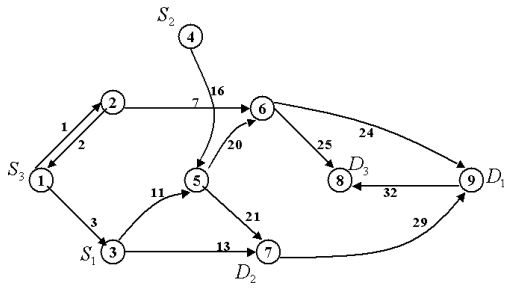

Fig. 4. The resulting topology of algorithm 2.

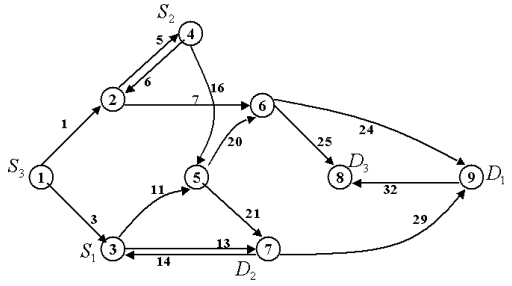

Fig. 5. The resulting topology of algorithm 3.

4- A= set of links $l \in (\Omega - \Lambda)$ such that the network with links $\Lambda \bigcup \{l\}$ does not have any link that violates the interface constraints.
5- Let $l_1 = Argmax_l\{u_l \quad l \in A\}$
6- $\Lambda = \Lambda \bigcup \{l_1\}$
7- Until A =Empty Set.
After running this algorithm, Λ returns the set of links in the feasible topology.

A very useful improvement of algorithms 2 and 3 can be done by running the optimization problem again after adding or deleting each link. Because in each iteration of the algorithm after adding or deleting a link, the set of potential links or the number of available interfaces change, by rerunning the optimization problem for the new set of potential links and interface constraint, the values of u_l are updated for the new set of potential links. To make this point clear, assume a node has transmitter constraint of 2, and at some iteration there are 2 outgoing links attached to this node in Λ. Therefore, no more transmitters are available, and the algorithm can continue by removing all other potential outgoing links of this node and rerunning the optimization for the new set of potential links.

IV. Numerical Examples and Simulations

In this section we demonstrate the performance of the algorithms that we proposed in the previous section. We start with a simple network example to illustrate our approach and then continue toward larger networks with a much higher number of potential links.

A. Simple Network Examples

Our first set of experiments is done on the virtual topology shown in Fig. 2. In this network, there are 9 nodes, and a set of 32 potential links that can be established in the network. The links and the nodes have been numbered as shown in the Fig. 2. All link capacities are 1 unit, and each node has 2 transmitters and 2 receivers. There are 3 source-destination pairs in this network. Nodes 3, 4 and 1 are three sources that correspond to the destinations 9, 7 and 8 respectively. The objective is to maximize the total throughput of the network. It can be shown that even for this simple network, finding the optimal network topology by exhaustive search needs inspecting about 10^8 topologies, which needs very heavy computation in practice.

Fig. 3 shows the resulting feasible topology by applying Algorithm 1. The total achieved throughput of the network is 3.0 units for this case. Fig. 4 shows the result of using Algorithm 2. The total achievable throughput for the topology generated by this algorithm increases to 4.0 units. Applying Algorithm 3 gives the same throughput with a slightly different network topology that is shown in Fig. 5. As it can be seen, link competition gives a considerable improvement in the network throughput, and results in more suitable topologies.

B. Larger Network Experiments

Our next set of experiments are on larger networks. We have simulated networks of 25 nodes, and each node can have up to 4 potential neighbors that are picked randomly at the start of simulation. Again each node has two transmitters and two receivers. There are 10 commodities corresponding to 10 source-destination pairs. The sources and destinations are chosen randomly. The link capacities are one unit. Table I shows the results for several simulation runs with different random seeds. As it can be seen in this table, Algorithms 2 and 3 give almost the same performance for most of the experiments. However, they perform much better than Algorithm 1. In all cases, using link competition has resulted in performance improvement. In some cases such as experiments 2 and 6 the improvement is more than 100 percent.

C. Comparison with HLDA

In this set of experiments, we compare the performance of our algorithms with the heuristic topology design algorithm (HLDA), which is known to give a very good performance in topology design. For this purpose, we consider a network with 14 nodes, and each node has 2 transmitters and 2 receivers, with 8 randomly generated potential neighbors. A node can send or receive from each of its potential neighbors. The capacity of each link is normalized to 1. We define seven source and destination pairs in the network, and for each pair, we

TABLE I

The comparison of the 3 Algorithms by different experiments: The numbers show the achievable throughput in the different simulation runs

Exp No.	Algo. 1	Algo. 2	Algo. 3
1	7.0	10.0	9.0
2	4.0	9.0	9.0
3	6.0	7.0	8.0
4	6.0	10.0	8.0
5	8.0	10.0	11.0
6	4.0	9.0	9.0
7	4.0	7.0	6.0
8	7.0	9.0	9.0

TABLE II

Performance comparison of the HLDA and our three algorithms in 10 simulation runs

Exp No.	HLDA	Algo. 1	Algo. 2	Algo. 3
1	8.0	8.0	9.5	9.5
2	8.5	8.0	9.0	9.0
3	8.4	8.6	9.6	10.3
4	8.5	8.3	8.9	9.1
5	8.0	8.6	9.3	8.9
6	9.0	9.7	11.4	10.7
7	8.1	8.0	9.7	10.6
8	8.8	8.1	9.3	9.9
9	9.5	8.8	10.4	9.8
10	8.6	9.0	10.6	11.4

define a random demand uniformly distributed between 0 and 3 units. For each simulation run, we use HLDA or one of our algorithms to design the topology, and after the design of the topology, we run the multi-commodity-flow problem for the resulting topology to find the optimal achievable throughput.

Table II shows the network throughput achieved by applying the algorithms in 10 simulation runs. As the table shows, algorithms 2 and 3 show better results than the other algorithms. The average throughput obtained using Algorithm 2 was 14.0% better than HLDA, and the average throughput obtained using Algorithm 3 was 15.8% better than HLDA. On the other hand, the average performance of Algorithm 1 is almost the same as HLDA. This confirms the benefit of using a multi-commodity flow formulation in which interface constraints are explicitly modelled by linearized constraints. On the other hand, if the MCF is used without explicitly modelling the interface constraints by linearized constraints, the resulting algorithm (Algorithm 1) offers no advantage over the existing algorithm HLDA.

V. Conclusion

In this paper we presented an approach for topology design in networks with interface constraints. Our approach is based on the multi-commodity flow formulation. We proposed three algorithms for finding good network topologies. The first algorithm is based on solving the multi-commodity flow problem for a virtual network of all potential links. The output of multi-commodity flow problem is the amount of utilization of potential links, which we use for evaluating them. The low utilization potential links that violate degree constraints are deleted to obtain a feasible topology.

One problem with the above algorithm is that different potential links that cannot co-exist in the feasible topology can take high utilizations independently. To improve this algorithm we added a new set of constraints that define dependency among the links that cannot co-exist and define competition among them. We gave a mathematical formulation for competition among links by adding some nonlinear constraints to the multi-commodity flow problem. Then we used a linear approximation of the nonlinear constraints to fit multi-commodity flow problem with link competition into a linear programming framework. Our simulations showed a considerable improvement in the network throughput by applying this approach.

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] A. Kashyap, S. Khuller, M. Shayman, "Topology Control and Routing over Wireless Optical Backbone Networks," *Proc. Conference on Information Sciences and Systems*, Princeton University, March 2004.

[5] B. Mukherjee, *Optical Communication Networks*, New York, McGraw-Hill, 1997.

[6] 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.

[7] 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, vol. 50, no. 10, October 2002.

[8] R. Dutta, G. N. Rouskas, "A Survey of Virtual Topology Design Algorithms for Wavelength Routed optical Networks," Optical Network Magazine, Jan. 2000.

[9] 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.

[10] 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.

[11] J-F, P. Labourdette, A.S. Acampora, "Logically Rearrangeable Lightwave Networks," IEEE Transactions on Communications, pp. 1223-1230, Aug. 1991.

[12] S. G. Nash, and A. Sofer, *Linear and Nonlinear Programming*, McGraw Hill, 1996.

[13] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, Second Edition, McGraw-Hill, 2001.