Rollout Algorithms for Topology Control and Routing of Unsplittable Flows in Wireless Optical Backbone Networks*

Abhishek Kashyap, Mehdi Kalantari, Kwangil Lee and Mark Shayman
Department of Electrical and Computer Engineering, University of Maryland, College Park MD 20742
Email: {kashyap, mehkalan, kilee, shayman}@glue.umd.edu

***Abstract*— We consider the problem of topology control and routing of unsplittable flows in wireless optical networks with point-to-point links. Such networks could form a backbone for either a cellular network or hierarchical ad hoc network. Each backbone node has a limited number of transceivers with which to establish links to neighbors. The objective is to form links and come up with routes so as to maximize the throughput for a given estimate of the traffic profile in the network. The problem is NP-hard. We propose a new framework for integrated topology control and routing using the mathematical technique of rollout. The rollout algorithms are derived from a heuristic that is widely used in logical topology design for *wireline* optical networks. Through simulation experiments, we show that the performance of the rollout algorithms is clearly superior to that of the heuristic on which they are based.**

I. Introduction

We consider the problem of integrated topology control and routing unsplittable flows in wireless optical backbone networks, taking the estimated traffic into consideration. The objective is to maximize the throughput for a given estimate of the traffic. The problem was proved NP-Hard in [1].The difficulty of the problem can be gauged by the fact that even the problem of routing unsplittable flows on a *fixed* topology is NP-Hard [2]. Our problem is NP-Hard even for splittable flows due to the interface (degree) constraints [1].

We consider wireless optical links because of their attractive characteristics which make them more suitable for backbone networks (compared to using RF and wireline optical links). Free-space optics (FSO) technology is expected to deliver unprecedented bandwidth, massive carrier reuse, ultra-low inter-channel interference, low power consumption, and cost savings where electrical wires and optical fibers are too expensive to deploy and maintain [3].

The problem of topology control for wireless optical networks is different from that in wireless RF (radio frequency) networks since the links are point-to-point as opposed to broadcast. In wireless optical networks, each node has a limited number of transceivers, and hence can establish links with only a limited number of nodes within its transmission range. We call the nodes within the transmission range of a node as its neighbors (if there is no obscuration) and the links as potential links (till they are formed). Thus, topology control is concerned with determining the neighbors with which to establish the limited number of possible links. In RF-wireless networks with isotropic antennas, topology control is closely related to power control. Power is controlled to reduce the transmission range to save power and decrease interference while providing adequate connectivity [4], [5], [6], [7].

*This research was partially supported by AFOSR under grant F496200210217.

There are important differences between topology control for reconfigurable wireline optical networks and topology control for FSO networks. In the wireline case, transmission range (lightpath length) is not a major issue. Furthermore, if the optical layer has sufficient resources so the routing and wavelength assignment problem is always solvable, then whenever a source and destination both have available interfaces, a direct connection (one logical hop) can be established. In contrast, in the wireless case, unless the destination is within the transmission range of the source, a multihop connection is required. [8] gives a detailed survey of the existing approaches proposed for logical network design in wireline optical networks. They group the heuristics into 3 categories of interest:

1) Mixed ILP formulation of the problem and using heuristics for solving it suboptimally: The heuristics include simulated annealing and genetic programming, variable depth local search techniques, and LP relaxation and rounding. These methods are very time intensive.
2) Maximization of single hop traffic flows: This set of heuristics ([9], [10], [11], [12]) tries to maximize the throughput by setting up direct lightpaths between sources and destinations having higher traffic demands. Thus, this set of heuristics is not directly applicable to our case as we cannot have a single hop path between each source and destination; we propose a heuristic derived from these in this paper, and show that the rollout algorithms are guaranteed to work better than that heuristic.
3) Heuristic maximization of single and multi-hop traffic flows: [8] divides these into two categories, one based on adding links to a null topology and another based on removing links from a full topology. These heuristics extensively use the fact that a direct lightpath can be created between any two nodes, and thus are not applicable to our problem.

There has been recent work on topology control in wireless optical networks ([13], [14], [1]): [14] does not take throughput into consideration, while [13] considers only ring topologies. [1] provides routes with multiple paths per traffic demand, which is not allowed in our framework.

We propose a framework for integrated topology design and routing. The algorithm consists of two phases - an offline phase and an online phase. The offline phase takes as an input the potential topology which consists of the backbone nodes and the potential links of each node. Each node is assumed to have a constraint on the number of interfaces it can have, and hence on the number of actual links that can be created from among the potential links. Given a traffic profile, the decision of selecting the links to be formed is made so as to maximize the total bandwidth guarantees we can give (which we call throughput for the offline phase). The traffic profile consists of aggregate traffic demands between different source-destination (SD) pairs. The routes are constrained to have one path per profile entry.

In the online phase the algorithm is distributed, and it attempts to route the requested flows using the paths computed during the offline phase. We exercise admission control by blocking the flows for which enough network bandwidth is not available from the bandwidth reserved for those flows. This way, we exercise admission control for the erring source-destination pairs (i.e., pairs having more demand than the bandwidth reserved during the offline phase). More details about this routing framework can be found in [15], [16]. This scheme can be used in MPLS networks in which LSPs are established in the offline phase and flows are rejected on the basis of bandwidth available for the SD pairs on the LSPs. Similar to MPLS, we can allow multiple service classes for each SD pair (having separate bandwidth reservations).

We describe the network model and problem statement in Section 2. Section 3 gives the integrated topology control and routing framework, followed by the rollout algorithms in Section 4. Section 5 gives the online algorithm. Section 6 discusses the computational complexity and simulation results. The paper is concluded in Section 7.

II. Network Model and Problem Definition

We model the network as a graph $G = (V, E)$, where V is the set of nodes and E is the set of potential links between them. We assume we do not have control over the position of the nodes. We consider wireless backbone networks in which each wireless node is equipped with point-to-point wireless optical interfaces. By the term ‘node’ we implicitly mean “backbone node”. Each node has the capability to perform routing. We assume that it does not move very frequently. We also assume that wireless links can be set up in any direction with all the nodes within transmission range, and take optical beam obscuration into consideration-i.e., some nodes within the transmission range may not be able to connect. Since the transmission distance is related to the power level of the node, the power level and thus the transmission range of each node can be different. The wireless links are unidirectional. If there is a pair of unidirectional links between two nodes, the link capacities may differ. The number of transmitters and receivers at each node is limited (which we call an interface constraint), thereby restricting the number of nodes to which it can connect.

We model the traffic as a collection of individual flows with Poisson arrival times with rate λ_i, exponential holding times (with mean T_i) and constant bit rate traffic (R_i) for each flow. The mean of the aggregate traffic demand for each ingress-egress pair (i) can be computed as $\lambda_i T_i R_i$. We generate the traffic profile (consisting of the aggregate traffic demands between ordered pairs of sources and destinations) using these mean aggregate demands.

The problem we address is to form a subgraph $G' = (V, E')$, such that the interface constraints are satisfied for all the nodes in the set V (i.e., the degree of each node is bounded by the number of available interfaces), and we maximize the throughput considering the traffic profile. We then use this information to achieve good throughput and low blocking rates when the network is functional. The offline part of the algorithm forms this subgraph, which we call topology control and comes up with routes and bandwidth reservations for the ingress-egress pairs given in the traffic profile. The topology computed is set up and the online part of the algorithm uses the information computed in offline phase to exercise admission control and select routes for individual flows. The server should recompute the topology, routes and bandwidth reservations whenever either the traffic profile or the (backbone) node locations change significantly. We do not anticipate that this would be done more often than hourly. The nodes then use this information to perform routing and traffic engineering on incoming flows.

III. Integrated Topology Control and Routing Framework

We propose a framework for finding the topology, routes and bandwidth reservations in an integrated way, so as to maximize the throughput while satisfying the interface and bandwidth constraints. Given a potential topology and traffic profile, we follow the steps:

1) A demand is chosen based on some criteria and a locally optimal path (satisfying the interface constraints and bandwidth constraints) is computed for the demand. If none exists, the demand is rejected.
2) If the path includes potential links, then those links are marked as actual links.
3) The capacity of each link on the path in the existing topology is updated (decreased) to incorporate the bandwidth allocated to the demand routed.
4) The topology is updated by eliminating all the potential links that lead to the violation of interface constraints i.e., at all the nodes for which the number of actual incoming (outgoing) links equals the number of interfaces, the incoming (outgoing) potential links incident on (going out of) those nodes are eliminated.
5) Steps 1, 2, 3 and 4 are repeated until all demands are either provisioned or rejected. This way, a topology is created from the potential topology and all the routes are computed for the demands given in the traffic profile (the ones we are able to route, the others are rejected).

Let us explain this framework of integrated topology control and routing with an example. In this example, we assume that each node has two interfaces available for establishing bidirectional links. The traffic profile is sorted in the order of decreasing demands, and demands are selected in that order. The link capacity of each link is assumed to be 10 units. We use constrained shortest-path routing for path selection, with the constraints being the limited interfaces and bandwidth.

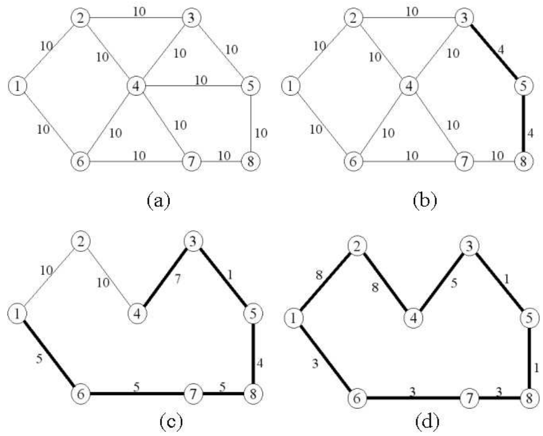

Fig. 1. (a) Potential Topology, (b) Topology after routing t_{38}, (c) Topology after routing t_{38}, t_{18}, t_{45}, (d) Topology after routing t_{38}, t_{18}, t_{45}, t_{37}.

The weight of each link is assumed to be 1. Let the traffic demands be: $t_{38} = 6, t_{18} = 5, t_{45} = 3, t_{37} = 2$. We compute the shortest path for the first demand t_{38} using the potential topology as shown in Fig. 1(a). The shortest path for the traffic demand t_{38} is $3-5-8$. Fig. 1(b) shows the topology after converting the potential links along the path $3-5-8$ to actual links and allocating the bandwidth for the demand. In Fig. 1(b), the actual links are represented by thick lines and the potential links are represented by thin lines. As the number of available interfaces per node is two and node 5 uses those interfaces for links with node 3 and node 8, there are no more interfaces available for node 5 to establish a link with other nodes. Thus, the potential link between node 4 and 5 is eliminated, as can be seen by comparing Fig. 1(a) and Fig. 1(b).

In the network of Fig. 1(b), we find the shortest path $1-6-7-8$ for the demand t_{18} and the shortest path $4-3-5$ for t_{45}. Fig. 1(c) shows the updated network which reflects the routing of these demands. Now we compute the shortest path for the demand t_{37} using the modified network, as shown in Fig. 1(c). There are two paths available for t_{37}: $3-5-8-7$ and $3-4-2-1-6-7$. Since the available bandwidth along the path $3-5-8-7$ is 1, which is less than the demand, the path cannot be selected even though it is the shortest path in the network. So, the shortest path for the demand t_{37} is computed as $3-4-2-1-6-7$, and the network topology updated to get the final topology as shown in Fig. 1(d).

A. Issues in Integrated Topology Control and Routing

The purpose of this integrated approach is to maximize the network throughput while routing demands sequentially. Let us consider the key issues with the help of two example networks shown in Fig. 2 and Fig. 3. In these examples, the number of available interfaces at each node is two, and the links are assumed to be bidirectional for simplicity. Given the traffic matrix $\{t_{12}, t_{34}, t_{56}, t_{78}\}$, all demands being the same, consider the path provisioning and topology design for the network in Fig. 2(a). When we provision a path for t_{12} first (shown by thick lines), we get the graph as shown in Fig. 2(b)

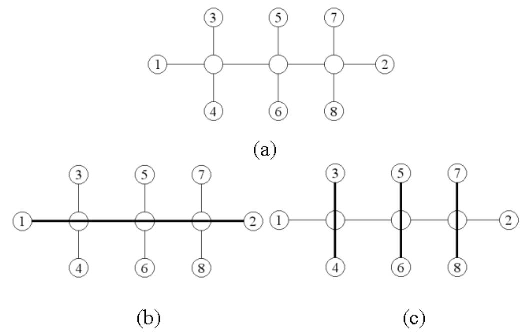

Fig. 2. Topology generation by considering demands in different order: (a) Example network, (b) Path for t_{12}, and (c) Paths for t_{34},t_{56},t_{78}.

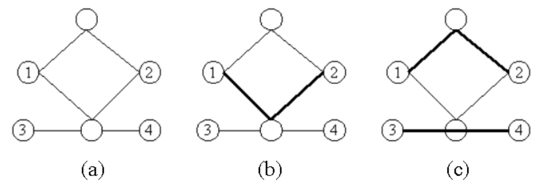

Fig. 3. Topology Generation by using different shortest paths for a demand: (a) Example network, (b) Path for t_{12}, and (c) Paths for t_{12} and t_{34}.

resulting in only one demand being provisioned. If we consider the other traffic demands first, then this demand cannot be provisioned but the other three demands can be provisioned, as shown in Fig. 2(c). Thus, the topology of Fig. 2(c) resulting from choosing the last three traffic demands before the first one gives a better throughput.

Let us consider another example using Fig. 3(a). We consider path provisioning for the sorted traffic demands $\{t_{12}, t_{34}\}$. There are two paths available for the demand t_{12}. If we choose a path for t_{12} as shown in Fig. 3(b), then we cannot provide a path for t_{34} because of the interface constraint at an intermediate node. However, when we choose the other path as shown in Fig. 3(c), both the traffic demands can be provisioned.

The above examples illustrate the importance of the two factors that affect the throughput in this framework: the sequence in which we route the demands given in the traffic matrix, and the selection of paths for routing the demands. These two factors affect the selection of links whose bandwidth will be used by the routed demand, and selection of links to be deleted due to interface constraints. So, these factors affect the future path computations and the output topology.

IV. Rollout Algorithms for Topology Control and Routing

As mentioned in Section 3.A, throughput of the network formed by our topology control and routing framework depends on the order in which the traffic demands are considered for link formation and routing, and the selection of the path for each demand. We start with reasonable heuristics for demand ordering and path selection and use the rollout technique to improve the heuristics to obtain potentially near-optimal solutions.

A. Basic Rollout Algorithm

Rollout is a general method for obtaining an improved policy for a Markov decision process starting with a base heuristic policy [17]. The rollout policy is a one step look-ahead policy, with the optimal cost-to-go approximated by the cost-to-go of the base policy. We use the specialization of rollout to discrete multistage deterministic optimization problems. Consider the problem of maximizing $G(u)$ over a finite set of feasible solutions U. Suppose each solution u consists of N components $u = (u_1, .., u_N)$. We can think of the process of solving this problem as a multistage decision problem in which we choose one component of the solution at a time. Suppose that we have a heuristic algorithm, the so-called "base heuristic", that given a partial solution $(u_1, .., u_n), (n < N)$, extends it to a complete solution $(u_1, .., u_N)$. Let $H(u_1, .., u_n) = G(u_1, .., u_N)$. In other words, the value of H on the partial solution is the value of G on the full solution resulting from application of the base heuristic. The rollout algorithm R takes a partial solution $(u_1, .., u_{n-1})$ and extends it by one component to $R(u_1, .., u_{n-1}) = (u_1, .., u_n)$ where u_n is chosen to maximize $H(u_1, .., u_n)$. Thus, the rollout algorithm considers all admissible choices for the next component of the solution and chooses the one that leads to the largest value of the objective function if the remaining components are selected according to the base heuristic.

It can be shown that under reasonable conditions, the rollout algorithm will produce a solution whose value is at least as great as the solution produced by the base heuristic. Note that the heuristic may be a greedy algorithm, but the rollout algorithms are not greedy as they make a decision based on the final expected value of the objective function, and not on the increment to the value of the objective function at that decision step. The rollout algorithm typically achieves a substantial performance improvement over the base heuristic at the expense of extra computation that is equal to the computation time of the base heuristic times a factor that increases polynomially with the problem size.

B. Rollout Algorithms for Topology Control and Routing

In this section, we propose three different rollout algorithms: index rollout, route rollout and integrated rollout. We start by explaining the base heuristic.

1) Base Heuristic: The base heuristic works as follows: Suppose that a partial topology has been obtained by choosing routes for n demands $(t_1, .., t_n)$ from the traffic profile. The base heuristic routes the remaining demands in decreasing order of magnitude. For each demand, it chooses a route using constrained shortest path first (CSPF), with constraints being that of interfaces and bandwidth. Thus, t_{n+1} is the largest remaining demand. The route chosen for this demand is a shortest unidirectional path in the partial topology satisfying the constraints. This means that every actual link in the path must have sufficient residual bandwidth for the demand; every potential link in the path must have an available transmitter at its head node and an available receiver at its tail node. If there is no feasible path, then the 'null' route is assigned– i.e., the demand is blocked. If there is a feasible path, the heuristic updates the topology by deleting the potential links which violate the interface constraints and decreasing the bandwidth of the links on that path (see Section 3 for description of this framework). Once t_{n+1} has been routed, the base heuristic routes the next largest demand t_{n+2} in the same way using the partial topology existing after t_{n+1} has been routed. The base heuristic algorithm continues in this way until all demands have been routed (or assigned null routes). The base heuristic is derived from the single hop traffic maximization heuristics for topology control and routing in wireline optical networks, [9], [10], [11], [12]. The main difference is that links are multi-hop in wireless networks, due to which interfaces are consumed at all intermediate nodes, while we need to take care of interfaces only at end-points in wireline networks. Another difference is that the routing and wavelength assignment (RWA) is discrete in wireline networks, while we check for bandwidth constraints on each link, which is continuous.

2) Index Rollout Algorithm: The example in Fig. 2 shows that the order in which traffic demands are routed plays an important role in determining the throughput of the resulting topology. Index rollout seeks to optimize this order. The index rollout algorithm works as follows: In the first step, the rollout algorithm uses CSPF to route the demand t_1 determined by the requirement that it maximize the total network throughput when the base heuristic is used to complete the topology starting with t_1.

Now, suppose that the demands $(t_1, .., t_{n-1})$ have been routed in this order by the rollout algorithm. In the next step, the rollout algorithm uses CSPF to route the demand t_n determined by the requirement that it maximize the total network throughput when the base heuristic is used to complete the topology starting with $(t_1, .., t_n)$. In other words, routing t_n next minimizes the sum of the remaining demands that are blocked. After routing each demand, the index rollout updates the topology to eliminate the links that violate the interface constraints, and decreases the residual bandwidth of the links on the path on which this demand is routed.

3) Route Rollout Algorithm: The example in Fig. 3 shows that the choice of path for each traffic demand plays an important role in determining the throughput of the resulting topology. Route rollout seeks to optimize the selection of path for each demand when the demands are considered in a fixed order. We consider the demands in decreasing order of magnitude. Let $(t_1, .., t_N)$ be the ordered sequence of demands. The base heuristic works as follows: Suppose that a partial topology has been obtained by choosing routes $(p_1, .., p_n)$ for the first n demands $(t_1, .., t_n)$. The base heuristic routes the remaining demands $(t_{n+1}, .., t_N)$ sequentially using CSPF. The route rollout algorithm works as follows: Fix an integer $K > 1$. In the first step, the rollout algorithm considers at most K feasible shortest paths as candidates for the route p_1 for the demand t_1. For each potential choice of p_1 it uses the base heuristic to complete the topology by routing the remaining traffic demands. The rollout algorithm then selects for p_1 the candidate that results in the maximum total network throughput. Now, suppose that the demands $(t_1, .., t_{n-1})$ have been given routes $(p_1, .., p_{n-1})$ by the rollout algorithm. In the next step, the rollout algorithm considers at most K feasible shortest paths as candidates for the route p_n for the demand t_n. For each potential choice of p_n it uses the base heuristic to complete the topology by routing the remaining traffic demands. The rollout algorithm then selects for p_n the candidate that results in the maximum total network throughput. Note that if there is only one feasible shortest path for a traffic demand, the routing decision made by the rollout algorithm coincides with the decision made by the base heuristic.

It might appear desirable to consider all feasible shortest paths as candidates for p_n. However, this is not possible since the problem of finding all such paths requires exponential time. Consequently, we limit the number of paths considered to K, where the upper bound K is chosen small enough to allow reasonable computation time given the size of the network.

4) Integrated Rollout Algorithm: In integrated rollout, we make the decisions of choosing the demand to be routed and the path to be used for that demand simultaneously. In integrated rollout, each component of a solution is a pair (t_k, p_k) consisting of a traffic demand and its path. Thus, the algorithm seeks to optimize the sequence $((t_1, p_1), .., (t_N, p_N))$. The base heuristic takes a partial solution $((t_1, p_1), .., (t_n, p_n))$ and extends it to a complete solution by choosing the remaining traffic demands $(t_{n+1}, .., t_N)$ in order of decreasing magnitude and choosing paths $(p_{n+1}, .., p_N)$ (some of which may be null) for these traffic demands sequentially using CSPF. The integrated rollout algorithm works as follows: In the first step it considers all pairs (t_1, p_1) where t_1 is any of the traffic demands and p_1 is any one of a maximum of K feasible shortest paths for t_1. It selects the pair (t_1, p_1) that gives the maximum total network throughput when the base heuristic is used to extend it to a full topology. Now, if the rollout algorithm has produced the sequence $((t_1, p_1), .., (t_{n-1}, p_{n-1}))$, it considers pairs (t_n, p_n) where t_n is a remaining demand and p_n is any one of a maximum of K feasible shortest paths for t_n. It selects the pair (t_n, p_n) that maximizes the total network throughput when the base heuristic is used to extend $((t_1, p_1), .., (t_n, p_n))$ to a full solution.

We show that all the proposed rollout algorithms work better than the heuristic: At the first step, the throughput for rollouts is at least as large as that for heuristic as we form the whole topology according to the heuristic. The method of choosing the routes makes sure that the rollout algorithms work at least as good as the heuristic, as at each decision step, they always have the choice of going according to the heuristic which gives the throughput which was calculated at the previous step. Thus, the rollouts perform at least as well as the heuristic in terms of the throughput (the objective function).

V. Online Routing and Admission Control

The topology is set-up, and the bandwidth reservation information and the route for each ingress-egress pair is given to the ingress node for that pair. Whenever a flow (new request of traffic between an ingress-egress pair) arrives, the ingress router checks to see if there is enough bandwidth left from the bandwidth reserved for this pair (similar to Intserv [18]). If there is bandwidth left, then the flow is routed through the path stored from the offline phase. We also have additional unreserved bandwidth on some links in the network, so in the case of reserved bandwidth being exhausted for an ingress-egress pair, the unreserved bandwidth is used (on a first-come-first-serve basis). If the flow cannot be routed using the reserved bandwidth or the extra unreserved bandwidth, it is blocked.

VI. Computational Complexity and Simulations

A. Computational Complexity

When the network is in operation (online phase), the time required for the routing decision is O(1). We now analyze the time complexity for offline phase algorithms: Let the number of nodes in the network be N and the number of aggregate demands in the traffic matrix be M ($M = O(N^2)$). We use a modified version of Dijkstra's shortest path algorithm, [19] for finding the shortest paths. It is modified to take care of the interface and bandwidth constraints while finding a shortest path. The graph at any intermediate state of the rollout algorithm is not sparse, so the process of finding a shortest path takes $O(N^2)$ time. The heuristic we use for sorting is sorting by decreasing order of traffic demands, which takes $O(MlogM)$ time for sorting M aggregate demands. This time is insignificant compared to the time taken by other components of the algorithms, so it does not show up in the time complexity of any of our algorithms. The time complexity of the heuristic algorithm is $O(MN^2)$, as shortest paths are computed M number of times.

The time complexity of the route rollout algorithm is $O(M^2N^2)$, as K is fixed. This complexity is due to the fact that at each decision step, $O(M)$ shortest paths are computed, and there are M decision steps in the algorithm. The time complexity for the index rollout algorithm is $O(M^3N^2)$. At each decision step in the algorithm, $O(M^2)$ shortest paths are computed, and there are M decision steps in the algorithm resulting in the above complexity. The complexity for the integrated rollout is also the same as the time is scaled by K which is a constant.

B. Simulation Results and Analysis

The network was assumed to have the following parameters:

- Number of nodes = 50, uniformly distributed, with each node having an average of 7.5 potential neighbors.
- Number of receive and transmit interfaces at each node = 3 each.
- The transmission range of all nodes is assumed to be the same.
- Capacity of each link = 100 in each direction.
- Number of nodes capable of being source/destination=12.
- Number of source-destination pairs = 100, selected from among the nodes which can be sources or destinations.
- Poisson arrival rate at each source (λ_i): Uniformly distributed between 10 and 20 per unit time.
- Mean of Holding Time (T_i): Uniformly distributed between 1 and 2 units of time.

TABLE I
AVERAGE BANDWIDTH GUARANTEES

Heuristic	Route Rollout	Index Rollout	Integrated Rollout
0.8782	0.9326	0.9582	0.9597

TABLE II
AVERAGE THROUGHPUT

Heuristic	Route Rollout	Index Rollout	Integrated Rollout
0.7729	0.8232	0.8520	0.8542

- Bit Rate of individual flows = 1 unit (same for all).
- Number of Shortest Paths considered in route and integrated rollout, $K = 4$.
- Weight of each link for constrained shortest path computation = 1, thus making the shortest path as the constrained min-hop path.

The simulation was run 10 times and in each simulation, network topology was formed starting with these parameters. Table I shows the average fractional throughput (bandwidth guarantees/total demand) for the heuristic and the rollout algorithms.

As can be seen from Table I, comparing with the heuristic in terms of throughput, the route rollout performs nearly 6.2% better, the index rollout performs 9.1% better and the integrated rollout performs 9.3% better. So, the integrated rollout is expected to perform the best among these rollouts. Another observation from the results is that the index selection is more critical than the selection of routes from among multiple routes. This can be inferred from the fact that the index rollout works much better than the route rollout while the integrated rollout does not work that much better than the index rollout.

The network was setup and Poisson traffic with exponential holding times and CBR rate (the parameters being the same as provided to offline phase) was generated and the network was run for 30 units of time (sufficient considering the call arrival parameters) for each of the 10 simulations. In each simulation, the traffic for evaluating the heuristic was the same as that for the rollout. Table II gives the average throughput (which is the same as call acceptance rate as traffic is CBR with same rate for all pairs) for each of the policies. As can be seen, the results are similar to the bandwidth guarantees we could achieve in the offline phase.

As explained above, the rollout algorithms are guaranteed to work better than the heuristic. The network parameters are chosen here to allow the demonstration of potential improvement. Changing the parameters effectively changes the amount of load on the links, and the relative performance of different methods remains similar [20] and is omitted for brevity.

The optimization of the throughput ensures with high probability that the source and destination nodes are all connected. If certain other nodes are not essential as transit nodes, it is possible that these nodes may be disconnected.

VII. CONCLUSION

The problem of profile based topology control and routing of unsplittable bandwidth-guaranteed flows for maximizing throughput is addressed. An integrated framework for topology control and routing decision is proposed, and efficient rollout algorithms are proposed for the offline component of the algorithm. An initial heuristic algorithm is developed based on some heuristic algorithms used for logical topology design in wireline optical networks. The rollout algorithms are proved to perform better than the initial heuristic, and the potential improvement is demonstrated by simulations. The time complexity of the online routing algorithm is O(1), and it is polynomial in the number of nodes for the offline computations.

REFERENCES

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

[2] M. Garey and D. Johnson, "Computers and Intractability: A Guide to the theory of NP-Completeness", Freeman and Company, 1979.

[3] N. A. Riza, "Reconfigurable Optical Wireless", *LEOS '99* IEEE , vol.1 , pp.70-71, 8-11 Nov. 1999.

[4] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment", *IEEE Infocom 2000*, vol. 2, pp. 404-413, 26-30 March 2000.

[5] Z. Huang, C-C. Shen, C. Srisathapornphat and C. Jaikaeo, "Topology Control for Ad Hoc Networks with Directional Antennas", *ICCCN 2002*, pp. 16-21, Miami, Florida, October 2002.

[6] R. Wattenhofer, L. Li, P. Bahl and Y-M. Wang, "Distributed Topology Control for Wireless Ad-hoc Networks", *IEEE Infocom 2001*, pp. 1388-1397.

[7] L. Bao and J. J. Garcia-Luna-Aceves, "Topology Management in Ad Hoc Networks", *ACM Mobihoc*, pp. 129-140, Annapolis, Maryland, June 2003.

[8] E. Leonardi, M. Mellia, M. A. Marsan, "Algorithms for the logical topology design in WDM all-optical networks", *Optical Networks Magazine, Premiere Issue*, vol. 1, no. 1, pp. 35-46, January 2000.

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

[10] R. Ramaswami, K. N. Sivarajan, "Design of topologies: A linear formulation for wavelength routed optical networks with no wavelength changers", *IEEE Journal of Selected Areas in Communications*, June 1996.

[11] J. F. Labourdette, A. Acampora, "Logically rearrageable multi-hop lightwave networks", *IEEE Transactions on Communications*, vol. 39, no. 8, pp. 1223-1230, October 1991.

[12] Z. Zhang, A. Acampora, "Heuristic wavelength assignment algorithm for multihop WDM networks with wavelength routing and wavelength reuse", *Proc. IEEE Infocom '94*, June 1994.

[13] A. Desai, "Dynamic Topology Control of Free Space Optical Networks", M.S. Thesis, Department of Electrical and Computer Engineering, University of Maryland, 2003.

[14] P. C. Gurumohan, J. Hui, "Topology Design for Free Space Optical Networks", *ICCCN 2003*.

[15] Y. Wang, Z. Wang, "Explicit Routing Algorithms for Internet Traffic Engineering" , *Proc. ICCCN*, pp. 582 - 588, March 1999.

[16] S. Suri, M. Waldvogel, D. Bauer, P. R. Warkhede, "Profile-Based Routing and Traffic Engineering", *Computer Communications*, vol. 24(4), pp. 351-365, March 2003.

[17] D. P. Bertsekas, "Dynamic Programming and Optimal Control", vol. 1, Athena Scientific, 2000.

[18] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: An Overview", RFC, Network Working Group, June 1994.

[19] Edsger W. Dijkstra, "A note on two problems in connection with graphs", *Numerische Mathematik*, vol. 1, pp. 269-271, 1959.

[20] A. Kashyap, "Profile Based Topology Control and Routing in Wireless Optical Networks", M.S. Thesis, Department of Electrical and Computer Engineering, University of Maryland, 2004.