Different Algorithms for Normal and Protection Paths

Rajarshi Gupta*, Eric Chi†, and Jean Walrand†
Department of Electrical Engineering and Computer Sciences
University of California, Berkeley
{guptar, echi, wlr}@eecs.berkeley.edu

***Abstract*— Many network routing situations commonly require backup paths that satisfy various constraints on bandwidth, link or node selection, and ease of configuration.**

In this paper, we attempt to validate whether it is beneficial to have distinct algorithmic treatments of normal and backup path calculation, configuration and maintenance. We present a modular suite of algorithms that enable us to manage normal and protection paths differently. We incorporate a distributed algorithm to separately calculate normal and backup paths in the network, using link state information, and present an asynchronous dynamic reorganization of backup paths to reduce congestion in the network. Simulations demonstrate quantitative reductions in blocking probabilities under certain conditions.

***Index Terms*— Optical Networks, MPLS, Protection Paths**

I. Introduction

Many network routing situations require a protection path in addition to a normal path. For example, MPLS and optical networks require backup paths that satisfy various constraints on bandwidth, link or node selection, and ease of configuration.

In this work, we examine the premise that algorithmic treatments of backup path calculation, configuration, and maintenance need not be the same as those applied to the corresponding normal paths.

Our chief contribution is a flexible and modular suite of algorithms to calculate, reserve, configure, and maintain these paths. We present a scheme by which the network operator may utilize different algorithms for normal and protection paths. Our research does not categorically adhere to any of the particular algorithms presented here – the very motivation for a modular infrastructure, in fact, is to facilitate incremental schemes.

We reuse many well-known routing and control protocol techniques to glue together our algorithms. However, in order to provide a flexible mechanism, we have also developed several novel schemes.

- A distributed method for separately calculating normal and backup paths in the network, using link state information
- Per-link state maintained at each node, that allows optimal sharing of backup bandwidth
- Asynchronous dynamic reorganization of backup paths to reduce congestion in the network.

* Rajarshi Gupta's work is supported in part by Extreme Networks, Inc.
† Research supported in part by a research grant from Darpa

We incorporate these methods and evaluate their effects on network resource utilization.

To validate our ideas, we have developed a modular simulation model that enables us to add or remove specific algorithm components, in addition to various hooks that measure call blocking probability, link loads, etc. The plan is to study the interaction of the various mechanisms (current as well as future) – which will allow us to choose the optimal suite of algorithms for a specific application scenario.

The rest of the paper is organized as follows. In Section II we describe two application scenarios for this research. In Section III, we take a closer look at how normal and protection paths differ and discuss how to exploit these differences. The algorithms are discussed in Section IV. Sections V and VI present the simulation model and a quantitative analysis of improvements gained using these methods. We close with a Conclusion and Future Works section.

II. Applications

Differentiating protection path and normal path management has immediate relevance to two cutting-edge technologies deployed in the internet today.

A. MPLS

MPLS [1] is a label-switched scheme widely deployed today to achieve quality and control on a connection carried over IP. Among its many benefits, MPLS provides an infrastructure that supports fault-redundant paths. The framework (Section III) described here is directly applicable to an MPLS network, employing an RSVP-like [2] mechanism to reserve and tear-down paths. The mechanisms described in Section IV-B can also be considered in an MPLS context.

B. Optical Network

In optical networks, wavelengths need to be allocated across optical cross-connects before traffic can flow across them. Extremely high bandwidths associated with fibers warrants the presence of a backup path to ensure rapid fail-overs. Finding suitable normal and protection paths is a common and critical problem in optical networks [3].

Since wavelengths are a limited resource on each link, the control plane must keep track of the current allocations and judiciously deal with oncoming demands. The need is for a dynamic system that handles the routing and configuration of protection paths.

C. *Generalizing the Description in the Paper*

The management of normal and backup paths need not be considered in MPLS and optical network contexts separately, since the central ideas are sufficiently general. Examples are drawn from both domains but may often be used in either one. For that matter, the ideas presented here will be useful in any other domains that require the configuration of multiple normal and protection paths through a network.

III. Normal vs. Protection Paths

In this section we describe the motivation behind a key concept in our paper: protection paths and normal paths need not be managed in the same way.

A. *Different Routing Algorithms*

Several sophisticated algorithms for the calculation of paths in a network have been developed, and much work has also gone into the computation of protection paths (e.g. [3], [4], [5]). The metric behind calculating the normal and protection paths may easily be different, depending on the application.

A simple example illustrates the point. In an optical network when a new wavelength λ needs to be routed through the network, the metric frequently used is to choose the path that has the greatest minimum number of λ available on any link [6]. When a protection path through the same network is determined, a λ is assigned so that the cumulative allocation of protection λs is minimally increased (e.g. this scheme is employed in the AT&T IP Backbone Optical Network). By re-using existing provisioned λ , it ensures that the backup path does not consume any further protection resource in the network. The optimization objectives are different for a normal and a protection path wavelength assignment.

Consequently, it might logically follow that we ought to use different algorithms for normal and protection paths, under certain situations. And in this paper, we study the merits of this design decision.

B. *Cost of Reconfiguration*

Many of today's network applications (e.g. bandwidth allocation across the country) employ greedy algorithms, because future demands are not known. And any greedy algorithm that deals with dynamic demands is liable to leave a network shouldering an unbalanced load. In today's world of guaranteed SLAs (Service Level Agreements), it is imperative that even brief service outages (caused by reconfiguration) be infrequent. So frequent reconfiguration of a live normal path for optimal redistribution of network load is out of the question.

Protection paths, on the other hand, are not subject to such constraints. A non-live protection path (i.e. one that is not actively carrying any traffic), may be reconfigured without any effect on the current traffic. Thus, we are encouraged to reconfigure these paths if a more balanced allocation of the network resources can be achieved.

C. *Decoupled vs. Coupled Algorithms*

The algorithms listed in the references above may broadly be classified into two categories – coupled algorithms, which compute the normal and protection paths simultaneously, and decoupled algorithms, which first compute the normal paths and then separately compute the protection paths. Clearly, coupled optimization techniques will always perform better than decoupled ones, because the decision-making process takes into account a larger state space. The simplicity and flexibility (Sections III-A and III-B) offered by the decoupled schemes, however, often make them desirable. In this paper, we deal only with algorithms where the calculation of normal and backup paths are decoupled.

IV. New Algorithms

In this section we present a distributed mechanism by which we can apply different treatments to the normal and protection paths.

A. *Summary of Algorithms*

The chief pieces of the schemes are as follows:

- Every source knows:
  - Topology of the network
  - Available bandwidth for every link in the network
- A new request arrives at the source and is characterized by the source and destination, as well as the normal and protection bandwidths (b_N and b_B)
- Based on its knowledge of the network, the source runs a source-based localized algorithm to generate the normal path:
  - The source makes a request along the normal path
  - All nodes in the path reserve the requested bandwidth
- The source then repeats the algorithm on a modified graph (without the links on the normal path) to generate the backup path
  - A request is sent for a protection path. The request also carries the normal path information in addition to the requested protection bandwidth
  - Each node in the backup path reserves a protection bandwidth.
- A periodic link-state advertisement scheme conveys the link state information to every node in the network
- The sources also carry out a periodic reconfiguration of the protection paths. Specifically, each source wakes up after a random amount of time and checks to see if any of its protection paths are using links that are approaching capacity. If so, it attempts to reconfigure those protection paths.

B. *Algorithm Details*

In this section, we describe in detail the algorithms that were listed above.

1) Algorithm for Normal Path: To calculate the normal path for the request, we need to employ an algorithm that is distributed and source-based. The most common such algorithms are shortest path algorithms like Dijkstra [7] and we can utilize these (see Section V). We consider variations like Minimum Interference Routing [8].

All algorithms, however, should meet certain conditions.

- A link may not reserve more traffic than it has capacity for
- Shorter paths are preferred because they consume fewer network resources
- Critical resources, e.g. residual bandwidth in bottleneck link, should be preserved for future demands

The last two conditions reflect that what we really seek is to lower the blocking probability of call requests, or equivalently increase the network utilization.

2) Link Weight Metric: A source-based routing algorithm like Dijkstra uses link weights to compute the best path. The link weights chosen essentially determine the behavior of the algorithm. The appropriate choice of the link weights and the path objective function depends on the specific network context.

3) Request-Response for Normal Path: The scheme for reserving the normal path is very similar to the RSVP protocol [2] used by MPLS. To make a reservation request, the source needs the path and the normal bandwidth that it is trying to reserve.

The request is sent by the source along with the path information. At every hop, the node determines if adequate bandwidth is available in the onward link. If the available bandwidth is inadequate, the node rejects the requests and sends a response back to the source. If the bandwidth is available, it is provisionally reserved, and the request packet is forwarded on to the next hop in the path.

If the request packet successfully reaches the destination, the destination acknowledges it by sending a reservation packet back along the same path. As each node in the path sees the reservation packet, it confirms the provisional reservation of bandwidth. In addition, it also performs the required configuration needed to support the coming traffic. This could include setting up labels for an MPLS system, or configuring the λ switching for an optical switching system.

4) Link Utilization Values: In order to accept/reject an incoming request, every node must have knowledge of its reserved bandwidth on each outgoing link. For each outgoing link, the node needs to keep track of three values:

- $B_T :=$ The total bandwidth available on this link.
- $B_N :=$ The normal bandwidth reserved on this link. This is the total amount of traffic that has been reserved on this link for normal paths.
- $B_B :=$ The protection bandwidth on this link. This is the maximum amount of bandwidth reserved on this link to support protection traffic.

The available bandwidth, B_V, on a link is calculated as

$$B_V = B_T - B_N - B_B$$

When a new request with bandwidth b_N is received by the node, capacity limits require that $b_N \leq B_V$ for every link in the path. Further, every node on the backup path needs to provision the resources required to support the backup bandwidth (b_B), in case of a failure in the normal path.

Traditional methods of calculating backup paths often do not account for the sharing of bandwidth between various backup paths and consequently utilize the network resources poorly. In [9], we have developed a method to take into account disjoint normal paths that can share their backup bandwidth, and also consider the fact that normal bandwidths need no longer be reserved along all the links if a link on the normal path fails. We utilize these methods to optimally calculate the value of B_B and thereby achieve greater network utilization.

5) Request-Response for Backup Path: The request-response mechanism for the backup path works in the same way as that of the normal path. The source sends a request along the backup path. As each node receives the request, it calculates whether or not it can accept the request. If so, it makes a provisional reservation. The difference in the request for the backup path is that in addition to the path information and the backup bandwidth, the request also carries the list of links in the normal path. This is necessary for accurately updating the link states [9].

When the acknowledgement comes back from the destination, the nodes perform the necessary configurations required to be prepared for oncoming protection traffic and update the link states.

6) Link Weight metric for Backup Path: For calculating the backup path, we could employ the same algorithm and metric as used for the normal path (Section IV-B.1), albeit with the links used for the normal path removed from the network. As discussed throughout this paper we are encouraged to explore other algorithmic options for protection path computation. For instance, the objective could be to minimize the cumulative backup resources allocated in the entire network (see Section III-A).

7) Link State Advertisements: For the calculation of normal and backup paths the sources need to know the current state of the network. Specifically, every source needs to know the available bandwidth for every link in the network. Consequently, the link-state update mechanism for our distributed mechanism needs to exchange the per-link B_V values.

Since the amount of per-link state information is very small, any appropriate link state advertisement scheme like those employed by OSPF [10] or BGP [11] should be adequate for this purpose.

C. Asynchronous Reconfiguration of Backup Paths

We employ a scheme to periodically reconfigure the protection paths to reduce congestion in the bottleneck links.

Every source carries out the following mechanism:

- Wakes up after a random interval
- Upon waking up, examines its current records of link states to identify congested bottleneck links (i.e. links whose B_V is less than some threshold)

- It checks if any backup paths originating from itself use the bottleneck link
- If so, it tries to calculate an alternate backup path. This may be calculated by routing a new path on a modified graph without the links on the normal path, as well as the bottleneck links.
- If a “better” alternative path is found, it reconfigures the backup path. An alternative backup path is said to improve the network congestion if it improves some pre-decided global metric (e.g. number of congested links)

In Section VI we present simulation results to demonstrate the benefits of periodic reconfiguration of backup paths and the situations when this mechanism ought to be used.

V. Analyzing Different Normal and Backup Routing Algorithms

A. Algorithms Simulated

To investigate claims in section III-A, we consider applying different routing algorithms to normal and protection paths. We used three different algorithms for calculating the primary path and added a fourth algorithm (Minimum Cumulative Backup) for the backup paths. We simulated all the twelve pairings on three topologies.

The four algorithms used are explained below:

1) Least Loaded: As the name suggests the routing algorithm searches for the route with the most spare capacity.

2) Minimum Hop Count: This is the standard Dijkstra’s shortest path algorithm.

3) Minimum Cumulative Backup: This chooses the path that minimizes the increase in the total amount of protection resources reserved in the network (see Section III-A).

4) Minimum Interference Routing Algorithm (MIRA): This is a Dijkstra’s shortest path algorithm. The links that are critical to source destination pairs are assigned higher weights. A link is classified as critical for a given source destination pair if routing a flow on it would reduce the maximum flow between the said source destination pair. Every source destination pair has an associated weight. The weight of a link equals the sum of the weights of all the source destination pairs for which it a critical link. Thus, critical links may be made more or less “critical” by particular source destination pairs that have greater assigned weights. In all cases when MIRA was used all source destination pairs were assigned a weight of 1. So all pairs were considered equally important.

Some explanation is needed as to how MIRA was extended to protection path routing. When a connection request arrives the MIRA extension should choose a path that interferes the least with all the maximum flows between the other source destination pairs. Interfering means reducing not just the maximum flow for primary paths but also a maximum flow for backup paths since connections are admitted only if both a primary and backup path are found. Computing an exact protection maximum flow for other source destination pairs by taking into account bandwidth sharing, however, does not make sense. This is because accurate accounting of backup resources requires an explicit primary path to have already been chosen which is not the case. Thus, only one maximum flow is computed for all the other source destination pairs when a primary route is being chosen. This is reasonable since both the primary and backup paths will be branches within the maximum flow anyway, if at least two disjoint paths exist.

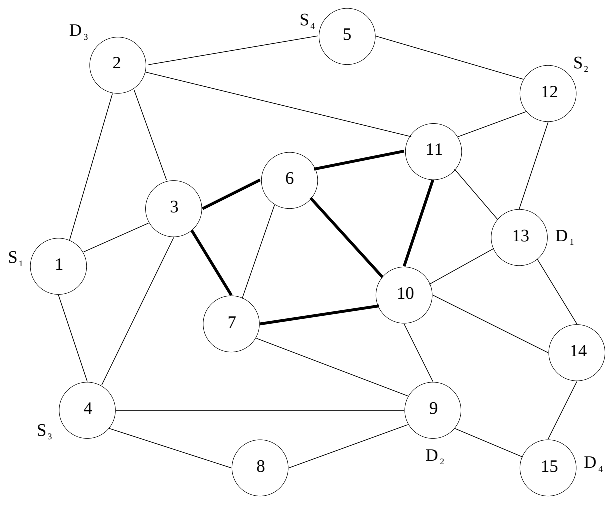

Fig. 1. Network for comparing different routing algorithm combinations

When MIRA is used to compute a backup path, an explicit primary path has already been chosen so backup bandwidth sharing can be exploited. Indeed it is possible that using a link for a primary path would reduce the maximum flow between another node pair but does not reduce the maximum flow when used for a backup path. In such a case that link is not considered critical. Thus, we maintain the spirit of MIRA in our extension.

B. Scenarios

In each of the following scenarios some subset of all the node pairs act as sources and destinations. Connection requests originate at the source for a connection request of primary bandwidth of one unit and backup bandwidth of one unit. The backup path must be link disjoint with the primary path, and the network must be robust to single link failure. Primary and backup path routing is decoupled. So a primary path is determined first. If a path is found, then a backup path is determined. If it is found, the connection is admitted into the network and appropriate primary and backup resources are reserved and held for the duration of the connection. If either primary or backup paths are not found the request is blocked.

In the first two topologies a small subset of the nodes are chosen to be source destination pairs. In the third topology, almost all node pairs act as source destination pairs.

1) MIRA Topology: Consider the network shown in Figure 1 (This is the same topology used to demonstrate the performance of MIRA [8]). Four pairs of nodes on the “edge” of the network are chosen as source destination pairs. The thin links can carry up to 6 calls while the fat links can carry up to 24 calls. Thus, the thin and fat links represent OC-12 and OC-48 links respectively. All call arrival processes are

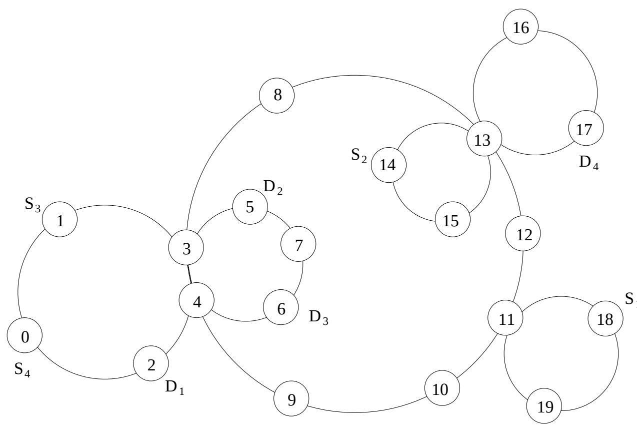

Fig. 2. MAN ring of rings

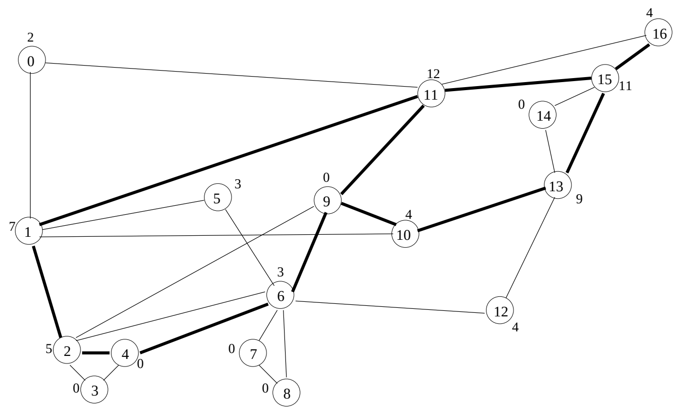

Fig. 3. AT & T IP Backbone: Early 2001

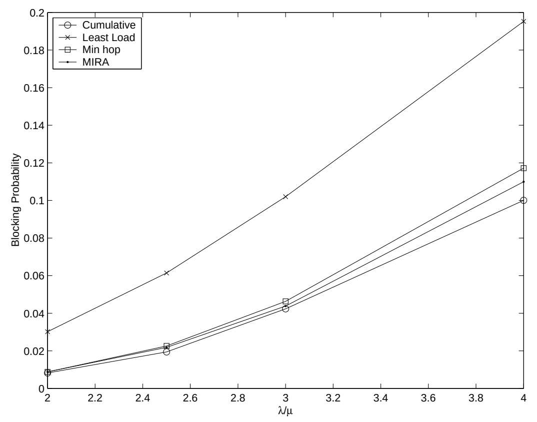

Fig. 4. MIRA Topology: : Total Blocking Probability for Least Loaded primary routing

Markovian, i.e. Poisson arrivals with rate λ and exponential holding times with rate μ. The arrival intensity is the same for all four source-destination pairs.

2) Ring of Rings Topology: The network shown in Figure 2 represents a typical metropolitan network. The larger center ring consists of links with capacity 48 and the smaller rings consist of links with capacity 12 to represent OC-48 and OC-12 links respectively. Again there are four source destination pairs; connection requests arrive as a Poisson process with intensity λ and connections hold for exponential times with rate μ. All source-destination pair statistics are independent and identical.

3) USA Topology: The network shown in Figure 3 is made in the image of the AT&T IP backbone [12], [13]. The fat links have capacity 24 and the thin ones capacity 6 to represent the capacity ratio between OC-192 and OC-48 links. The numbers just to the outside of the nodes represent a score that is roughly proportional to the number of optical links with capacity less than OC-48 incident on the node. The higher the node score the more likely the node is to be a source or a destination for a connection request. The assumption is that a node with a greater degree of sub OC-48 links is more likely to generate or terminate a connection request.

Again connection arrivals and holding times are Markovian. This time, however, all node pairs may act as source destination pairs with intensity proportional to the sum of their scores (Some sums are zero. No traffic is sent between these pairs.). So connections arrive for pair $(10, 12)$ at a rate four times as great as the rate they arrive for pair $(0, 3)$. Thus arrival intensities are proportional to their scores. The actual intensities were chosen so that the blocking rates observed were in a reasonable range – about two to three percent.

C. Results

1) MIRA Toplogy: Figures 4, 5, and 6 compare the blocking probabilities under different loads λ/μ using different algorithm combinations.

We can make a few observations from the above graphs. Using the least loaded algorithm for backup path routing uniformly performs the worst. For both minimum hop and MIRA algorithm for the normal paths, using the same algorithm for the backup path does the best, albeit only slightly better than using other algorithms for backup path.

Figure 7 compares the best backup path curves, for each of the normal path algorithms. We observe that the combination of MIRA primary routing and MIRA backup routing yields the lowest blocking probability in the MIRA topology.

2) Ring of Rings Topology: Table I shows the total blocking rates observed when the load is $\lambda/\mu = 3$. Each column lists the blocking rates observed when using the algorithm at the top of the column for primary paths and the algorithm indicated by the row for backup paths.

It is not too surprising that the blocking rates are almost all identical because there are very few routing options in the topology and different algorithms ended up choosing the same paths. What is interesting is that using MIRA for primary path routing yielded a blocking rate that was nearly 29% worse. We are evaluating why MIRA is picking different primary paths in the ring topology.

3) USA Topology: Table II shows the total blocking rates observed under different loads. Each of the three possible

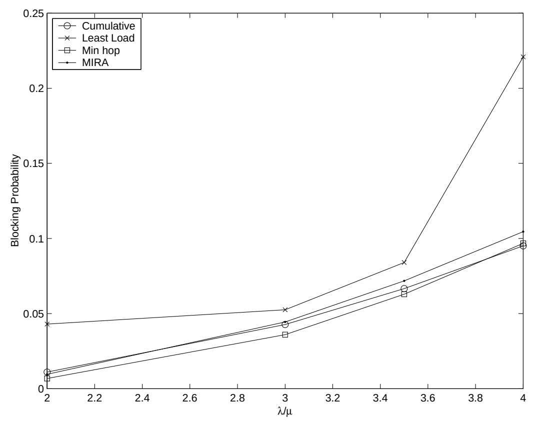

Fig. 5. MIRA Topology: Total Blocking Probability for Minimum Hop Count primary routing

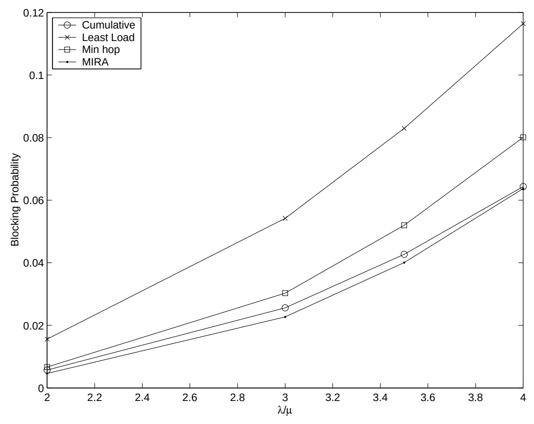

Fig. 6. MIRA Topology: Total Blocking Probability for MIRA primary routing

primary path algorithms is given a sub-table. The algorithms listed per row of sub-table indicate the corresponding backup path routing algorithm. The blocking rates were obtained for two different load scenarios. The first column holds blocking rates when the total load between all source destination pairs was 20; the second column holds values when this load was 24.

Interestingly using the least loaded algorithm for primary path routing and MIRA for the backup path yields the lowest blocking rate among all the combinations. In particular this blocking rate was approximately 10% less than when using MIRA for both the normal and backup paths and 50% less than when using least loaded for both paths. In general using

TABLE I

MAN TOTAL BLOCKING PROBABILITIES

	Least Loaded	Min Hop	MIRA
Cumulative Backup	0.06527	0.06527	0.08446
Least Loaded	0.09616	0.09616	0.11728
Min Hop	0.06527	0.06527	0.08446
MIRA	0.06527	0.06527	0.08446

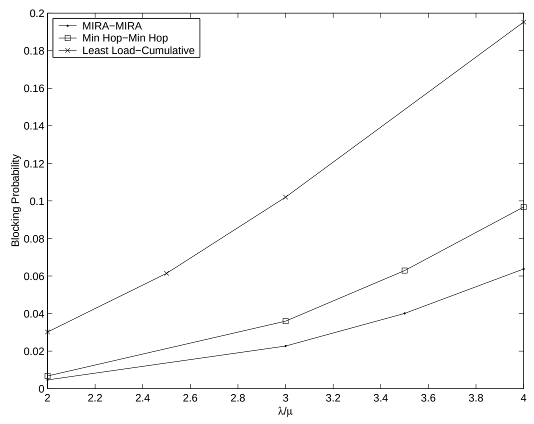

Fig. 7. MIRA Topology: Comparing the Best Combination for Different Normal Path Algorithms

TABLE II

USA TOTAL BLOCKING PROBABILITIES

Least Loaded Primary

	Total Load = 20	Total Load = 24
Cumulative Backup	0.01863	0.03238
Least Loaded	0.03357	0.05330
Minimum Hop	0.01841	0.03756
MIRA	0.01338	0.02526

Minimum Hop Primary

	Total Load = 20	Total Load = 24
Cumulative Backup	0.02363	0.03997
Least Loaded	0.03173	0.05035
Minimum Hop	0.02264	0.03756
MIRA	0.01791	0.03196

MIRA Primary

	Total Load = 20	Total Load = 24
Cumulative Backup	0.02082	0.03525
Least Loaded	0.02999	0.04842
Minimum Hop	0.02526	0.03271
MIRA	0.01490	0.02728

least loaded routing for the backup path yielded results that were significantly worse than all the other combinations.

Recall, however, that the arrival rates are different for different node pairs, but all critical links regardless of node pair are assigned the same weight. Thus, MIRA for primary path routing is likely not performing as well because it is choosing longer paths in order to avoid links important to even node pairs which generate little traffic.

VI. OCCASIONAL RECONFIGURATION OF BACKUP PATHS

Routes assigned to incoming path demands should be assigned so that the blocking probability is minimized in some sense. Admitting a connection across a given route incurs the opportunity cost of future connection requests being blocked. Intuitively we expect that a connection request should be routed in a way that balances the load across the network to best match the demand offered to the network.

A. Rebalancing backup paths for a single source

Each (s, d) pair has an alarm clock that rings periodically. To avoid synchronization problems clock rings are random-

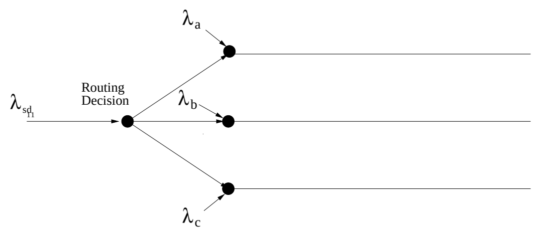

Fig. 8. Sample network: n = 3

ized. When a (s, d) pair's clock rings, the backup paths for that (s, d) pair are re-routed to better balance network load, if a better routing allocation exists.

We focus on the following example for illustrative purposes. Bandwidth is counted in units of connections and each link can support up to C connections. Assume (s_1, d_1) uses only one route for primary paths and uses n routes that are link disjoint with each other and with respect to the primary route, for backup paths. Protection paths are assigned per connection. So it is possible that one primary connection from s_1 to d_1 uses one backup path while another primary connection for (s_1, d_1) uses one of the other $n - 1$ backup paths. For all (s, d) pairs in the network, connection requests arrive as a Poisson process with rate λ_{sd}. Each connection holds for an exponentially distributed amount of time with holding rate μ independent of the arrival processes and other holding times. The alarm clock rings of (s_1, d_1) occur as a Poisson process with rate γ. The blocking probability is non-increasing with the number of available backup paths n. To make computations tractable, we assume that on every link in each backup path the aggregate cross-traffic connections arrive as independent Poisson arrivals with an intensity scaled by the complement to the aggregate blocking probability.

Note that the blocking probability will likely be dominated by certain bottleneck links. Thus, we assume for a given load profile on the network that all the blocking will occur on the most heavily loaded link in each of the backup paths. Figure 8 provides a picture of the control problem that (s_1, d_1) is faced with for $n = 3$.

We compute the blocking probability as a function of load for the scenario depicted in Figure 8. We bound blocking probabilities in the worst case where backup paths cannot share bandwidth reservations on any links.

We took $C = 10$ and $\lambda := \lambda_{s_1 d_1} = \lambda_a = \lambda_b = \lambda_c$. Arriving calls are routed to the least loaded link. Upon clock rings, switchable calls are routed to the least loaded link. Ties are broken by choosing at random among the links with the maximum spare capacity.

We computed blocking probabilities for λ/μ such that the blocking probabilities were on the order of 1 to 2 percent for $\gamma = 0, 10, \infty$ and $1/\mu = 1$. The total blocking probability as a function of load λ/μ is shown in Figure 9.

We notice that if up to 1% blocking probability is tolerated, it is possible to achieve a ten percent gain in network load (λ/μ ranges from 4 to 4.5) if γ is taken to be sufficiently

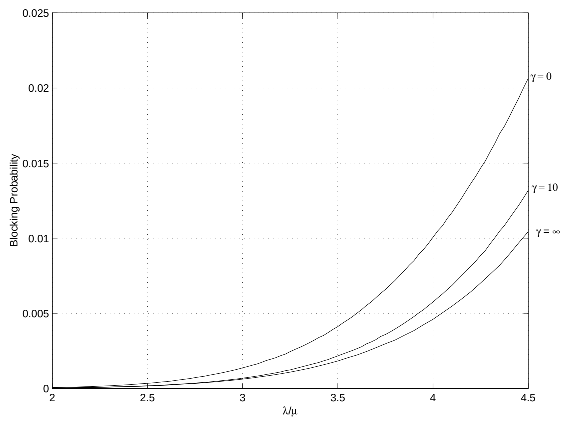

Fig. 9. Total Blocking Probability

large. Note also that for a load of $\lambda/\mu = 4.5$ a 1% blocking probability can be achieved if $\gamma = \infty$ whereas the blocking probability is 2% if no rebalancing is allowed.

B. *Rebalancing backup paths for two sources*

Consider a slightly more elaborate topology and demand load shown in Figure 10. We now have two links of interest that carry both primary and backup paths and two sources that generate with equal likelihood primary and backup requests. The backup paths for the two sources have primary paths not shown that are link disjoint and thus may share bandwidth. C is again taken to be 10 and we vary the arrival intensity λ at both sources. Calls are routed to the least loaded link. Uniform randomization breaks ties. Here we allow primary paths to also be rerouted. Thus gains with increasing load should only be more pronounced. The resulting blocking probabilities for the backup paths are shown in Figure 11.

There is marginal improvement by varying the rate at which the paths are rerouted. Although not shown gains in blocking probability for the primary paths are equally marginal. Having two sources that are routed into the network initially under a load balancing rule tends to keep the network overall balanced more often than not. So post-admission rerouting corrections have very little benefit. In the previous numerical example, much larger gains are achieved because poor greedy decisions occur more frequently.

All network topologies and traffic matrices will be mixtures of the two basic configurations and load profiles. The subsequent performance gains will be some "convex combination"

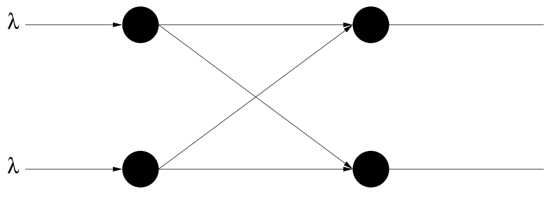

Fig. 10. Two source model

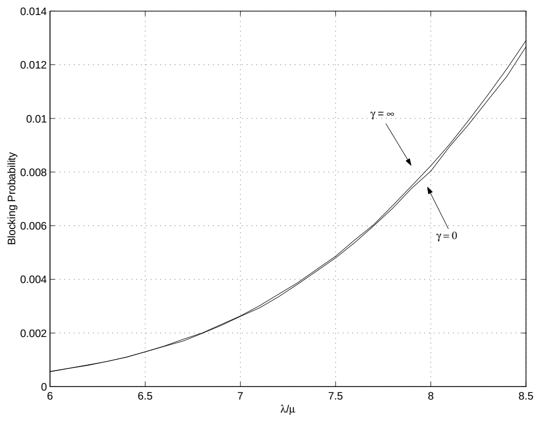

Fig. 11. Blocking probabilities: Backup Paths

of the performance gains of the first case and second case. Thus, gains can range from significant to trivial depending on the specific network context.

VII. Conclusion and Future Work

We considered the hypothesis that gains could be leveraged out of distinctly treating normal and protection paths, discussing reasons for using distinct algorithms for normal and backup paths and potential advantages of these schemes. We presented techniques to glue together different normal and protection algorithms in a distributed fashion.

We computed the blocking probabilities on different topologies using different combinations of routing algorithms for primary and backup path calculation. Our routing experiments indicate that depending on the topology and demand matrix the lowest blocking rates may be achieved by using different algorithms for primary and backup paths. The occasional reconfiguration of backup paths also reduced the blocking rates, but the gains were less significant as the number of source destination pairs increased.

We are continuing to run simulations for different scenarios to better understand why certain algorithm combinations do better than others under specific scenarios. We hope to present a more comprehensive set of results and analysis for the final version of the paper. It may be that the best combination is sensitive to the traffic matrix. Moreover, if the objective is not the blocking rate but some other metric at a data plane level (e.g. latency), it is not clear how different combinations would fair. Both of these issues are directions that warrant more exploration. Eventually we want to extend this work to a framework whereby a given network topology and predicted demand scenario will allow us to determine the best possible combination of normal and backup path routing.

Acknowledgement

We acknowledge the most valuable contributions of Dr. John Strand – his ideas, suggestions, and comments were extremely helpful. We also acknowledge the contributions of Tian Yu, for his help coding the simulator used in this work.

References

[1] D. Awduche *et al.*, "Requirements for Traffi c Engineering over MPLS," *RFC 2702*, September 1999.
[2] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala, "RSVP: A New Resource Reservation Protocol," *IEEE Communications Magazine*, 31(9):8-18, September 1993.
[3] R. Ramaswami, and K. N. Sivarajan, *Optical Networks: A Practical Perspective*, Morgan Kaufman Publishers, Inc., San Francisco, CA, 1998.
[4] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1992 Second Edition.
[5] J. Walrand and P. Varaiya, "*High-Performance Communication Networks,*" Morgan-Kaufman, 2000 Second Edition.
[6] E. Karasan and E. Ayanoglu, "Effects of Wavelength Routing and Selection Algorithms on Wavelength Conversion Gain in WDN Optical Networks," *IEEE/ACM Transactions on Networking*, Vol.6, No. 2, pp.186-197, April 1998.
[7] E.W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," *Numerische Mathematik*, 1:269-271, 1959.
[8] M. Kodialam and T. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffi c Engineering," IEEE INFOCOM 2000.
[9] R. Gupta and J. Walrand, "Routing Algorithms for Protected Guaranteed Bandwidth," submitted to DRCN 2003. Available: http://www.eecs.berkeley.edu/∼guptar/publications/.
[10] J. Moy. OSPF Version 2. *RFC 1583*, March 1994.
[11] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (bgp-4). *RFC 1771*, March 1995.
[12] UC Berkeley EECS 290Q Course website (Fall 2001): http://photonics.eecs.berkeley.edu/ee290q/jls/JLS%200911.pdf, Slide 37.
[13] AT&T CERFnet OC/12 Network Backbone website: http://www.cerf.net/about/Bbone-map.html