Integrated Topology Control and Routing in Wireless Optical Mesh Networks*

Abhishek Kashyap*, Kwangil Lee*, Mehdi Kalantari*,
Samir Khuller† and Mark Shayman*
*Department of Electrical and Computer Engineering
University of Maryland, College Park, USA
Email: {kashyap, kilee, mehkalan, shayman}@glue.umd.edu
†Department of Computer Science
University of Maryland, College Park, USA
Email: samir@cs.umd.edu

***Abstract*— We study the problem of integrated topology control and routing in Free Space Optical (FSO) mesh backbone networks. FSO links are high-bandwidth, low interference links that can be set-up very fast, making them suitable for mesh networking. FSO networks are highly constrained by interface constraints, i.e., constraints on the number of FSO links a node can establish. We prove the problem to be NP-Hard and propose efficient algorithms for integrated topology control and single-path or multi-path routing.**

Keywords: Wireless mesh networks, free space optical networks, topology control and routing, rollout algorithms, matching, multi-commodity flow.

I. Introduction

Wireless mesh networks have gained much popularity due to their application in last mile extension of metropolitan area networks (MANs), and local area networks (LANs). In last mile extensions of MANs, wireless mesh networks are multi-hop networks being used as backbone networks connecting end-users (LAN gateways connected to mesh routers in the network) with the access points connected to the Internet. Wireless mesh networks are an attractive option over optical fibers because of their ease of installation and cost-effectiveness of deployment.

Various technologies have been proposed for use in wireless mesh networks: 802.15 (UWB) [1] is proposed for personal area mesh networks, 802.11s [22] is proposed for local area mesh networking, and 802.16 (WiMax) [2] is proposed for MANs (last mile extension). All the proposed standards assume a shared physical channel and use RF technology, which is prone to interference among transmitters and provides low security [5]. Also, a node farther from an access point may get a much lower throughput (compared to nodes closer to the access point, resulting in unfairness) using currently deployed MAC and routing protocols [19]. Thus, the throughput, end-to-end delay and fairness degrade as the number of nodes or number of hops increases. Therefore the RF-based technologies using current protocols are not scalable with the network size [5]. Generally, it is desired to provide bandwidth guarantees and QoS to traffic on wireless mesh backbone networks, as in IP networks. Thus, due to aforementioned reasons, RF-based technologies are not completely suitable to be deployed in wireless mesh backbone networks.

Free Space Optics (FSO) [40] technology is an attractive option for use in mesh networks [44]. FSO links are point-to-point links that use long-range (up to 4 km in current products), high bandwidth (up to 2.5Gbps in current products) narrow laser beams. There is very low interference among transmitters, and difficulty in intercepting the very narrow laser beam provides high security. Therefore, the technology is scalable with the number of nodes and number of hops in the network and FSO links can provide the QoS required by the end-users. Also, FSO links offer the flexibility of fast tracking and setup of links [23]. Thus, the network topology can be reconfigured very easily as the traffic pattern or node positions change. These attractive characteristics of FSO links make them suitable for use in military backbones networks as well. FSO links have the problem of obscuration due to atmospheric conditions such as fog and snow. This problem can be alleviated by using hybrid RF/FSO networks, in which RF links are used for back-up in the event of FSO link obscuration [24], [30], [29]. The algorithms for hybrid RF/FSO networks can be used to route back-up traffic (and optimize RF topology is RF

*This research was partially supported by AFOSR under grant F496200210217 and NSF under grant CNS-0435206.

interfaces are separate [29]) for the FSO topology and routing computed by algorithms proposed in this paper.

The number of FSO transceivers at a node is limited due to cost concerns, and the selection of links to be established at each node is very critical as that affects the performance of the network. We call this problem *topology control*. In this paper, we propose algorithms for establishing network topologies that achieve good performance (our performance measure is the throughput) for a given traffic profile. The traffic profile (expected aggregate demands) within a domain can either be predicted by observing the traffic in the network ([16], [15], [9], [37]), or can be inferred from the Service Level Agreements (SLAs). It has been shown that the traffic profile in IP backbone has a pseudo-periodic behavior on different time-scales (e.g., day, week, etc.), which is predictable based on the past history of the traffic [16], [17]. We can exercise topology control based on these profiles to achieve better network performance.

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 such that 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. Most algorithms proposed for topology control for reconfigurable wireline optical networks use the fact that a direct lightpath can be established between any two nodes [34], [6], [39], [33], [45], and hence are not applicable to our problem. The other algorithms are based on simulated annealing and genetic programming, which are too slow (they have exponential time complexity) to be run periodically.

Leonardi et al. [34] give 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 ([6], [39], [33], [45]) 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: [34] 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.
4) Another set of heuristics works on optimizing node placement, which is not the problem we are considering here. We assume we cannot control the placement of nodes.

The paper is organized as follows: Section II presents the related work. Section III presents the network model and problem statement. Section IV describes the integrated topology control and single-path routing framework. Section V describes rollout algorithms for topology control and single-path routing, along with the simulation results. Section VI gives the algorithms for topology control and multi-path routing, with the simulation results. Section VII discusses the computational complexity of the algorithms. Section VIII concludes the paper.

II. Related Work

There has been recent work on routing in wireless optical (or Free Space Optical (FSO)) networks [38]. The authors assume a network model in which each node can transmit to a certain sector of angle α and radius r around it, and any node in that sector can hear it. This leads to multiple uni-directional links from each node, depending on the number of receiving nodes within its transmission sector. The network is a sensor network, and the nodes communicate with a base station through multi-hop paths using the above-described uni-directional links. The authors provide efficient algorithms for node discovery and path computation in this network model. We assume a model in which one FSO transmitter can transmit to only one receiver at any point in time, and a receiver can receive from only one transmitter at a time; and there is a limitation on the number of transmitters and receivers at each node. Thus, there cannot be an arbitrary number of uni-directional links from a single node. We also consider multiple

ingress/egress nodes rather than having a single base station. We also consider the joint problem of topology control (deciding which transmitter should align with which receiver) and routing. Thus, our problem is very different from their problem.

There has also been recent work on topology control in FSO networks [21]. The authors consider a model as ours, in which there are a limited number of transmitters and receivers at each node, and for the transmitter needs to be aligned with the receiver for communication. The authors propose algorithms for constructing a topology that is shown to be strongly connected for most instances of the problem (it cannot be guaranteed for all values of limit on transmitter/receiver to be connected since that problem is NP-hard).

There have been recent studies on topology control in wireless optical networks: the problem of topology control and multi-path routing for maximizing throughput for a given traffic profile has been considered in [27]. The problem of topology control and single-path routing for maximizing throughput has been studied in [26]. The constraints are on the number of transceivers at each node, and on the capacity of each link. The current paper integrates and expands on the results presented in [27], [26]. The problem of maximizing throughput with the assumption of infinite amount of traffic to be sent between every ingress-egress pair has been considered in [25].

A specific instance of the topology control and routing problem has been addressed in [13], that assumes two transceivers per node, and establishes ring topologies. The problem is NP-hard, and thus they provide heuristics that are shown to generate a connected ring topology in most instances. The objective is to minimize congestion in the network for the given traffic matrix. Another specific instance of the problem has been addressed in [12], with each node having three transceivers. The authors provide heuristics to design bi-connected networks with minimum congestion for a given traffic matrix. The authors provide cross-layer algorithms for topology control (without routing) in [35]. The algorithms weight each link according to the Bit Error Rate (BER) with the objective to compute a minimum weight bi-connected topology for the two specific cases of two and three transceivers per node. In another work [46], the two objectives of minimizing the total BER and minimizing the congestion in the network is considered and heuristics are provided to solve the multi-objective problem. Our algorithms do not assume any such restriction on the number of transceivers per node.

We now mention some related work in the area of routing and topology control for hybrid RF/FSO networks. Izadpanah et al. [24] propose the use of monitoring average recent Bit Error Rates at each FSO link, and switching partial traffic to the parallel RF link if the value is below a threshold. They provide with decision steps for when to switch and how much traffic to switch. The authors also propose routing on shortest paths with the BERs representing the link weights. The problem of jointly optimizing RF and FSO routes to provide min-max weighted back-up guarantees has been studied in [30]. The traffic between each source-destination pair (traffic demand) is weighted according to percentage real-time traffic in it, and the objective is to maximize the minimum weighted fraction of each traffic demand that has back-up on RF links. The secondary objective is to maximize the throughput on remaining (FSO+RF) capacity of the network. The traffic and its back-up on RF links flows simultaneously so immediate back-up can be provided in the event of obscuration. The authors also propose algorithms for joint topology control and routing for RF links for given routing on the FSO network [29]. The network is assumed to be divided into grids, and it is assumed a single grid (thus all links passing through it) is obscured at a time. Their is a probability associated with the obscuration of each grid, and the RF topology and back-up routes are computed so as the maximize the average (over grid failures) of the minimum back-up provided among traffic demands for each grid failure. The above-mentioned algorithms can be used after design of topology and routing in the FSO network using the algorithms presented in the current paper.

III. Network Model and Problem Statement

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. A potential link exists between two vertices if they are within each other's transmission range. We consider wireless backbone networks in which each wireless node is equipped with point-to-point wireless optical interfaces. The terminal nodes (users) may connect to the backbone nodes using omni-directional RF links. By the term 'node' we implicitly mean "backbone node". We assume that a node can transmit to another node within its transmission range by aligning its transmitter with the other node's receiver. The steering of the optical beam can be performed in a variety of ways [43], [32], [11]. Coarse steering of the beams can be done using gimbals, and finer alignment can be done using Fast Steering Mirrors (FSMs), electro-optic devices or acousto-optic devices [43]. Since the alignment can be disturbed due to movement of transmitter/receiver, or building sway, the transmitter can either use a beam with higher cone angle to counter building

sway [44], or use active beam tracking techniques like photodiode arrays or CCD arrays [43]. Since very accurate steering hardware and active tracking techniques can be expensive, we expect the use of larger beam angles to counter the slight alignment mismatch and building sways, while losing on the link capacity a little.

Each node has the capability to perform routing, which is multi-hop between source and destination nodes[1]. We also assume that wireless links can be set up in any direction with all the nodes within transmission range. 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 uni-directional[2]. If there is a pair of uni-directional 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 $G' = (V, E')$ and compute a routing on the subgraph, such that the interface constraints are satisfied for all nodes in set V (i.e., the degree of each node is bounded by the number of available interfaces), and we maximize the throughput while routing the traffic profile. We consider both single-path routing, i.e., each aggregate demand should be routed on a single path; and splittable routing, i.e., each demand can be split among multiple paths. 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 the 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. We show the problem to be NP-Hard in Theorem 3.1 [27].

Theorem 3.1: Given a graph $G = (V, E)$ and a traffic profile consisting of traffic demands between different vertices of G, the problem of finding a degree constrained subgraph $G' = (V, E')$ and routes for the traffic demands such that the total demand routed is maximized is NP-Hard.

Proof: Consider a small amount of traffic between each pair of nodes in the network (small enough not to violate any link bandwidth constraints). The problem of maximizing the throughput reduces to finding a connected subgraph (satisfying the degree constraints) here. We can remove the extra edges and the problem reduces to finding a degree-constrained spanning tree, which is a known NP-Hard problem. A special case is where we have a degree constraint of one incoming and one outgoing edge at each node. In this case, the problem reduces to finding a hamiltonian cycle, which is known to be NP-Complete [20].

IV. Integrated Topology Control and Single-path Routing Framework

We propose a framework for finding the topology, single-path 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 execute the following steps:

1: A demand is chosen based on some criteria and a locally optimal path (satisfying the interface 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 the potential links that lead to the violation of interface constraints, i.e., at 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

[1]Multi-hop routing has been shown to outperform single-hop routing in FSO networks (where all nodes can potentially connect to all other nodes) [4], thus, the realistic restriction of having multiple hops actually improves the routing performance.

[2]There has been work that enables bi-directional links as well [41], [3]. Our algorithms will work in this scenario as well, although with changes in the shortest path computation algorithms [31].

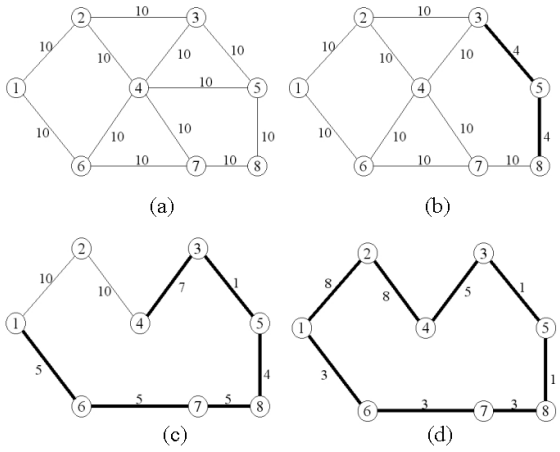


Fig. 1. (a) Potential topology, (b) Topology after routing t_{38}, (c) Topology after routing t_{38}, t_{18}, t_{45}, (d) Topology after routing all demands

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).

As an example, 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. 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} $(3-5-8)$ using the potential topology as shown in Fig. 1(a). 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 there are no more interfaces available for node 5 to establish a link with other nodes, the potential link between node 4 and 5 is eliminated, as can be seen by comparing Fig. 1(a) and Fig. 1(b).

Fig. 1(c) shows the updated network, reflecting the routing of t_{18} and t_{45}. 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, t_{37} is routed on $3-4-2-1-6-7$, and the network topology is updated to get the final topology as shown in Fig. 1(d).

V. Rollout Algorithms for Topology Control and Routing

We propose heuristics for demand ordering and path selection for the integrated topology control and routing framework. We then 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 [7]. 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. We proved in [26] that the rollout algorithms proposed here perform better than the 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.

B. Rollout Algorithms for Topology Control and Routing

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 the first 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)[3], with constraints being that of available 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 demand is blocked. If there is a feasible path, the heuristic updates the topology by deleting the potential links that violate the interface constraints and decreasing the available 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, [6], [39], [33], [45]. 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 of lightpaths 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: Index rollout seeks to optimize the order in which the traffic demands are routed. 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: Route rollout seeks to optimize the selection of path[4] 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, an exponential number of shortest paths may exist. 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

[3]The route is non-unique, and we use a modified version of Dijkstra's algorithm to find one of the possible shortest paths.

[4]The algorithm evaluates multiple constrained shortest paths.

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.

C. Online Routing and Admission Control

After the topology and routing is determined, 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 call (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. If there is bandwidth left, then the flow is routed through the path stored from the offline phase. We may have additional unreserved bandwidth on some links in the network (the bandwidth left unreserved), so in the case of reserved bandwidth being exhausted for an ingress-egress pair, the unreserved bandwidth is used (on a first-come-first-served basis). If the call cannot be routed using the reserved bandwidth or the extra unreserved bandwidth, it is blocked.

D. Simulation Results and Analysis

The network we considered for simulations was assumed to have the parameters listed in Table I. All the nodes were assumed to have the same transmission range.

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

Table II shows that compared to the heuristic, the route rollout performs nearly 6.2% better, the index rollout performs 9.1% better and the integrated rollout performs 9.3% better. So, generally the integrated rollout is expected to perform the best among these rollouts.

TABLE I
SIMULATION PARAMETERS

Number of nodes in the network (uniformly distributed)	50
Average degree of each node	7.5
Number of receive interfaces at a node	3
Number of transmit interfaces at a node	3
Link capacity (unidirectional)	100
Number of nodes capable of being a source/destination	12
Number of source-destination pairs	100
Poisson Rate (λ_i), uniformly distributed	10 - 20
Mean of Holding Time (T_i), uniformly distributed	1 - 2
Bit Rate of individual calls	1
No. of shortest paths considered in rollout algorithms (K)	4
Weight of each link in shortest path computation	1

TABLE II
AVERAGE BANDWIDTH GUARANTEES

Heuristic	Route Rollout	Index Rollout	Integrated Rollout
0.8782	0.9326	0.9582	0.9597

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 algorithms have been shown to have similar performance under various traffic conditions (light, medium, heavy) in [28].

The network was setup and Poisson traffic with exponential holding times and constant bit rate (the parameters being the same as provided to offline phase) was generated, and the network was run for 30 units of time for each of the 10 simulations. In each simulation, the traffic for evaluating the heuristic was the same as that for the rollout. Table III 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 algorithms. As the results show, the relative performance is similar to the bandwidth guarantees we could achieve in the offline phase. The throughput is lower than guaranteed since the guarantees were for a constant arrival rate (equal to the mean arrival rate of the Poisson process in the simulation), while it actually is Poisson with the given mean arrival rate, so there are time periods when the arrival rate is higher, leading to lower throughput.

TABLE III
AVERAGE THROUGHPUT

Heuristic	Route Rollout	Index Rollout	Integrated Rollout
0.7729	0.8232	0.8520	0.8542

VI. TOPOLOGY CONTROL AND MULTI-PATH ROUTING

In this section, we propose algorithms for topology control and multi-path routing. We first describe the multi-commodity flow formulation of the multi-path routing problem on a fixed topology, which is a linear program and can be solved efficiently. We then propose algorithms based on maximum weight matching [36] for topology computation.

A. Multi-commodity Flow Formulation of Routing

We set up the problem of routing a given traffic profile over a computed topology for maximizing the throughput as a linear multi-commodity flow problem [42]. Let there be M commodities (profile entries, the value of each entry is b_i), N nodes and L links in the network. We add a dummy link (very high cost, very large capacity) between the source and destination of each commodity to achieve feasibility (thus, there are M such links). Let x_i^l be the amount of commodity i routed through link l. Let c_l represent the cost of each link, which is 1 for an actual link for our objective of maximizing the throughput. Let the set of incoming and outgoing links at node j be denoted by I_j and O_j respectively. Let s_i and d_i represent the source and destination of profile entry i. Let σ_l represent the capacity of link l. Equation 1a achieves the objective of maximizing the throughput as the algorithm tries to route on the actual links due to large cost of the dummy links. Equation 1b represents the bandwidth constraints. Equations 1c and 1d represent the flow conservation laws.

$$\min \sum_{l=1}^{L+M} (c_l \sum_{i=1}^{M} x_i^l) \tag{1a}$$

$$\text{s.t.} \sum_{i=1}^{M} x_i^l \leq \sigma_l, \ \forall l \in \{1,..,L\} \tag{1b}$$

$$\sum_{l \in I_j} x_i^l = \sum_{l \in O_j} x_i^l, \ \forall j \in \{1,..,N\} - \{s_i, d_i\}, \forall i \in \{1,..,M\} \tag{1c}$$

$$\sum_{l \in O_j} x_i^l - \sum_{l \in I_j} x_i^l = b_i, j = s_i, \forall i \in \{1,..,M\} \tag{1d}$$

$$x_i^l \geq 0, \ \forall i \in \{1,..,M\}, \forall l \in \{1,..,L+M\} \tag{1e}$$

B. Topology Control using Matching Theory

We present an algorithm based on maximum weight matching. The algorithm is outlined below and explained in the following subsections.

1: Weight the links in the initial graph to favor the links that are expected to carry higher traffic.
2: Construct an auxiliary graph based on the network. We will explain the construction procedure later.
3: Find a maximum weight matching [36] to compute a maximum weight subgraph that satisfies the interface constraints.
4: Construct a topology based on the matching output.
5: Find the routing by solving the linear program of Section VI-A on this topology.
6: Modify the topology sequentially and solve the MCF linear program again each time, and finally, keep the topology that gives the maximum throughput.

1) Initial weight assignment: We propose three strategies for weighting the network links for construction of the auxiliary graph. The way we map the problem to maximum weight matching problem, if we give the same weight to all links in the network, the output topology will have the maximum number of links while satisfying the degree constraints (as we try to maximize the weight during maximum weight matching, same weight to all edges would result in maximizing the number of edges). We call the algorithm using this policy *Uniform Weighted Matching (UWM)*. As our objective is maximizing the throughput, so it may be better to give extra weight to edges that are expected to carry more traffic. We call the algorithm using this policy as *Traffic Weighted Matching (TWM)*, and explain it below:

1: Give a weight of one to all edges.
2: Find (maximum) K shortest paths of same length for each traffic profile (over the potential topology).
3: Each time a link occurs in a path, increment its weight by $profile(i)/numSP$, where $profile(i)$ is the value of the profile entry, and $numSP$ the number of shortest paths found ($\leq K$) for that entry.

Another possibility is to assign weights according to the number of paths a link occurs on, irrespective of the amount of traffic. In this case, we increment the weight of a link by one in step 3 above. We call the algorithm using this strategy *Flow Weighted Matching (FWM)*. We work with shortest paths as the multi-commodity flow formulation will prefer shorter paths for a commodity because it minimizes the cost function of Equation 1a (to maximize the throughput).

2) Construction of auxiliary graph: We now explain the procedure for constructing the auxiliary graph to be used as an input for maximum weight matching. Given

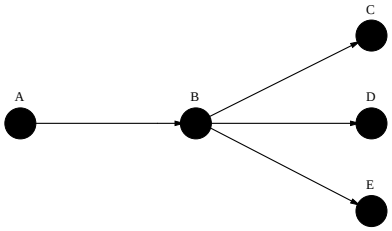


Fig. 2. Potential topology, $\Delta = 2$

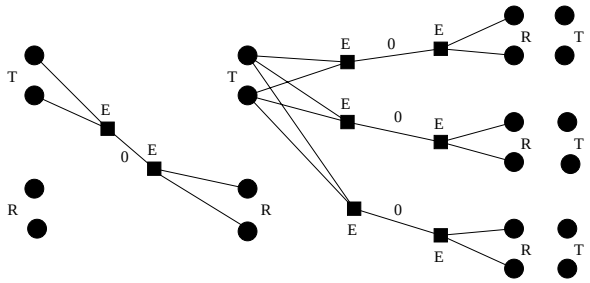
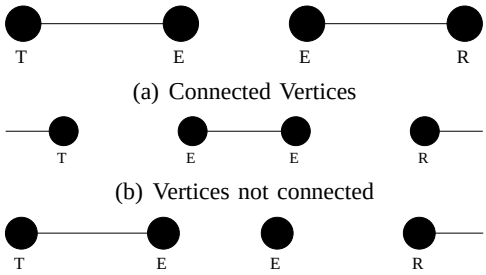


Fig. 3. Modified graph from potential topology



(a) Connected Vertices

(b) Vertices not connected

(c) Connection between vertices undecidable

Fig. 4. Rules for deducing connectivity

a graph $G = (V, E)$ (that corresponds to the potential network) with edge weights as explained in Section VI-B.1, we form a graph $G' = (V', E')$ and give it as an input to the maximum weight matching problem. Let the number of transmit and receive interfaces (i.e., incoming and outgoing degree constraints) at each vertex be Δ. Figure 2 shows an example potential topology with $\Delta = 2$. The steps of mapping are explained below.

1: For each vertex, form 2Δ vertices in graph G'. Δ of them correspond to the transmit interfaces, and Δ to receive interfaces.
2: For each edge e in G, form two vertices (t_e, h_e) in G', and add an edge of weight zero between them.
3: For each edge e in G with weight w_e, add edges of weight w_e between t_e (in G') and all transmit interface vertices of the tail vertex of e. Also, add edges of weight w_e between h_e and all receive interface vertices of the head vertex of e. Figure 3 shows G' for our example. T, R and E refer to the vertices corresponding to transmit interfaces, receive interfaces and edges in G respectively. The edges shown without weights have a weight of 1.
4: Add a clique with $2\Delta N$ vertices to the graph G', with each edge having a weight zero. Add zero weight edges between each of these vertices and each of the vertices in G' that correspond to the transmit and receive interfaces of the vertices in graph G.
5: Sum the weights of all the edges in G' and add that to the weight of all edges in G'. We do this to ensure we get a perfect matching using a maximum weight matching algorithm.

We solve the maximum weight matching problem on G', and deduce the resulting topology based on the result we get. We show the rules for mapping from the output of matching to the graph representing the topology: Figure 4(a) shows the scenario (a subgraph from the output of matching algorithm) when two vertices will have an edge between them in the final topology. Figure 4(b) shows the case when two vertices that had an edge between them in G will not have the edge in the resulting topology (as they are not connected to the edge vertices corresponding to the edge between them). We detect such subgraphs in the output graph from matching algorithm (the output graph will be composed of such subgraphs only), and use these rules to deduce the resulting topology.

We added a clique to the graph G' to avoid the scenario in which we cannot deduce the topology from the result of the matching algorithm. Figure 4(c) shows the case when this has happened between two vertices. In this case, one of the vertices is matched to the edge vertex in G', while the other vertex is not matched with the corresponding edge vertex. This can be avoided by having perfect matching, and for achieving that we add a clique of size $2\Delta N$ and connect each of them to all the vertices in G' that correspond to transmit and receive interfaces of vertices in G. We accommodate the worst case in which all the vertices will be disconnected in the final topology.

Figure 5 shows the output (minus the clique vertices and corresponding edges) of the matching algorithm and Figure 6 shows the final topology we get for the example network of Figure 2.

3) Topology change strategy: We solve the multi-commodity flow problem (as explained in Section VI-A) on the topology we get from the algorithm explained in Section VI-B. We sequentially change this topology and solve the multi-commodity flow problem on the resulting topologies to get an improvement in the throughput. The algorithm for changing the topology is as explained below:

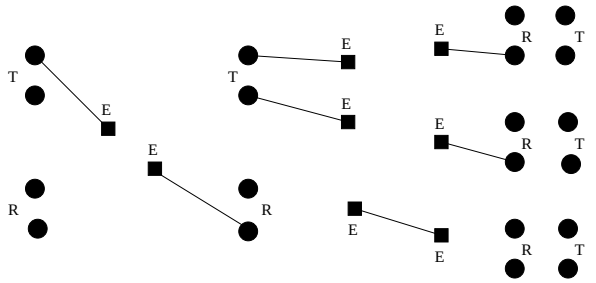


Fig. 5. Matching algorithm output

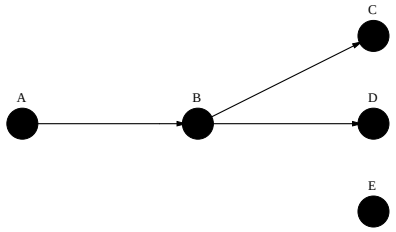


Fig. 6. Final topology

- Make a list of the profile entries for which we could route less than $x\%$ of the demand in decreasing order of demand.
- For the first entry in this list, find (maximum) K shortest paths each in the potential topology and the current topology. Form the first path that is present in the potential topology, but not in the current topology by deleting the least loaded links (using routing given by the MCF) at each of the interface-starved nodes in the current topology on this path. If all the paths are the same then repeat this step for the next entry in the list.
- Solve the multi-commodity flow problem for this changed topology. If the throughput is more than the throughput in the current topology then change the current topology to this topology and update the list by deleting the entries for which we have routed more than $x\%$ on this topology. If the throughput is less than before, then let the current topology remain the same and start with step 2 for the next entry in the list.

We finally keep the topology we have at the end of this procedure.

C. Simulation Results and Discussion

We simulate two types of networks. In the first set of simulations, all nodes in the network can be ingress or egress nodes. In the second set, only a small subset of nodes in the network are ingress or egress nodes.

1) Simulation set 1: The network used for simulations was assumed to have the parameters listed in Table IV.

TABLE IV
SIMULATION PARAMETERS

Number of nodes in the network (uniformly distributed)	20
Average degree of each node (no. of potential neighbors)	6.5
Number of receive interfaces at a node	3
Number of transmit interfaces at a node	3
Link capacity (unidirectional)	100
Number of nodes capable of being a source/destination	20
Number of source-destination pairs	160
Aggregate traffic between each pair, uniformly distributed	1 - 40
No. of shortest paths considered in rollout algorithms	4
Threshold (x) for sequential topology change	20
No. of shortest paths considered for weighting in matching	3

The simulation was run on a different random network and random profile 10 times and in each simulation, the network topology was formed starting with these parameters. The matching algorithm used is an implementation of Gabow's N-cubed weighted matching algorithm [18].

For comparison, we implemented rollout algorithms followed by the MCF formulation of Section VI-A (that we call extended rollout algorithms), in which the rollout algorithms are executed, and the topology is taken to be the output of those algorithms. The MCF is solved on that topology to get a multi-path routing. Table V shows the average bandwidth guarantees for the extended rollout algorithms. Table VI shows the average bandwidth buarantees for the matching based algorithms. As we can see, the algorithm with *Traffic Weighted Matching (TWM)* works the best, followed by *Flow Weighted Matching (FWM)*, *Uniform Weighted Matching (UWM)*. *Traffic Weighted Matching* works 8.88% better than the heuristic for rollout and 2.25% better than the integrated rollout. Thus, TWM and FWM work better than the extended rollout algorithms, and the execution time is much less than the rollout algorithms (discussed in next section). Also, rollout-based algorithms work better than UWM.

Table VI also shows the average bandwidth guarantees we get in the proposed algorithms without using the sequential topology change strategy (we call the corresponding algorithms *UWM1*, *FWM1*, *TWM1*). The results with and without sequential change show that the improvement in bandwidth guarantees is the maximum in *UMW1* followed by *FWM1* and *TWM1*. This, along with the results of Table VI, shows that *TWM* works the best among these three strategies, and weighting strategies of TWM and FWM are good for this network model.

2) Simulation set 2: In this set of simulations, the networks have 50 nodes and the number of nodes capable

TABLE V
AVERAGE BANDWIDTH GUARANTEES FOR EXTENDED ROLLOUT ALGORITHMS

Heuristic	Route Rollout	Index Rollout	Integrated Rollout
0.7090	0.7223	0.7335	0.7550

TABLE VI
AVERAGE BANDWIDTH GUARANTEES FOR MATCHING BASED ALGORITHMS

UWM	FWM	TWM	UWM1	FWM1	TWM1
0.7178	0.7606	0.7720	0.6650	0.7294	0.7535

TABLE VII
AVERAGE BANDWIDTH GUARANTEES FOR MATCHING BASED ALGORITHMS

UWM	FWM	TWM	UWM1	FWM1	TWM1
0.9045	0.8749	0.8797	0.8436	0.8698	0.8742

TABLE VIII
AVERAGE BANDWIDTH GUARANTEES FOR EXTENDED ROLLOUT ALGORITHMS

Heuristic	Route Rollout	Index Rollout	Integrated Rollout
0.8935	0.9222	0.9247	0.9400

of being source/destination is 12. The number of source-destination pairs is fixed at 90. The nodes have an average degree of 7.5. All other parameters are the same as in simulation set 1.

The simulation was run on a different random network and random profile 10 times, and in each simulation the network topology was formed starting with these parameters.

Table VII shows the average bandwidth guarantees for the matching based algorithms. The algorithm with *UWM* works the best, followed by *TWM* and *FWM*. Table VIII shows the average bandwidth guarantees for the extended rollout algorithms. Results show that the extended rollout algorithms work better than the matching-based algorithms. Thus, in networks with small number of source/destination nodes, the weighting strategies of TWM and FWM are not suitable, and rollout-based algorithms work better than the proposed matching-based algorithms. The reason seems to be that the rollout algorithms are sequential, and treat each source-destination pair separately, while the matching-based algorithms compute a maximum total weight topology, with weights heuristically dependent on the traffic and expected routes. In case of relatively small number of source-destination pairs, the proposed weighting strategies as well as the idea of looking at total weight according to these strategies do not seem to work as well as treating those separately (as in rollout algorithms).

Table VII also shows the average bandwidth guarantees we get in the proposed algorithms without using the sequential topology change strategy. The results with and without sequential change show that the improvement in bandwidth guarantees is the maximum in *UWM*, followed by *FWM* and *TWM*.

The problem of finding a degree constrained connected topology is NP-Hard [20]. In the topologies computed by our algorithms, traffic could be routed between all ingress-egress pairs in almost all cases.

VII. COMPUTATIONAL COMPLEXITY

A. Topology Control and Single-path Routing Algorithms

We first discuss the worst case time complexity of single-path routing algorithms. Let the number of nodes in the network be N, number of (potential) edges be E $(= O(N^2))$ and the number of aggregate demands in the traffic matrix be M. We use a modified version of Dijkstra's shortest path algorithm [14] as a heuristic for finding the shortest paths. It is modified to take care of the interface and bandwidth constraints while finding a shortest path. As the topology at any intermediate state of the algorithms is not expected to be sparse, so the process of finding a shortest path takes $O(N^2)$ time. We sort the profile by decreasing order of traffic demands, which takes $O(MlogM)$ time. The time complexity of the heuristic algorithm is $O(MN^2)$, as shortest paths are computed M number of times. If the set of source/destination nodes is constant, then so is the number of aggregate demands. In this case, the complexity becomes $O(N^2)$.

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. In the case of fixed M, the complexity is $O(N^2)$. 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. This also reduces to $O(N^2)$ for fixed M. The complexity for the integrated rollout is the same as index rollout time is scaled by K, which is a constant.

In the case where each node in the network can be a source or a destination, M scales as N^2 and the complexity of the heuristic algorithm becomes $O(N^4)$, while the route rollout algorithm takes $O(N^6)$, and the other two rollout algorithms take $O(N^8)$ time.

B. Complexity of Matching-based Algorithms

The computational complexity of the algorithms can be broken into the following steps:

1) Weighting: It calculates shortest paths (takes $O(N^2)$ time for each path calculation) for $O(M)$ flows. So, this takes $O(MN^2)$ time.
2) Maximum Weight Matching: There are N vertices, $O(N^2)$ edges in the original graph G. In the graph G' (input to matching), we create $O(N+N^2)$ vertices from the vertices and edges of graph G, and add $O(N)$ vertices that form a clique. So, number of vertices in G' is $N' = O(N^2)$. In G', there are $O(N^2)$ edges between the vertices corresponding to edges and vertices in G. There are $O(N^2)$ edges in the clique that we add. There are edges between all vertices of the clique ($O(N)$ vertices) and all $O(N)$ vertices that correspond to vertices in G. So, number of edges in G', $E' = O(N^2)$. Maximum weight matching takes $O(E'N'logN')$ time and thus matching takes $O(N^4logN)$ time.

The maximum order of M is $O(N^2)$, and so the maximum weight matching step determines the order of the proposed algorithms (not counting the time taken to solve the MCF). Thus, these parts of the algorithms take $O(N^4logN)$ time, with N being the number of nodes in the network. This complexity is much lower than that for route rollout ($O(N^6)$) and index and integrated rollout algorithms ($O(N^8)$). The heuristic used by these rollout algorithms takes $O(N^4)$ time.

The worst case time complexity of the multi-path algorithms is dominated by the worst case complexity of solving the linear program (LP), which takes $O((MN+E)^{3.5})$ time using interior point methods [8]. This worst case bound is misleading as current LP-solvers like CPLEX [10] are very fast for most instances of moderately sized linear programs, and thus the multi-path algorithms are much faster than the single-path rollout algorithms in our simulations. To summarize, we tabulate the computational complexity of the proposed algorithms in Table IX.

TABLE IX
COMPUTATIONAL TIME COMPLEXITY

Route Rollout	$O(M^2N^2)$
Index Rollout	$O(M^3N^2)$
Integrated Rollout	$O(M^3N^2)$
Matching Algorithms	$O(N^4 \log N)$

VIII. CONCLUSION

We consider the problem of integrated topology control and routing in Free Space Optical (FSO) wireless mesh networks. FSO links have characteristics that make them more suitable than RF links for use in mesh networks, where certain bandwidth guarantees need to be provided to end-users. As the number of FSO transceivers at each node is scarce, we consider the critical problem of topology control so that the network throughput is high for a given traffic estimate. We prove the problem NP-Hard and propose efficient algorithms for topology control and routing (both single-path and multi-path). The simulation results show significant gains over simple algorithms.

REFERENCES

[1] "IEEE 802.15 standard group," *http://www.ieee802.org/15/*.
[2] "IEEE 802.16 standard group," *http://www.ieee802.org/16/*.
[3] J. Akella, C. Liu, D. Partyka, M. Yuksel, S. Kalyanaraman, and P. Dutta, "Building blocks for mobile free-space-optical networks," *IFIP/IEEE International Conference on Wireless and Optical Communications Networks (WOCN)*, 2005.
[4] J. Akella, M. Yuksel, and S. Kalyanaraman, "Error analysis of multi-hop free-space-optical communication," *IEEE International Conference on Communications (ICC)*, vol. 3, pp. 1777–1781, 2005.
[5] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47(4), pp. 445–487, 2005.
[6] D. Banerjee and 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, 2000.
[7] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2000, vol. 1.
[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2003.
[9] J. Cao, D. Davis, S. V. Wiel, and B. Yu, "Time-varying network tomography: Router link data," *American Statistical Association Journal*, vol. 95, pp. 1063–1075, 2000.
[10] CPLEX, *http://www.cplex.com*.
[11] J. Derenick, C. Thorne, and J. Spletzer, "Control system analysis for ground/air-to-air laser communications using simulation," *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 3990–3996, 2005.
[12] A. Desai, J. Llorca, and S. Milner, "Autonomous reconfiguration of backbones in free space optical networks," *IEEE MILCOM*, pp. 1226–1232, 2004.
[13] A. Desai and S. Milner, "Autonomous reconfiguration in free-space optical sensor networks," *IEEE JSAC Optical Communications and Networking Series*, vol. 23, no. 8, pp. 1556–1563, 2005.
[14] E. W. Dijkstra, "A note on two problems in connection with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.

[15] N. G. Duffield and M. Glossglauser, "Trajectory sampling for direct traffic observation," *IEEE/ACM Trans. on Networking*, vol. 9(3), pp. 280–292, 2001.

[16] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving traffic demands for operational IP networks: Methodology and experience," *IEEE/ACM Trans. on Networking*, vol. 9(3), pp. 265–279, 2001.

[17] A. Feldmann, J. Rexford, and R. Caceres, "Efficient policies for carrying web traffic over flow-switched networks," *IEEE/ACM Trans. on Networking*, vol. 6(6), pp. 673–685, 1998.

[18] H. N. Gabow, "Implementation of algorithms for maximum matching on nonbipartite graphs," *Ph.D. thesis, Stanford University*, 1973.

[19] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-end performance and fairness in multihop wireless backhaul networks," *ACM Mobicom*, pp. 287–291, 2004.

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

[21] P. C. Gurumohan and J. Hui, "Topology design for free space optical networks," *ICCCN*, 2003.

[22] J. Hauser, "Draft PAR for IEEE 802.11 ESS mesh," *IEEE Document Number IEEE 802.11-03/759r2*.

[23] T.-H. Ho, S. D. Milner, and C. C. Davis, "Fully optical real-time pointing, acquisition, and tracking system for free space optical link," *SPIE, Free-Space Laser Communication Technologies XVII, G. Stephen Mecherle, Ed.*, vol. 5712, pp. 81–92, 2005.

[24] H. Izadpanah, T. Elbatt, V. Kukshya, F. Dolezal, and B. K. Ryu, "High-availability free space optical and RF hybrid wireless networks," *IEEE Wireless Networks*, vol. 10, no. 2, pp. 45–53, 2003.

[25] M. Kalantari, A. Kashyap, K. Lee, and M. Shayman, "Network topology control and routing under interface constraints by link evaluation," *Conference on Information Sciences and Systems*, 2005.

[26] A. Kashyap, M. Kalantari, K. Lee, and M. Shayman, "Rollout algorithms for topology control and routing of unsplittable flows in wireless optical backbone networks," *Conference on Information Sciences and Systems*, 2005.

[27] A. Kashyap, S. Khuller, and M. Shayman, "Topology control and routing over wireless optical backbone networks," *Conference on Information Sciences and Systems*, 2004.

[28] A. Kashyap, "Profile based topology control and routing in wireless optical networks," *MS Thesis, University of Maryland*, 2004.

[29] A. Kashyap, A. Rawat, and M. Shayman, "Integrated backup topology control and routing of obscured traffic in hybrid RF/FSO networks," *IEEE Globecom*, 2006.

[30] A. Kashyap and M. Shayman, "Routing and traffic engineering in hybrid RF/FSO networks," *IEEE International Conference on Communications (ICC)*, 2005.

[31] S. Khuller, K. Lee, and M. Shayman, "On degree constrained shortest paths," *European Symposium on Algorithms (ESA)*, 2005.

[32] T. I. King, H. H. Refai, J. J. S. Jr., Y. Lee, and P. G. LoPresti, "Control system analysis for ground/air-to-air laser communications using simulation," *IEEE Digital Avionics Systems Conference (DASC)*, 2005.

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

[34] E. Leonardi, M. Mellia, and 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, 2000.

[35] J. Llorca, A. Desai, and S. Milner, "Obscuration minimization in dynamic free space optical networks through topology control," *IEEE MILCOM*, pp. 1247–1253, 2004.

[36] L. Lovász and M. D. Plummer, *Matching Theory*. North-Holland, 1986.

[37] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diott, "Traffic matrix estimation: Existing techniques and new directions," *ACM SIGCOMM Computer Communication Review*, pp. 161–174, 2002.

[38] U. N. Okorafor and D. Kundur, "Efficient routing protocols for a free space optical sensor network," *IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 251–258, 2005.

[39] R. M. Ramaswami and K. N. Sivarajan, "Design of topologies: A linear formulation for wavelength routed optical networks with no wavelength changers," *IEEE/ACM Transactions on Networking*, vol. 9, no. 2, pp. 186–198, 2001.

[40] N. A. Riza, "Reconfigurable optical wireless," *IEEE LEOS*, vol. 1, pp. 70–71, 1999.

[41] C. Singh, J. John, Y. N. Singh, and K. K. Tripathi, "Design aspects of high performance indoor optical wireless transceivers," *IEEE International Conference on Personal Wireless Communications (ICPWC)*, 2005.

[42] S. Suri, M. Waldvogel, D. Bauer, and P. R. Warkhede, "Profile-based routing and traffic engineering," *Computer Communications*, vol. 24, no. 4, pp. 351–365, 2003.

[43] P. Yan, J. J. S. Jr., H. H. Refai, and P. G. LoPresti, "An initial study of mobile ad hoc networks with free space optical capabilities," *IEEE Digital Avionics Systems Conference (DASC)*, 2005.

[44] M. O. Zaatari, "Wireless optical communications systems in enterprise networks," *The Telecommunications Review*, pp. 49–57, 2003.

[45] Z. Zhang and A. Acampora, "Heuristic wavelength assignment algorithm for multihop wdm networks with wavelength routing and wavelength re-use," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 281–288, 1995.

[46] J. Zhuang, M. J. Casey, S. D. Milner, S. A. Gabriel, and G. Baecher, "Multi-objective optimization techniques in topology control of free space optical networks," *IEEE MILCOM*, pp. 430–435, 2004.