

Routing and Traffic Engineering in Hybrid RF/FSO Networks*

Abhishek Kashyap and Mark Shayman

Department of Electrical and Computer Engineering, University of Maryland, College Park MD 20742

Email: {kashyap, shayman}@glue.umd.edu

Abstract—We provide a routing framework for hybrid RF/FSO backbone networks, utilizing the characteristics of RF and Free Space Optical (FSO) links. FSO links offer higher bandwidth and security, while RF links offer more reliability. We propose the concept of having *criticality index* for different classes of traffic and providing obscuration-tolerant paths to the traffic in a weighted max-min fair way. We provide an optimal algorithm for the case where a traffic demand can be routed along multiple paths. The problem of routing unsplittable traffic is NP-Hard, so we propose efficient heuristics for routing them. We do extensive simulations to demonstrate that our algorithms outperform the algorithms currently in use.

Keywords: *Hybrid RF/FSO Networks, Traffic Engineering, Routing, Multi-Commodity Flow, QoS, Link Obscuration, Obscuration Tolerant Paths, Fault Tolerant Paths.*

I. INTRODUCTION

Free Space Optical (FSO) links are being considered as an alternative to optical fibers for deployment in Metropolitan Area Networks (MANs) due to their attractive characteristics [1], [2]. FSO links are very easy to deploy and have high capacity; ease and low cost of deployment making them suitable for backbones in military applications and also for MANs and extension of existing MANs. The attenuation in FSO links is very high in fog and snow, thus disrupting the traffic flowing through the affected links. RF links are more reliable than FSO links, and can be deployed as easily as FSO links, but are not considered for backbone networks due to their low capacity and low security.

Fiber being too expensive and time-consuming to deploy, and considering the pros and cons of FSO and RF links, hybrid RF/FSO networks have been gaining attention as they combine the characteristics of both RF and FSO links and thus can lead to a reliable, high-capacity and easy to deploy backbone.

RF and FSO links can be deployed simultaneously in two ways: the transmitters and receivers at the nodes can either have a common aperture for the two links, thus making the two networks formed by them identical; or the nodes can have different apertures for the two types of links, thus breaking the network into two (one corresponding to RF, and the other FSO). Currently, the transmitters and receivers have been developed to support a common aperture for both the link types [3], [4], and we consider this type of network.

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

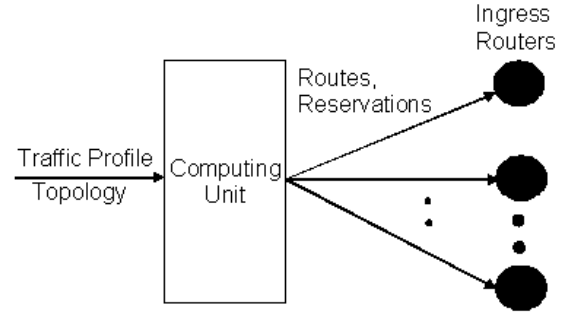


Fig. 1. High level routing architecture

Commercial products ([3], [4]) using common aperture use RF links as a backup for FSO links by transmitting identical data on both RF and FSO links, and switching to the incoming data on RF links when FSO links are obscured. In this strategy, the RF links carry traffic on a FCFS basis, thus providing backup for traffic irrespective of its characteristics; i.e., it would not give priority to traffic which is more sensitive to potential delays due to link failures. RF links have a much lower capacity (1:25) than FSO links, so they can provide backup only to a very small fraction of traffic. Upon link failure, traffic that does not have backup must be rerouted, thereby incurring a delay. We propose a framework in which we characterize the traffic to make the RF links more available to traffic which is more delay sensitive.

We introduce the concept of criticality of traffic classes. A more *critical* class of traffic is more sensitive to delay, and thus requires more RF bandwidth for backup than a less *critical* traffic class. Real time applications like VoIP are an example of very critical traffic and email traffic is an example of less critical traffic.

To the best of our knowledge, this is the first attempt at providing a routing framework for hybrid RF/FSO networks. We propose a routing framework which enables traffic engineering on the incoming traffic. We use the framework proposed in [5], [6] for integrating traffic engineering and bandwidth management at an offline computing unit and using that information at ingress routers for traffic engineering and routing when the network is online, as shown in Fig. 1. This gives the option of exercising admission control at the ingress routers and managing link bandwidth effectively. The input to the offline computing unit is the network topology and traffic profile, which consists of aggregate traffic demand

between source-destination (SD) pairs. The traffic profile can be estimated from the previous use of the network or computed from the service level agreements (SLAs). The computing unit computes routes and bandwidth reservations for the given traffic profile based on some optimization criteria (like network throughput or link congestion) and forwards the information to ingress routers, which then use that for routing and traffic engineering. We assume we have traffic of different classes (similar to MPLS networks), with each class having a criticality measure, which we call *criticality index*.

The optimization criteria we consider is maximizing the fairness we can achieve in routing the traffic (according to the fairness criteria we define, which is similar to weighted max-min fairness, [7], [8]) and then maximizing the throughput for that level of fairness.

We consider two classes of problems: one where the traffic for a given class and SD pair can be split on multiple paths, and another in which the traffic has to go through a single path. For the splittable case, we find an optimal solution by formulating the problem as a series of multi-commodity flow (MCF) problems. For the unsplittable case, the problem is NP-Hard as it is a generalization of the disjoint-path problem, which is NP-Hard [9]. So, we provide efficient heuristics to solve them.

The paper is organized as follows: Section 2 gives the network model and problem definition. Section 3 discusses the framework we follow to compute the routes and bandwidth reservations. Section 4 gives the algorithms for routing splittable traffic. Section 5 gives the algorithms for the unsplittable case. Section 6 contains the simulation results and discussion, followed by the conclusion in Section 7.

II. NETWORK MODEL AND PROBLEM DEFINITION

The network we consider is a hybrid RF/FSO backbone network, with each node capable of doing routing. Each node has a limited transmission range, which limits the number of nodes it can connect to, which we call its neighbors. The transmitters and receivers at each node have the same aperture for RF and FSO links. Thus, if we form an RF link between two nodes, the corresponding FSO link is also formed, and vice versa. The links are assumed to have a limited capacity, with the capacity of FSO links being much greater (25:1) than the capacity of RF links. The FSO links can be obscured because of fog, snow, clouds and flying birds. (We do not take the effects of scintillation into account.)

We assume as an input a traffic profile, which consists of the source-destination (SD) pairs and the expected traffic between them. Each profile entry i (which we call a traffic demand) has four elements: $\{source_i, dest_i, b_i, \Gamma_i\}$: $source_i$ and $dest_i$ are the ingress and egress nodes of the profile entry, b_i is the estimated demand for the profile entry, and Γ_i is the criticality index of the traffic profile entry, which represents the importance of this profile entry with respect to other profile entries. If $\Gamma_i > \Gamma_j$, then profile entry i is more critical than profile entry j . It could be assigned by the backbone managing entity based on the kind of traffic the profile entry is expected

to carry (e.g., VoIP traffic requires immediate backup in the case of link obscuration, so is more critical as compared to HTTP/FTP traffic). It can also be based on the price the users are willing to pay per unit bandwidth; thus if the user for a profile entry is paying more, then that traffic will have a higher criticality index. Note that a SD pair can have multiple profile entries with different criticality indices and bandwidth demands. Another option of assigning the criticality indices is by the fraction of traffic which is delay sensitive in each profile entry. As an example, this might be the case when we have just two classes of traffic: real-time traffic and email traffic. The criticality index of each profile entry will represent the fraction of real-time traffic in the profile entry. Our framework and algorithms are consistent with both the views of criticality index.

The problem we solve is to compute routes and bandwidth reservations for the given traffic profile. The objective is to provide duplicate backup paths to a fraction of each traffic profile entry according to its criticality index, and be max-min fair to each traffic profile entry in terms of how much traffic we route for each of them. The secondary objective is to maximize the throughput while satisfying the primary objectives. Then, we try to maximize the fairness level and throughput (as secondary objective) for the residual traffic profile (for which we cannot provide backup).

The routes and bandwidth reservations are forwarded to the ingress router (source) of each SD pair, which then uses it for routing and traffic engineering.

III. ROUTING FRAMEWORK

A. Fairness Index

We start by defining the fairness metric we use. For the case where the criticality index (Γ) of all the profile entries is the same, we define the fairness metric as the minimum of the fraction of each profile entry which we are able to route. For each profile entry i , if we have a demand of b_i , and we are able to route r_i , then the *fairness index* (ϕ) of the algorithm, which represents the fairness level, is as defined in Equation 1.

$$\phi = \min_i \{\phi_i : \phi_i = \frac{r_i}{b_i}\} \quad (1)$$

We extend this definition to the case when the criticality indices are not the same. We first modify the criticality index by normalizing it so the maximum criticality index is 1. We write the normalized criticality index as γ . We then define the fairness index as in Equation 2.

$$\phi = \min_i \{\phi_i : \phi_i = \frac{r_i}{\gamma_i b_i}\} \quad (2)$$

The maximum value of γ_i is 1 and r_i cannot exceed b_i , so ϕ cannot be greater than 1, as can easily be seen from Equation 2. Note that ϕ_i can be greater than 1 for profile entries having γ_i less than 1.

B. Framework

We compute the routes and bandwidth reservations in two phases. The first phase consists of computing bandwidth reservations on RF links and duplicating them on FSO links (resulting in obscuration-tolerant paths). This is done as RF links have a much lower bandwidth than FSO links. The second phase routes the remaining traffic on the residual network (with each link's capacity being the sum of residual RF and FSO capacities). The algorithm is outlined below:

- 1) **(Phase I)** Compute routes and bandwidth reservations on the network consisting of only RF links.
- 2) Duplicate the resulting bandwidth reservations on the FSO links.
- 3) **(Phase II)** Construct a residual network with each link representing the sum of unreserved bandwidths on RF and FSO links (corresponding to each pair of neighbors).
- 4) Construct the residual traffic profile by subtracting the demands for which reservation has been done.
- 5) Set the criticality index of all profile entries to 1.
- 6) Give this residual network and the residual traffic profile as an input and compute the routes and bandwidth reservations for these. For the unsplittable case, routes are the same as computed in Step 1.

Note that we follow different fairness guidelines while doing bandwidth reservations for the the residual traffic. The reason is that we are not providing any guarantees for this traffic in the event of link obscuration, thus all demands should be given equal priority irrespective of their criticality indices.

C. Routing Decision at Ingress Routers

The routes and bandwidth reservations are given to the ingress routers, which use them for routing and admission control when the network is functional. When a new flow arrives, the ingress router determines the path for the flow, if it decides to route it. The process is explained below.

- 1) Check if sufficient bandwidth is available for this traffic class of this SD pair from Phase I reservations of the offline algorithm. Route on the RF and corresponding FSO links (RF links are backup here).
- 2) Else, check if sufficient bandwidth is available from the bandwidth reservations made in Phase II and route on the corresponding path. We will prove later that this path cannot consist of purely RF links (as otherwise it would have been included in Phase I).
- 3) If the flow cannot be routed, check for unreserved bandwidth on the links. Route if a path with enough bandwidth exists, else block the flow.

IV. ROUTING SPLITTABLE TRAFFIC

In this section, we consider the case where traffic can be split over multiple paths. We formulate the routing and bandwidth reservation problem as a series of multi-commodity flow (MCF) problems, treating each profile entry as a commodity, which can be split over multiple paths. The first MCF formulation is for maximizing the fairness index. The

second MCF maximizes the throughput of the network for the computed value of fairness index.

Let there be M commodities (the value of commodity i is the profile entry demand b_i), N nodes and L links in the network. We add a dummy link (infinite cost, infinite 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 $cost(l)$ represent the cost of each link, which is 1 for all actual links in our problem. Let the set of incoming and outgoing links at node j be denoted by in_j and out_j respectively. Let ain_j denote the actual incoming links at node j , and $aout_j$ denote the actual outgoing links at j . Let $source_i$ and $dest_i$ represent the source and destination of profile i . Equation 3 achieves the objective of maximizing the fairness index. Equation 4 represents the bandwidth constraints. κ_l represents the capacity of link l . Equations 5 and 6 represent the flow conservation laws at transit nodes and source nodes for each commodity respectively. The fairness constraints are represented by Equation 7. Equation 8 gives the bounds on the variables. The traffic that goes over the dummy links is the traffic that is not routed in the actual network.

$$\text{maximize } \phi \quad (3)$$

$$\sum_{i=1}^M x_i(l) \leq \kappa_l, \quad \forall l \in \{1, \dots, L\} \quad (4)$$

$$\sum_{l \in ain_j} x_i(l) = \sum_{l \in aout_j} x_i(l), \quad \forall j \in \{1, \dots, N\} - \{source_i, dest_i\}, \forall i \in \{1, \dots, M\} \quad (5)$$

$$\sum_{l \in out_j} x_i(l) - \sum_{l \in in_j} x_i(l) = b_i, \quad j = source_i, \forall i \in \{1, \dots, M\} \quad (6)$$

$$\sum_{l \in aout_j} x_i(l) - \sum_{l \in ain_j} x_i(l) \geq \phi \gamma_i b_i, \quad j = source_i, \forall i \in \{1, \dots, M\} \quad (7)$$

$$x_i \geq 0, \quad 0 \leq \phi \leq 1 \quad (8)$$

Once we have the maximum value of ϕ (which we denote by Φ), we solve the MCF formulation for maximizing the throughput with the same constraints as in the previous formulation (with ϕ replaced by Φ in Equation 7), and the objective as in Equation 9. Equation 9 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. Along with maximizing the throughput, the objective function also

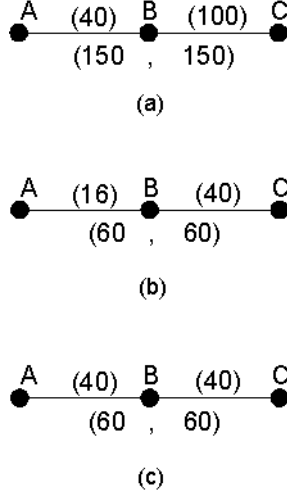


Fig. 2. (a) Example network and demands, (b) Reservations after MCF 1 (c) Reservations after MCF 2

minimizes the weighted hop count (the number of links used for each path) for the value of throughput it achieves.

$$\text{minimize} \quad \sum_{l=1}^{L+M} (\text{cost}(l) \sum_{i=1}^M x_i(l)) \quad (9)$$

We give an example to show the potential improvement by the second MCF. Fig. 2(a) shows a network with traffic demand of 40 units between $A \rightarrow B$, 100 units between $B \rightarrow C$ and 150 units between $A \rightarrow C$, all having the same criticality index. The link capacity is 100 units for each link in each direction. Fig. 2(b) shows the bandwidth reservations on the links after the first MCF. It achieves a fairness index (ϕ) of 0.4. As we can see, we could route an additional 24 units of traffic from $A \rightarrow B$, but as that would not increase the value of ϕ , the algorithm may not do that. Thus, we need to run the second MCF with $\phi = 0.4$, and get the reservations as in Fig. 2(c).

The second MCF gives a throughput at least as high as the first MCF, as in the worst case it will always output the routes and reservations which resulted from the first MCF.

A. Routing Algorithm

We have as an input a network with RF and FSO links and the traffic profile (with each entry being a quadruple). We follow the two-phase algorithm given in Section 3 as follows:

- 1) **(Phase I)** Solve the 2-step MCF formulation with the RF network and the traffic profile as the input.
- 2) Duplicate the resulting bandwidth reservations on the FSO links.
- 3) **(Phase II)** Repeat step 2 on residual network and profile.

The output of Phase II cannot have any route consisting of purely RF links, as else that route would have been included in the routes given by the 2-step MCF in Phase I of the algorithm as that is optimal.

V. ROUTING UNSPLITTABLE TRAFFIC

We now consider the case when the traffic is unsplittable, i.e., the traffic for a profile entry needs to be routed through a single path. This problem is NP-Hard as it is a generalization of the disjoint-paths problem [9]. Thus, we cannot find an optimal value for the fairness index ϕ (defined in Equation 2) and a maximum throughput for that fairness level. We resort to heuristics for achieving the objectives. We use the two-phase algorithm given in Section 3. We first compute a set of paths, and then compute bandwidth reservations in two phases (Phase I reservations have backup paths, Phase II do not).

A. Computing Bandwidth Reservations

We first describe the algorithm of computing bandwidth reservations, given a set of paths. The algorithm is a weighted version of the water-filling algorithm for achieving max-min fair routing [7], [8], so as to maximize the value of fairness index and then maximize the throughput. This algorithm is optimal for the given set of paths.

- 1) **(Phase I)** Repeat the following procedure until no more of any profile entry can be routed:
 - a) Route $\gamma_i b_i \delta$ of all profile entries i (whose computed paths have non-zero residual bandwidth). Here, δ is a very small constant (step size) so the discrete steps of routing them sequentially approximates the process of routing them at once.
 - b) Update the residual bandwidth of each link.
- 2) Duplicate the resulting bandwidth reservations on FSO links.
- 3) **(Phase II)** Repeat Step 1 on residual network and profile, using the same paths and criticality index of 1 for all profile entries.

B. Path Computation

Given the method of bandwidth reservation assignment, the more the paths are disjoint, the higher will be the value of fairness index (Φ) achieved. The problem of finding disjoint paths for a set of flows is NP-Hard [9], so we try to compute maximally disjoint paths. We start with a reasonable heuristic for path selection and use the rollout technique, [10] to improve the heuristic to obtain potentially near-optimal solutions.

1) **Basic Rollout Algorithm:** Rollout is a general method for obtaining an improved policy for a Markov decision process starting with a base heuristic policy [10]. 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 $u = (u_1, \dots, u_N)$. 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 $(u_1, \dots, u_n), (n < N)$, extends it to a complete

solution (u_1, \dots, u_N) . Let $H(u_1, \dots, u_N) = G(u_1, \dots, u_N)$. 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, \dots, u_{n-1}) and extends it by one component to $R(u_1, \dots, u_{n-1}) = (u_1, \dots, u_n)$ where u_n is chosen to maximize $H(u_1, \dots, u_n)$. 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 the increment to the value of the objective function at that decision step.

2) *Base Heuristic*: The heuristic takes the profile entries in decreasing order of $\gamma_i b_i$, so the profile entries expected to carry more traffic are routed on shorter paths. For each profile entry, it computes a shortest widest path (SWP) [11], and routes κ_{min}/M units of flow along that path, where κ_{min} is the minimum capacity of a link and M is the number of profile entries. It updates the link capacities after routing each entry. Shortest Path (SP) and Widest Shortest Path (WSP) [11], are alternatives to SWP for path computation, but we will show in the simulations that SWP outperforms these significantly.

3) *Objective Function*: The objective is to route the flows through paths as disjoint as possible. The objective function should reflect the cost of having common links between paths for different demands. We consider three objective functions (or cost), as in Equations 10 (linear), 11 (quadratic) and 12 (exponential). Here, μ_l represents the used bandwidth on link l , and L is the total number of links. $I_{\{x\}}$ is the indicator function; its value is 1 if the condition x is true, 0 otherwise.

$$\text{minimize} \quad \sum_{l=1}^L \left(\frac{\mu_l}{\frac{\kappa_{min}}{M}} - 1 \right) I_{\{\mu_l > 0\}} \quad (10)$$

$$\text{minimize} \quad \sum_{l=1}^L \left(\frac{\mu_l}{\frac{\kappa_{min}}{M}} - 1 \right)^2 I_{\{\mu_l > 0\}} \quad (11)$$

$$\text{minimize} \quad \sum_{l=1}^L \left(\frac{\mu_l}{\frac{\kappa_{min}}{M}} - 1 \right) 2^{\frac{\mu_l}{\frac{\kappa_{min}}{M}} - 1} I_{\{\mu_l > 0\}} \quad (12)$$

The cost for links carrying one flow is zero. We show via simulations that the cost given by Equation 10 works the best among these.

4) *Index Rollout Algorithm*: Index rollout seeks to optimize the order in which SWPs are computed, as the order of routing changes the SWPs. The index rollout algorithm works as follows: In the first step, the rollout algorithm computes a SWP to route the demand t_1 determined by the requirement that it minimize the cost given by Equation 10 when the base

TABLE I
AVERAGE FAIRNESS INDEX (ϕ)

SP Heuristic	0.3400
WSP Heuristic	0.5216
SWP Heuristic	0.6302
Index Rollout (Linear Cost)	0.6327
Index Rollout (Quadratic Cost)	0.6279
Index Rollout (Exponential Cost)	0.6287
Splittable Algorithm	0.6860

heuristic is used to complete the routing starting with t_1 . The base heuristic orders the remaining demands in decreasing order of $\gamma_i b_i$, and routes them sequentially.

Now, suppose that the demands (t_1, \dots, t_{n-1}) have been routed in this order by the rollout algorithm. In the next step, the rollout algorithm uses SWP to route the remaining demand t_n determined by the requirement that it minimize the cost when the base heuristic is used to complete the routing starting with (t_1, \dots, t_n) . After routing each demand, the index rollout updates the residual bandwidth of the links on the path on which this demand is routed.

At the first step, the cost for rollout is the same as that for the heuristic as we do the whole routing according to the heuristic. The method of choosing the routes makes sure that the rollout algorithm works at least as well as the heuristic, as at each decision step, it always has the choice of going according to the heuristic which gives the cost which was calculated at the previous step. Thus, the rollout performs at least as well as the heuristic in terms of the objective function.

VI. SIMULATION RESULTS AND DISCUSSION

We assume that the traffic between each SD pair for each traffic profile entry consists of flows with Poisson arrival and exponential holding times. The information given is the mean holding time (T_i), bit-rate (R_i) and Poisson arrival rate (λ_i) of each demand. The mean aggregate value (b_i) for each profile entry can be computed as in Equation 13.

$$b_i = \lambda_i T_i R_i \quad (13)$$

We generate a random network and a random traffic profile with the following parameters. In the network, each node connects to all the nodes within its transmission range, without any interface constraints.

- Size of the network = 10km x 10km.
- Number of nodes in the network = 20, five each uniformly distributed in grids of size 5km x 5km.
- Transmission range of each node = 4km.
- Capacity of each FSO link = 1000 in each direction.
- Capacity of each RF link = 40 in each direction.
- Number of source-destination pairs = 100.
- Poisson Arrival Rate for each demand (λ_i): Uniformly distributed between 20 and 40 per unit time.
- Mean of Holding Time (T_i): Uniformly distributed between 2.5 and 5 units of time.

TABLE II
AVERAGE FRACTION (MULTIPLIED BY CRITICALITY INDEX) BLOCKED/DISRUPTED BY FOG

SP Heuristic	WSP Heuristic	SWP Heuristic	Index Rollout	Splittable Algorithm	Benchmark Algorithm
0.0854	0.0580	0.0566	0.0519	0.0463	0.2051

TABLE III
MINIMUM FRACTION (NORMALIZED BY CRITICALITY INDEX) ROUTED ON OBSCURATION-TOLERANT PATHS

SP Heuristic	WSP Heuristic	SWP Heuristic	Index Rollout	Splittable Algorithm	Benchmark Algorithm
0.3118	0.4935	0.6057	0.6111	0.5976	0.2636

TABLE IV
AVERAGE FRACTION (NORMALIZED BY CRITICALITY INDEX) ROUTED ON OBSCURATION-TOLERANT PATHS

SP Heuristic	WSP Heuristic	SWP Heuristic	Index Rollout	Splittable Algorithm	Benchmark Algorithm
0.8793	0.9979	1.0256	1.0380	1.0537	0.8260

TABLE V
FRACTIONAL THROUGHPUT

SP Heuristic	WSP Heuristic	SWP Heuristic	Index Rollout	Splittable Algorithm	Benchmark Algorithm
0.8394	0.8732	0.8741	0.8802	0.8829	0.7011

- Bit Rate of individual calls = 0.1 units (same for all).
- Criticality index (Γ_i) for each demand: Uniformly distributed between 1 and 2.

The network is formed, and routes and bandwidth reservations are computed for the splittable as well as unsplittable traffic case, for 10 simulations. For the simulations, we implement two more algorithms for the unsplittable case: Shortest Path (SP) heuristic and Widest Shortest Path (WSP) heuristic. These are the same as the SWP heuristic proposed in Section V.B with shortest widest paths replaced by shortest paths (widest shortest paths for WSP heuristic). These heuristics use the algorithm of Section V.A for bandwidth reservations. Table I gives the average value of the fairness index (see Equation 2) over the 10 simulations achieved by the SP heuristic, the WSP heuristic, the SWP heuristic, the index rollout (using SWP heuristic) with linear objective function (see Equation 10), the index rollout with quadratic objective function (Equation 11), the index rollout with exponential objective function (see Equation 12) and by the splittable algorithm (proposed in Section IV). As can be seen, among the heuristics, the SWP heuristic works the best, thus we use SWP heuristic in index rollout. Also, among the unsplittable traffic algorithms, index rollout with linear cost (as in Equation 10) works the best. So, we work with the linear cost function for the online simulations. The SWP heuristic and linear cost index rollout work very close to the optimal splittable algorithm, whose output is at least as high as the optimal unsplittable algorithm. Thus, these unsplittable heuristics perform very

close to the optimal unsplittable algorithm. Note that the SWP heuristic works better than the rollout with quadratic and exponential costs (even though they (are guaranteed to) outperform the heuristic in terms of the cost functions). This indicates that those cost functions are not suitable for representing the desired objectives.

Once the routes and bandwidth reservations have been computed, we run the network for 30 units of time with the computed data as the input. Fog is assumed to be an event with Poisson arrivals with rate 0.1 per unit time, holding time uniform between 5 and 10 units. A fog event is assumed to be circular (network is assumed to be two-dimensional). The center of a fog event is uniformly distributed, and the radius is uniformly distributed between values that would make each fog event cover 5-10% of the network area. For simplicity, we do not use the unreserved bandwidth for routing in the simulations.

The algorithm we compare the proposed algorithms with is similar to the one used in commercial products [3], [4], which we call the *Benchmark Algorithm*. This algorithm works as follows: The series of MCFs given in Section IV (without criticality indices) is solved to get bandwidth reservations for FSO links (allowing splitting). When a flow arrives at an ingress node, it checks for reserved bandwidth on the FSO links and routes/blocks the flow. The RF links are used for backup on an FCFS basis (i.e., no reservations are done on the RF links). We assume the ingress router checks for available bandwidth on RF links on the whole path, and uses them as backup only if enough bandwidth is available on them on the

whole path. This is done as in our other algorithms, each flow has backup either on all the links of the path or on no link of the path.

We note four parameters from the simulations, and average them over 10 simulation runs. The first parameter is the average of the fraction of profile entries blocked/disrupted by fog (multiplied by criticality index), as shown in Table II. For each profile entry, we multiply the fraction of traffic (number of flows) blocked/disrupted by fog with its criticality index. The first parameter represents the average of this value over all profile entries. As can be seen, the splittable algorithm, index rollout, SWP heuristic work much better than the SP heuristic and WSP heuristic, and all the proposed algorithms outperform the benchmark algorithm by a significant amount. The splittable algorithm works 77% better than the benchmark algorithm, 11% better than the index rollout and 18% better than the SWP heuristic.

The second parameter (shown in Table III) is the criticality normalized minimum fraction of traffic for profile entries (similar to fairness index (ϕ) of Phase I, see Equation 2) which was routed on a backup path (i.e., routed using reservations from Phase I; and for the benchmark algorithm, routed on a backup path). For each profile entry, we note the traffic being routed using backup paths, and divide it by the criticality index of the profile entry. The second parameter represents the minimum of this value over all profile entries. We also note the average of this value over the profile entries, which is shown in Table IV. The average values can be greater than one as criticality index is less than or equal to one for all profile entries. As can be seen, the proposed algorithms work much better than the benchmark algorithm. Among the proposed algorithms, the SWP heuristic, index rollout and splittable algorithm work much better than the WSP heuristic, which works much better than the SP heuristic. The splittable algorithm outperforms the benchmark algorithm by 27.5% in the average value, and by 126% in the minimum value.

The unsplittable SWP heuristic and index rollout perform slightly better than the splittable algorithm in terms of minimum bandwidth routed with backup. This can be explained very easily with an example. Suppose we reserve 10.15 units of bandwidth for a profile entry on a path during offline phase. While evaluating ϕ_i (see Equation 2) for this profile entry during offline phase, the whole 10.15 is considered usable. In the online phase, the bit-rate of each flow is 0.1, so the 0.05 bandwidth is never used (we call it a granularity loss). In the unsplittable algorithms, there is one path per profile entry, so the maximum granularity loss is less than 0.1. For the splittable algorithms, there can be a large number of paths for some profile entries, so this loss can be very high, which leads to a lower value of minimum bandwidth routed with backup (ϕ).

Table V gives the average of the fraction of traffic routed (fraction of flows routed without being affected by fog, which we call the throughput) for each of the algorithms. As can be seen, the splittable algorithm outperforms the benchmark algorithm by 26%, and the order of performance (increasing)

is: benchmark algorithm, SP heuristic, WSP heuristic, SWP heuristic, index rollout, splittable algorithm.

VII. CONCLUSION

This paper addresses the issue of providing obscuration-tolerant paths (paths having instantaneous backup) to traffic on a hybrid RF/FSO backbone network, given a traffic profile. The capacity of RF links is much less than the backbone traffic demand, so we introduce the concept of *criticality index* for traffic profile entries, which is used in determining the fraction of traffic for each profile entry which is routed on obscuration-tolerant paths. We compute the bandwidth reservations in a weighted max-min fair way, and provide optimal algorithms for the case where traffic can be split over multiple paths. The problem of routing unsplittable traffic is proved NP-Hard, and heuristics are proposed which are shown to perform close to the optimal. The shortest widest path (SWP) is shown to be the best path computation strategy among SP, WSP and SWP. The optimal splittable algorithm gives a 26% improvement in throughput over the benchmark algorithm, and gives an improvement of 126% on the optimization variable (the criticality normalized minimum fraction routed with backup paths).

REFERENCES

- [1] N. A. Riza, "Reconfigurable Optical Wireless," *IEEE LEOS '99*, vol. 1, pp. 70-71, 8-11 Nov. 1999.
- [2] Z. Yaqoob, N. A. Riza, "Smart Free-Space Optical Interconnects and Communication Links using Agile WDM Transmitters," *2001 Digest of the LEOS Summer Topical Meetings*, 30 July-1 Aug. 2001.
- [3] <http://www.airfiber.com/news/news05292002hfr.htm>
- [4] S. Bloom, W. S. Hartley, "The last mile solution: Hybrid FSO Radio," *Airfiber*, May 2002.
- [5] Y. Wang, Z. Wang, "Explicit Routing Algorithms for Internet Traffic Engineering," *Proc. ICCCN*, pp. 582 - 588, March 1999.
- [6] S. Suri, M. Waldvogel, D. Bauer, P. R. Warkhede, "Profile-Based Routing and Traffic Engineering," *Computer Communications*, vol. 24, no. 4, pp. 351-365, March 2003.
- [7] J. M. Jaffe, "Bottleneck Flow Control," *IEEE Transactions on Communications*, vol. COM-29, no. 7, pp. 954-962, July 1981.
- [8] D. P. Bertsekas, R. Gallager, "Data Networks," 2nd Edition, Prentice-Hall, 2000.
- [9] M. Garey and D. Johnson, "Computers and Intractability: A Guide to the theory of NP-Completeness," Freeman and Company, 1979.
- [10] D. P. Bertsekas, "Dynamic Programming and Optimal Control," vol. 1, Athena Scientific, 2000.
- [11] R. Guerin, A. Orda, D. Williams, "QoS Routing Mechanisms and OSPF Extensions," *IETF Internet Draft*, 6 November 1996.