Minimum Interference Algorithm for Integrated Topology Control and Routing in Wireless Optical Backbone Networks

Fangting Sun and Mark Shayman
Department of Electrical and Computer Engineering, University of Maryland, College Park, MD 20742

***Abstract*— We consider a wireless backbone network with free space optical point-to-point links. Each backbone node has a limited number of transceivers with which to establish links to neighbors. Requests for aggregate bandwidth between pairs of backbone nodes arrive one-by-one and future demands are unknown. When a demand arrives, a bandwidth guaranteed path is established between the source and destination backbone nodes. Each time a path is established, it appropriates resources–link bandwidth and transceivers–that might be needed for future demands. The problem we consider is that of determining how to choose the bandwidth guaranteed paths in order to minimize the likelihood that future demands will be rejected due to lack of resources. The algorithm we propose is distinguished by its taking into account the potential interference with future demands caused by the reduction of the number of available transceivers when new links are established. Through simulations, we demonstrate that the performance of the new algorithm is superior to existing alternatives.**

I. Introduction

Free space optics is attracting interest as an alternative to radio for links in certain wireless networks. Such technology is expected to provide unprecedented bandwidth, massive carrier reuse, ultra-low interchannel interference, low power consumption, and cost savings where electrical wires and optical fibers are too expensive to deploy and maintain [1]. A key distinguishing feature of optical wireless networks is that the links are point-to-point rather than broadcast.

Topology control and routing are essential concerns in wireless networks. In a broadcast wireless network, there are two aspects of topology control. The first is to control power to regulate the transmission range in order to reduce interference between transmissions. This determines which nodes are within the transmission range of a given node and hence which *potential* links are present. The second aspect is the selection of which potential links to use for transmission; such links will be called *actual links*. Although considerable research has been done on topology control for wireless networks, it mostly focuses on broadcast networks [2], [3]. The topology control problem in a point-to-point network is significantly different than that in a broadcast network. In a broadcast network, a node can establish links with all of its potential neighbors. In contrast, in a point-to-point wireless network, the number of links a node may establish is limited by the number of transmitting and receiving interfaces (transceivers) it has. This interface constraint considerably changes the nature of the algorithmic problem as described below.

This research was partially supported by AFOSR under grant F496200210217.

There are also important differences between topology design for reconfigurable *wireline* optical networks and topology control for wireless optical networks. For wireline networks, transmission range is not a major issue. Also, if the optical layer has sufficient resources so the routing and wavelength assignment problem is always solvable, then whenever a source and destination both have an available interface, a direct connection (one logical hop) can be established. On the other hand, in the wireless case, unless the destination is within the transmission range of the source, a multihop connection is required. For these reasons, many published results on logical topology design for wireline optical networks [4], [5], [6], [7] are not directly applicable to free space optical networks.

There have been many routing protocols developed for mobile ad hoc networks [8], [9]. The focus is generally on nodes with significant mobility and the routing is generally done on a packet-by-packet basis. Our focus is different. We are interested in routing for networks with optical wireless *backbones*. Such a network could be a cellular radio network in which the base stations are interconnected by free space optical links. The optical wireless backbone consists of the base stations together with additional switching nodes. Alternatively, the free space optical backbone could be used to interconnect ad hoc networks. For these applications, the backbone nodes have limited mobility and it is reasonable to set up bandwidth guaranteed paths to carry aggregate traffic between these nodes. Since mobility is less of an issue, the holding time for paths is much greater than in the typical mobile ad hoc networks. By using bandwidth guaranteed paths, QoS can be provided on an aggregate, and hence scalable, basis.

If resources are limited, the topology should reflect the current pattern of traffic demands. However, we are assuming that demands cannot be accurately forecast in advance[1]. Consequently, the topology must be modified as demands arrive and depart. In particular, links must be established as paths are chosen to route arriving demands, so topology control is

[1]Depending on how the wireless network is deployed, it may or may not be reasonable to assume that aggregate traffic demands between backbone nodes can be estimated in advance. In a companion paper, we have developed algorithms for topology control and routing when such estimates (e.g. traffic matrix) are available [10].

integrated with routing.

The specific problem we consider is as follows: We have a geographically distributed set of nodes which are either stationary or have limited mobility. Each node has a limited number of wireless optical transmitters and receivers. A subset of nodes constitutes sources and destinations. Requests for aggregate bandwidth between pairs of backbone nodes arrive one-by-one and future demands are unknown. Each time a demand arrives, a bandwidth guaranteed path is established between the source and destination backbone nodes. The path may include potential links as well as actual (already existing) links. When the path is established the potential links become actual links and interfaces are consumed. Thus, the establishment of the wireless topology is integrated with the routing of the paths. Each time a path is established, it appropriates resources–link bandwidth and transceivers–that might be needed for future demands. The problem we consider is that of determining how to choose the bandwidth guaranteed paths in order to create minimum interference with unknown future demands–to minimize the likelihood that future demands will be rejected due to lack of resources. Note that since the demand sequence is not known in advance, an on-line algorithm is required that routes an arriving demand and establishes new links based on the current state of the network. This is in contrast to the case where estimates of the traffic demands (e.g., a traffic matrix) are known in advance so the topology and routes can be determined off-line.

The problem we are considering is related to the "Minimum Interference Routing" problem in the literature. (See e.g., [11], [12], [13].) However, it differs from the problem considered in these references in that it includes topology control as well as routing. In Minimum Interference Routing, the network topology is given, and the problem is to determine routes for the arriving demands. A route can always be chosen if the path contains sufficient residual bandwidth. In our problem, it is the *potential* topology that is given. When a demand is routed, any potential links on the path must be converted to actual links. This can only be done if interfaces are available. In [14], minimum interference routing is extended to include lightpath establishment (wavelength routing) as well as routing in the logical topology. However, that work explicitly assumes that the establishment of paths is never prevented by limitation on the number of interfaces.

Since the Minimum Interference Routing problem is NP-hard [11] and our problem is an extension of it, our problem is NP-hard. We propose a heuristic algorithm to solve the problem. In our algorithm, we exploit the knowledge of the ingress-egress pairs to select a path that does not interfere too much with potential future demands. The basic idea is to defer consuming bandwidth on certain 'critical' links or establishing new links that consume 'critical' interfaces. The notion 'critical' means that if the link is heavily loaded or the interfaces are used up, it would make it difficult or impossible to satisfy future demands.

The rest of paper is organized as follow: Section II describes the system model and defines the routing and topology control problem. The integrated algorithm is given and analyzed in section III. In section IV, we examine the performance of the algorithm proposed and compare with other algorithms. Finally, the conclusion is drawn in section V.

II. System Model and Problem Definition

In this section, we first describe and model the point-to-point wireless backbone network with interface constraints; then we present the routing and topology control problem for this network. Each node is equipped with a limited number of point-to-point wireless interfaces (transmitters and receivers), and is either stationary or has limited mobility. We assume that a unidirectional wireless communication link can be setup between each node and any node within its transmission range provided the first node has an available transmitter and the second node has an available receiver. Furthermore, there is no interference between transmissions.

All network information information, such as the existence of potential and actual links, the residual bandwidth of actual links, and the number of free interfaces on each node are known to all nodes in the network–e.g., via an extended link state routing protocol. The routing path of a request is computed at its source node. We make the simplifying assumption that after a request is routed, the network information available to each node is immediately updated and that the updating does not occupy any bandwidth.

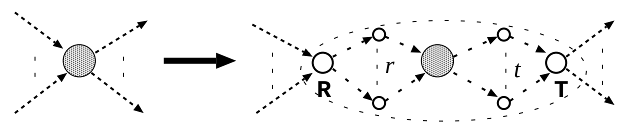
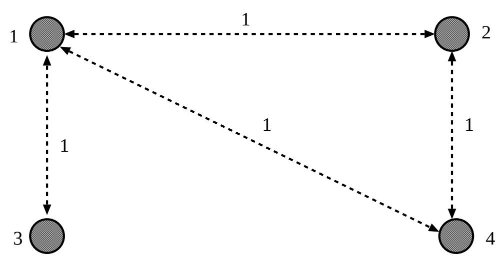


Fig. 1: Node Tranformation

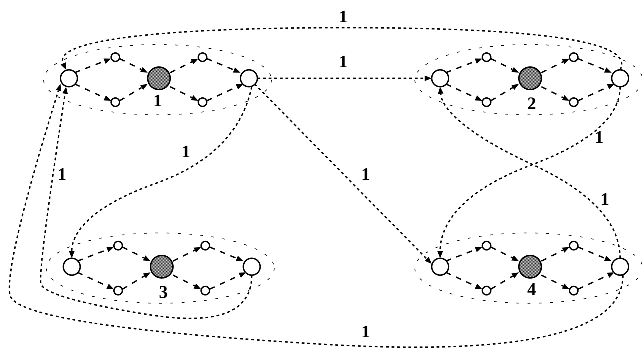
Figures 1 and 2 illustrate the key feature we use to incorporate interface constraints into the graphical modeling of the network. As shown in Figure 1, we transform a node in the network into a group of nodes, which includes a main node, a transmitter node, a receiver node, t transmitter interface nodes and r receiver interface nodes. The incoming potential links to the original node are redirected to the receiver node, and the outgoing potential links go out from the transmitter node. The main node has an interface link with every transmitter (receiver) interface node where the interface link has infinite bandwidth. The interface links also exist between the transmitter (receiver) node and every transmitter (receiver) interface node.

In Figure 2 we give an example to show how this model works. The original network is shown in Figure 2(a) which includes 4 nodes, and those nodes are connected with bidirectional potential links (dotted lines) each of whose bandwidth is 1 in each direction. Suppose each node has two transmitter interfaces and two receiver interfaces; then we model the original network as in Figure 2(b). After one request $1 \rightarrow 2$ with demand r is routed on the network, the interface links and potential link are replaced by a cut-through path and the residual bandwidth on the cut-through path is $1-r$, as shown in

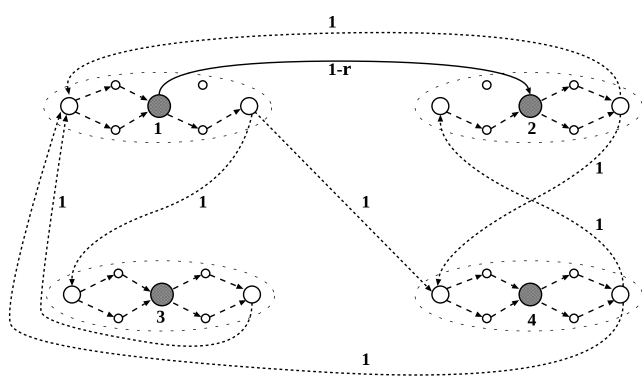
Fig. 2(c). The solid line represents the actual link. Removing the interface links means that those interfaces are occupied and cannot be used to build other point-to-point connections. Multiple connections between a pair of potential neighbors are not allowed; thus we remove the potential link as well. For future reference, we note that when the algorithms we consider require counting hops, interface links are not counted.



(a) Original Network



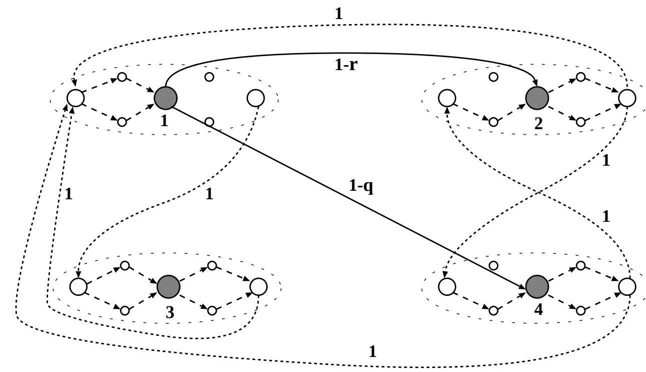
(b) Transformed Network



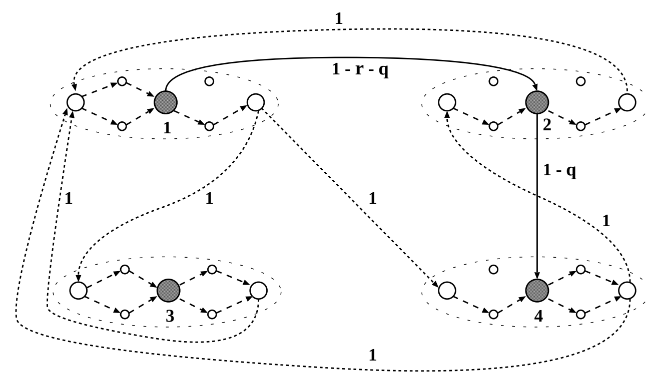
(c) Network status after one request routed

Fig. 2: Network Model

Continuing the example, now we have another setup request $1 \rightarrow 4$. There are two possible paths: 1-4 and 1-2-4, as shown in Fig. 3. The first path is a shorter one, while it uses another transmitter interface of node 1. Thus, if a request $1 \rightarrow 3$ arrives, it will be rejected. On the other hand, the second path is longer, but it will not cause requests to be rejected just because of the interface constraint. However, the second path consumes more bandwidth compared with the first one, and it may cause requests to be rejected on link 1-2 due to insufficient bandwidth. Therefore, in selecting a 'good' path for a setup request to avoid interfering too much with future demands, the algorithm must consider tradeoffs between consumption of interfaces and consumption of bandwidth on existing links.



(a) path 1-4



(b) path 1-2-4

Fig. 3: Different routing paths for request $1 \rightarrow 4$

Now we define the topology control and routing problem. Given an ad hoc network of $\mathcal{N}$ nodes interconnected by a set of unidirectional links $\mathcal{E}$ (actual, potential or interface links). A set $\mathcal{B}$ which includes information about residual bandwidth on every link, and a set $\mathcal{I}$ which stores the information about availability of each interface are maintained and updated when necessary. A list of ingress-egress pairs $\mathcal{P}$ is given and every setup request is from $\mathcal{P}$. A setup request i is defined by a triple (s_i, d_i, b_i) where s_i is the ingress node, d_i is the egress node and b_i is the amount of bandwidth required for request i. We assume setup requests come in one at a time and there is no knowledge of the characteristics of future demands. In static case, a routed request never leaves the network. While in dynamic case, a routed request will leave the network after some holding time, and as result, some unused actual links will be torn down and unused interfaces will be released. The objective is to determine a path (if one exists) along which each demand is routed so as to make 'optimal' use of network resources under interface and bandwidth constraints, and at the same time build the network topology (construct actual links).

III. Integrated Algorithm

In this section we first review the notion of minimum interference routing, then define the critical links and critical interfaces and give the weight function to evaluate their criticality. At last we present the integrated algorithm and analyze its complexity.

Since the future connection requests are unknown, when routing a setup request, we should select a path that does not interfere too much with the paths that may be critical to future requests. This concept is referred to in the literature as *minimum interference routing* [11]. Since any link or interface could be used for future setup requests, we need to decide which ones are the most important to future requests. Thus, the notion of *critical link* and *critical interface* are introduced. Once a request is routed over critical links or interfaces, the chance that some future requests can be satisfied decreases dramatically. Since some links and interfaces are more important than others, it is reasonable to assign weight to the links and interfaces to reflect their criticality. Some weight evaluation functions for *links* in wired network have been proposed. In [11], the maximum of the weighted sum of the remaining maxflows (SWUM-MAX) is used, the maximum throughput of the corresponding multi-commodity flow problem is used in [12], and [13] uses a procedure based on considering K widest-shortest paths (WSP) for each ingress-egress pair to compute the weight for each link.

In this paper, we extend the procedure used in [13] to compute the critical weight for each link (potential and actual) *and for each interface*. This procedure, *K-WSP under bottleneck and interface elimination*, computes the critical weight by making use of WSP algorithm. The procedure starts by selecting the widest-shortest path between pair (s, d). Let $\mathcal{L}p^1_{sd}$ denote the set of links constituting this widest-shortest path, and btl^1_{sd} be the corresponding available bandwidth for this path. Let $\mathcal{L}b^1_{sd}$ denote the subset of link(s) whose residual bandwidth is equal to btl^1_{sd}, and $\mathcal{I}b^1_{sd}$ denote the subset of interface links in $\mathcal{L}p^1_{sd}$. The second WSP for (s, d) is computed after the links of the set $\mathcal{L}b^1_{sd}$ and $\mathcal{I}b^1_{sd}$ are removed from the network. This procedure is repeated until either K WSPs are found or no more WSP is available.

In the above procedure, the significance of the paths is in descending order, i.e., the i^{th} WSP is more important than the $(i+1)^{th}$ WSP, since the i^{th} WSP has more residual bandwidth than the $(i+1)^{th}$ WSP. In accordance, the weights of links in the set $\mathcal{L}p^i_{sd}$ should be proportional to a factor v^i_{sd}, where v^i_{sd} is a decreasing function of i. Here we use function $v^i_{sd} = (K - i + 1)/K$.

Intuitively, a link with less residual bandwidth should be more critical. Consequently, bottleneck links in $\mathcal{L}b^i_{sd}$ should be assigned a higher weight than other links. Therefore, the weight of link $l \in \mathcal{L}p^i_{sd} \setminus \mathcal{I}b^i_{sd}$ should be proportional to a factor $u^{i,l}_{sd}$. We set $u^{i,l}_{sd} = btl^i_{sd}/r(l)$, where $r(l)$ is the residual bandwidth of link l.

Intuitively, a free interface on a node with fewer free interfaces should be more critical since using the free interface causes a greater restriction on the number of additional links that may be formed with that node. Thus we use another factor $z^l_{sd} = 1/o^2(l)$ to affect the weight of interface link $l \in \mathcal{I}b^i_{sd}$, where $o(l)$ is the number of free interfaces of l's owner node.

Another observation is that the basic weight of the general links ($\mathcal{L}p^i_{sd} \setminus \mathcal{I}b^i_{sd}$) and the interface links ($\mathcal{I}b^i_{sd}$) should be different since they are different types of links. We use w_B as the basic weight for general links and w_I for interface links. The ratio of w_I by w_B should be different for different networks since the relative importance of interfaces compared with bandwidth depends on factors such as the node density, link bandwidth and number of interfaces per node.

By taking into account all factors above, we get equations (1) and (2) which are used to compute the weights for general links and interface links respectively. a_{sd} is the weight of the ingress-egress pair (s, d) reflecting the relative importance of that pair, and $\mathcal{P}$ is the set of all ingress-egress pairs.

$$w(l) = w_B \sum_{(s,d)\in\mathcal{P}} a_{sd} \sum_{\substack{i=1 \\ l\in\mathcal{L}p^i_{sd}\setminus\mathcal{I}b^i_{sd}}}^{K} \frac{K-i+1}{K} \times \frac{btl^i_{sd}}{r(l)} \quad (1)$$

$$w(l) = w_I \sum_{(s,d)\in\mathcal{P}} a_{sd} \sum_{\substack{i=1 \\ l\in\mathcal{I}b^i_{sd}}}^{K} \frac{K-i+1}{K} \times \frac{1}{o^2(l)} \quad (2)$$

Algorithm 1 Integrated Routing Algorithm

Input: A transformed network graph $G(\mathcal{N}, \mathcal{E})$, ($\mathcal{E}$ includes actual links, potential links and interface links), a set $\mathcal{B}$ indicating the residual bandwidth for each link, a set $\mathcal{I}$ indicating the number of free interfaces for each node, a set $\mathcal{P}$ of ingress-egress pairs and a setup request (s_k, d_k, b_k)

Output:A path between s_k and d_k with bandwidth b_k units or none

1: Compute the critical link sets using K-WSP under bottleneck and interface elimination procedure for each pair $(s, d) \in \mathcal{P}$. Find sets $\mathcal{L}p^i_{sd}$, $\mathcal{L}b^i_{sd}$ and $\mathcal{I}b^i_{sd}$ for the i^{th} WSP path.
2: Assign weight $w(l)$ for each link according to equation (1) and (2).
3: Eliminate all links whose residual bandwidth is smaller than b_k.
4: Use Dijkstra's algorithm to compute min-weighted path R on the reduced network with $w(l)$ as weight of link l.
5: If R exists, reserve b_k units from s_k to d_k along R, update sets $\mathcal{E}, \mathcal{B}$ and $\mathcal{I}$.

Using the Dijkstra algorithm [15] to compute WSP, the complexity is in order of $O(n \log n + |\mathcal{E}|)$, where n is the number of nodes. So in algorithm 1, the complexity of step 1 is $O(|\mathcal{P}|K(n \log n + |\mathcal{E}|))$, step 2 is $O(|\mathcal{P}|K(n + |\mathcal{E}|))$, step 3 is $O(|\mathcal{E}|)$, step 4 is $O(n \log n + |\mathcal{E}|)$ time and step 5 is $O(n)$. Typically, K is a small number and $|\mathcal{P}|$ is constant. Therefore, the the computational complexity of the algorithm is $O(n \log n + |\mathcal{E}|)$ totally. We refer to algorithm 1 as SMIRA_I.

IV. Performance Studies

In this section, we compare the performance of our algorithm, SMIRA_I, with SMIRA ([13]), widest shortest path WSP and shortest path (SP). SP routes the traffic demand on the shortest path from source to destination consisting of both potential links and actual links. A potential link can only be used if there is an available transmitter at the head and an available receiver at the tail of the link. When a path is chosen, the potential links are converted to actual links and interfaces are consumed. Link bandwidth is decremented by the amount of the traffic demand. WSP is similar to SP but if there are multiple shortest paths, it chooses the one whose minimum residual link bandwidth is maximum. SMIRA was designed for minimum interference routing in an completely specified topology–i.e., all links are actual links. As such, it does not consider constraints on the number of transmitters and receivers. However, we can adapt it to our problem by applying it to the partially specified topology consisting of both potential and actual links, and requiring that a transmitter and receiver be available whenever a potential link is used. Adapted in this way, the difference between SMIRA and our algorithm, SMIRA_I, is that SMIRA does not assign an interference weight w_I to interface links in the transformed graph. Consequently, in contrast to SMIRA_I, SMIRA does not have a tendency to avoid paths that include potential links whose use consumes interfaces that are likely to be needed for the paths of future demands. It is also important to emphasize that in all of these algorithms, when the length of a path is computed, only the potential and actual links are counted, not the interface links. E.g., a path consisting of three potential links is regarded as a three hop path even though in the transformed graph the path also consumes six interface links. We set $K = 3$, $w_B = 1$ and $\forall(s,d) \in \mathcal{P}, a_{sd} = 1$ for SMIRA_I. The choice of w_I is left as an adjustable parameter to be optimized based on experimental results. (See below.)

The experiments are carried out using two different randomly generated networks. In both networks, there are 100 nodes randomly located on a 1000×1000 plane. In network 1 the transmission range is 150, and the transmission range is 175 in network 2. Each node has 4 transmitter and 4 receiver interfaces. Each ingress-egress set $\mathcal{P}$ includes 50 different ingress-egress pairs that are randomly selected from 100 nodes. Requests are generated randomly and are uniformly distributed among all pairs in $\mathcal{P}$. In all experiments, the bandwidth demand of each request is uniformly distributed from 1 to 3 units and is integer. The performance is measured by the number of rejected setup requests.

A. Static requests

Static requests means a routed request never leaves the network. In both networks, the initial bandwidth of each general link is 1000 units. For each of the two networks, we test 10 different ingress-egress-pair sets. For each ingress-egress-pair set we generated 10 different request sequences and computed the average number of rejected requests. In the experiments on network 1, every request sequence includes 5000 requests, while for network 2, every request sequence includes 10,000 requests. In network 2, the larger transmission range leads to more possibility to route setup requests, so we need more requests to get some rejected requests to show the performance. In both tests, we finally chose $w_I = 1$ since this w_I/w_B ratio worked better than others.

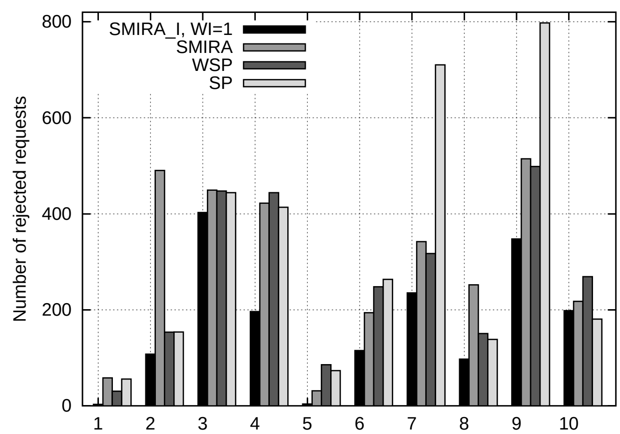


Fig. 4: Static Case: Number of rejected requests for 10 different ingress-egress-pair sets on network1

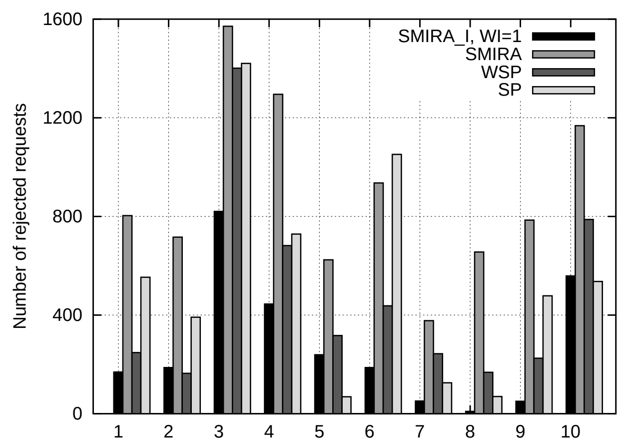


Fig. 5: Static Case: Number of rejected requests for 10 different ingress-egress-pair sets on network 2

The test results for network 1 are shown in Figure 4, and the results for network 2 are shown in Figure 5. The figures show that SMIRA_I performs the best for most ingress-egress-pair sets, and WSP follows. SMIRA_I reduced the rejection rate up to 98.6% compared with SMIRA, 95.2% compared with WSP and 94.4% compared with SP. Averaged over all the experiments, SMIRA_I reduced the rejection rate by 62.7% compared with SMIRA, 39.5% compared with WSP and 48.9% compared with SP.

B. Dynamic requests

Dynamic requests means after some holding time, a routed request will leave the network. Consequently, it may happen that after some time, there are no demands that are routed over an actual link. In this case, the link is torn down and becomes a potential link once again, and the interfaces are freed up and become available to other potential links.

The tests are conducted under the following scenario. Requests arrive between each ingress-egress pair according to a Poisson process with an average rate λ, and the holding times are exponentially distributed with mean $\frac{1}{\mu}$. In our experiments, $\frac{\lambda}{\mu} = 5$. We use the same networks as in static case, but the bandwidth for each link is 20 units. As in static case, we test 10 different ingress-egress-pair sets. For each ingress-egress-pair set we conducted 10 different 10,000 request sequences and computed the average number of rejected requests.

The test results for network 1 are shown in Figure 6, and the results for network 2 are shown in Figure 7. For network 1, we use as the interface weight factor $w_I = 3$, while for network 2, we use $w_I = 2$. The figures show that SMIRA_I performs best for every ingress-egress-pair set. SMIRA_I reduced the rejection rate up to 29.4% compared with SMIRA, 19.5% compared with WSP and 20.1% compared with SP. Averaged over all experiments, SMIRA_I reduced the rejection rate by 9.0% compared with SMIRA, 6.1% compared with WSP and 9.7% compared with SP.

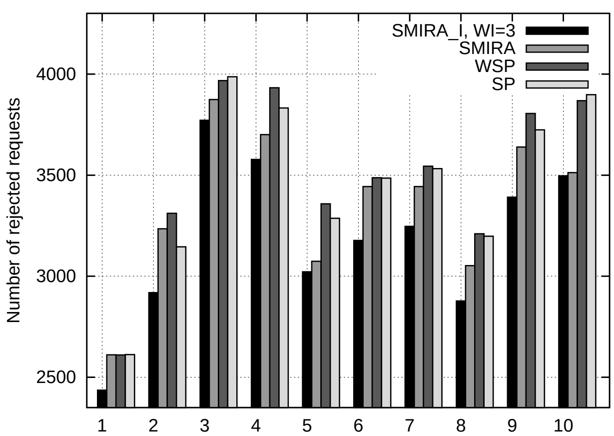


Fig. 6: Dynamic Case: Number of rejected requests for 10 different ingress-egress-pair sets on network 1

V. Conclusion

The primary contribution of this paper is the development of an algorithm for integrated topology control and routing in wireless optical backbone networks. The algorithm sets up bandwidth guaranteed paths between nodes online when the demands for such paths arrive sequentially and future demands are unknown. This work extends the concept of minimum interference routing to include topology design by modeling constraints on the number of available interfaces and introducing the notion of "interface interference." Through extensive simulation experiments, we show that the performance of this algorithm is superior to existing alternatives that do not explicitly take into account the potential interference with the accommodation of future demands caused by the consumption of available interfaces to set up new links.

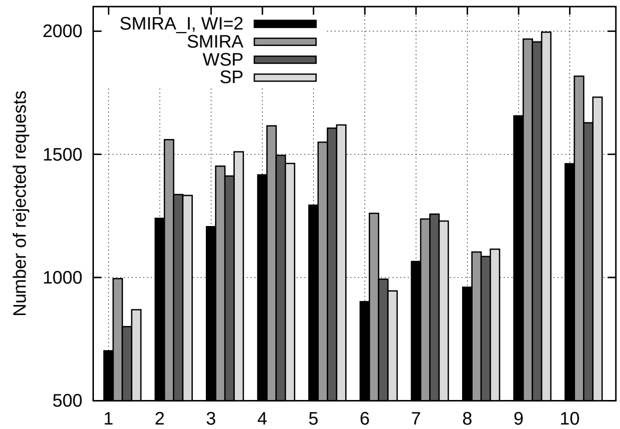


Fig. 7: Dynamic Case: Number of rejected requests for 10 different ingress-egress-pair sets on network 2

References

[1] N. Riza, Ed., *Optics in Information Systems*. Special Issue: Optical Wireless Communications, Oct. 2001, vol. 12, no. 2.

[2] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless Networks using Transmit Power Adjustment," *INFOCOM*, pp. 404–413, April 2000.

[3] R. Wattenhofer, P. B. L. Li, and Y.-M. Wang, "Distributed Topology Control for Power Efficient Operation in Multihop Wireless Ad Hoc Networks," *INFOCOM*, pp. 1388–1397, April 2001.

[4] D. Banerjee and B. Mukherjee, "Wavelength-routed Optical Networks: Linear Formulation, Resource Budgeting Tradeoffs and a Reconfiguration Study," *IEEE/ACM Trans. Networking*, vol. 8, pp. 598–607, 2000.

[5] K. H. Liu, C. Liu, J. L. Pastor, A. Roy, and J. Y. Wei, "Performance and Testbed Study of Topology Reconfiguration in IP over Optical Networks," *IEEE Trans. Communications*, vol. 50, no. 10, pp. 1662–1679, Oct. 2002.

[6] E. Leonardi, M. Mellia, and M. A. Marsan, "Algorithms for the Logical Topology Design in WDM All-Optical Networks," *Optical Networks Magazine*, pp. 35–46, Jan. 2000.

[7] R. Ramaswami and K. N. Sivarajan, "Design of Logical Topologies for Wavelength-Routed optical Networks," *IEEE JSAC*, pp. 840–851, 1996.

[8] S. Ramanathan and M. Steenstrup, "A Survey of Routing Techniques for Mobile Communications Networks," *Mobile Networks and Applications*, vol. 1, no. 2, pp. 89–104, 1996.

[9] M. Mauve, J. Widmer, and H. Hartenstein, "A Survey on Position-based Routing in Mobile Ad Hoc Networks," *IEEE Network*, vol. 15, no. 6, pp. 30–39, 2001.

[10] A. Kashyap, K. Lee, and M. Shayman, "Rollout algorithms for integrated topology control and routing in wireless optical backbone networks," Institute for Systems Research, University of Maryland, Tech. Rep. TR2003-51, 2003.

[11] K. Kar, M. Kodialam, and T. V. Lakshman, "Minimum Interference Routing of Vandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 12, pp. 2566–2579, December 2000.

[12] S. Suri, M. Waldvogel, and P. R. Warkhede, "Profile-Based Routing: A New Framework for MPLS Traffic Engineering," *Proc. of the 2 International Workshop on Quality of future Internet Services (QofIS01)*, pp. 138–157, September 2001.

[13] I. Iliadis and D. Bauer, "A New Class of Online Minimum-Interference Routing Algorithms," *Networking 2002*, vol. LNCS 2345, pp. 959–971, May 2002.

[14] M. Kodialam and T. V. Lakshma, "Integrated Dynamic IP and Wavelength Routing in IP over WDM Networks," *INFOCOM*, pp. 358–366, April 2001.

[15] E. W. Dijksra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.