DYNAMIC ROUTING WITH PARTIAL INFORMATION IN MESH-RESTORABLE OPTICAL NETWORKS*

Murari Sridharan[1], R. Srinivasan[2] and Arun K. Somani[3]

[1] *Dependable Computing & Networking Laboratory*
murari@iastate.edu

[2] *Department of Electrical and Computer Engineering*
rsrini@iastate.edu

[3] *Iowa State University, Ames, IA 50011*
arun@iastate.edu

Abstract   Changing trends in backbone transport networks towards dynamic path provisioning and evolving optical technologies have motivated the study of dynamic routing algorithms in the context of Multi Protocol Label Switching (MPLS) based networks. Several methods have been proposed for joint optimization of working and spare capacity in survivable optical networks. These techniques are centralized and do not scale well as they rely on per-flow information. This motivates the need for developing a) distributed algorithms with complete information, b) source based algorithms with partial information which can be easily obtained from traffic engineering extensions to routing protocols.

In this paper, we develop dynamic algorithms for source based routing with partial information. The algorithms are classified based on the path selection approach used for the primary path. We compare the performance of various routing algorithms through simulation studies, based on metrics such as the call blocking probability, average path length of an accepted connections, capacity redundancy, and effective network utilization. Our studies show that dynamic routing algorithms perform better than static routing algorithms using pre-computed paths even when the path selection in static algorithms is based on optimizing a global network metric. The other interesting observation we make is that the performance improvement of dynamic routing algorithms using K pre-computed paths is significant even for small values of K.

Keywords: Optical networks, Dynamic routing, Protection, Restoration, Survivability

A. Bianco et al. (eds.), *Next Generation Optical Network Design and Modelling*
© Springer Science+Business Media New York 2003

1.   Introduction

Network survivability is a serious concern for today's businesses as they rely heavily on a reliable and continuously available high speed communications infrastructure. With millions of wavelength-miles laid out in typical global and nation wide networks, fiber optic cables are among the most prone to failures. GTS Trans European Network (formerly Hermes Europe Railtel) estimates an average of one cable cut every four days on their network [5]. Therefore, it is imperative to design networks that can quickly and efficiently recover from failures.

Survivable network architectures based on mesh-based (arbitrary) topologies offer better capacity efficiency and efficient re-routing around failed links. This is a result of the route diversity in arbitrary topologies, which is highly sensitive to the average nodal degree. Several survivability paradigms have been explored for surviving single link failures in mesh-based networks. They can be classified based on different criteria: (1) Centralized or Distributed based on their route computation and execution mechanisms; (2) Path-based or Link-based depending on the employed re-routing mechanism; (3) Pre-computed or Real-time based on the nature of the backup path discovery procedure; and (4) Dedicated or Shared based on the capacity sharing mechanisms across different connections. A detailed study of these different paradigms can be found in [4, 14, 17, 20, 23, 24].

As networks transition from SONET based ring architectures to mesh-based architectures, it is desirable to provide fast protection switching similar to the 50ms recovery time in SONET architecture. The recovery time in SONET is dictated by the fact that voice calls could be dropped if the recovery times were any longer. As a result, it is desirable to employ pre-compute paths to facilitate fast protection switching in mesh-based architectures followed by dynamic real-time spare capacity search mechanisms as part of the restoration process, which typically tends to be longer (order of minutes). It has been shown in [6] that path-based restoration with *stub release*[1] was found to be the most capacity efficient approach for mesh-based networks as compared to link-based re-routing mechanisms.

Capacity among the primary and restoration paths can be dedicated or shared. The dedicated technique uses 1:1 protection where each primary path has a corresponding restoration path. In the case of shared restoration, several primaries can share a same backup path as long as the primaries are node and link disjoint. This scheme is called the *backup multiplexing technique.*

Several methods have been proposed for joint working and spare capacity planning in survivable optical networks [4, 6, 7, 12, 17, 18, 20]. These methods consider a static traffic demand and perform routing and wavelength assignment (RWA) to optimize network cost assuming different cost models

and survivability paradigms. The application of these algorithms are restricted for the network provisioning and offline reconfiguration phases due to their high computational complexity. For network operation under dynamic traffic scenarios, the direct use of such methods for online reconfiguration remains limited to small networks with few tens of wavelengths.

Several heuristics and decomposition techniques [2, 4, 19] are being explored to significantly reduce the computational complexity of the RWA optimization problems. However, these techniques are centralized and require per-flow information, hence does not scale well with increasing network size. This motivates the need for developing: (a) distributed algorithms with complete information, and/or (b) source based algorithms with partial information which can be easily obtained from traffic engineering extensions to routing protocols.

1.1. Related work

Dynamic routing in wavelength-routed WDM networks has been studied extensively in the literature and several dynamic routing algorithms have been proposed [1, 8, 10, 11, 16, 24]. A methodology for dynamic routing of sub-wavelength traffic in WDM grooming networks is developed in [21]. The effect of grooming complexity and dispersity routing, where higher capacity requests are broken into multiple unit capacity requests, are analyzed. These studies are limited to establishing one connection per request and do not guarantee restoration paths.

Routing dependable connections under dynamic traffic was studied extensively in [13, 15, 22]. Dynamic routing of restorable bandwidth guaranteed tunnels is studied in [9]. They study dynamic routing using Integer Linear Programming (ILP) formulation under three information scenarios: none, complete, and partial information routing. These approaches do not scale well with the network size either due to the prohibitively large amount of information that needs to be maintained or high computational complexity involved in solving ILP formulations.

In this paper, we develop source-based dynamic routing algorithms to provide primary and backup paths for connection requests with 100% protection guarantee under single-link failures with partial information. Section2 introduces the framework for dynamic routing of primary and backup connections and the proposed routing algorithms. Section 3 discusses the experimental setup, performance metrics, and performance results of different routing algorithms. Section 4 presents our conclusions and discusses further improvements based on our observations.

2.   Dynamic Routing with Partial Information

In this section, we present a framework for dynamic routing of primary and backup connections and propose various routing algorithms. The framework presented in this section is an extension of the framework presented in [21].

Every link in the network is denoted by a *link-state vector*. The vector consists of a set of properties associated with a link, eg. available bandwidth, primary capacity, backup capacity, hop-length, fiber length etc. Each entity in the vector is referred to as a *metric*. Every path from a source to destination has a *path-vector* that is obtained by combining the link-state vectors of the links in the path. Note that the link vector is a special case of a path vector when the path has only one link.

In WDM networks, the metrics can be classified either as path-specific or wavelength-specific. Path-specific metrics are those metrics that depend only on the route from a source to destination and are independent of the wavelength used. One example of path metric is the hop-length. The usage of a wavelength as a primary or backup on a link is an example of a wavelength-specific metric. A metric is said to be *concave* if its value in a path vector is the minimum among the corresponding metrics on the individual links of the path, *additive* if its value in a path vector is the sum of the corresponding metrics of the individual links in the path, *multiplicative* if its value in a path vector is the product of the corresponding metrics on individual links. Various dynamic path selection algorithms can be developed based on the above specified metrics.

Every node in the network is assumed to maintain the global state information through a link-state protocol. The information available for each link (i, j) is the status of each wavelength, set to 1 if used as a primary, 0 if used as backup, and -1 if available. Note that, although we know if a wavelength is used as a primary or backup, we still do not know which node pair's flow is using the wavelength as a primary, or which flows are sharing the wavelength for backup.

Dijkstra's shortest path algorithm is extended to the above link-state vector, referred to as extended Dijkstra's shortest path (EDSP) algorithm, and is employed at every node in the network. The EDSP algorithm uses the link-state vector as defined above instead of a single metric that is traditionally used. The EDSP algorithm has two important operations: (1) combining two path vectors and (2) selecting the best path vector. A path vector is a combination of link state vectors. The various combine functions are described later in the section. We will illustrate the path vector combine function using an example shown in Figure 1.

As defined earlier, associated with each wavelength in a link-state vector lsvij is a status variable which is set to 1 if used as a primary, 0 if used as backup, and -1 if available. By examining each lsv, we can determine the total primary wavelengths by adding all the wavelengths whose status value is 1, the
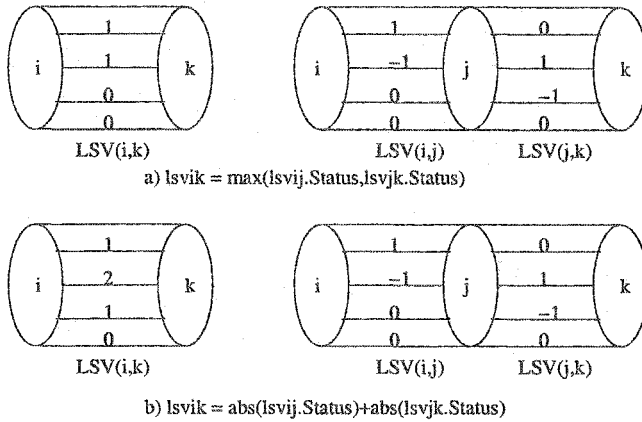
Figure 1. Combine function for two path vectors.

total backup wavelengths by adding wavelengths whose status value is 0, and total available wavelengths by adding wavelengths whose status value is -1. Although a status variable is indexed and maintained for each wavelength in a link, in the discussions to follow, we use a generic status variable to represent all wavelengths. If the combine function is the maximum of the status variables, i.e., $lsv_{ik}.Status = max(lsv_{ij}.Status, lsv_{jk}.Status)$. By examining the combined $lsv_{ik}.Status$ as shown in Figure 1a, the following information can be obtained. Since we are interested in wavelength continuous paths, any wavelength λ whose status in the combined lsv is 1 cannot be used as there is atleast one link in the corresponding wavelength continuous path where λ is being used as a primary wavelength. Any λ whose status in the combined lsv is -1 means the wavelength continuous path corresponding to λ is available on all links on the path. Similarly, any λ whose status in the combined lsv is 0 means the wavelength continuous path has λ either being used as a backup or is available. Therefore, to check if a wavelength continuous path can be established or not, it is sufficient to check the status of the wavelengths in the combined path vector.

The second operation of selecting the best path vector from a given set of path vectors is defined by a specific path selection policy. For example, the traditional shortest path algorithm selects a path with minimum hop length. To understand how different combine functions have different effects on the path selection, and hence the routing algorithm, refer to Figure 1. Let us assume that we would like to choose a path which has maximum number of backups (already reserved) available. If we use the combine function in Figure 1a, we will get paths where, in the worst case only one link along the path has the corresponding wavelength being used as backup and on all other links the

wavelength is available. In such a scenario, once the path is selected we need to reserve capacity on all links where the wavelength was available. On the other hand, if we assume that the combine function as shown in Figure 1b is used, i.e., $lsv_{ik}.Status = abs(lsv_{ij}.Status) + abs(lsv_{jk}.Status)$. Now in this case, as can be seen in the figure, any λ whose status in the combined lsv is 0, means that λ is being used as backup on all links on the path. In such a scenario, once the path is selected, we do not need to make any excess reservations. It is to be noted here that, we merely use the above illustration to explain the process of combine and selection, actual decision on whether we can multiplex a given connection's backup on to an already reserved backup wavelength will depend on whether such an assignment will violate the 100% protection guarantee. We will explain how the algorithms ensure 100% protection guarantee later in the section.

2.1. Routing algorithms

We first discuss the basis upon which the routing algorithms are developed. The key is to use the aggregate information in the partial information scenario to obtain significant gains in network performance. Let us assume that the primary path for an arriving request has been selected. The maximum number of primary connections on a link along the path is referred to as the *conflict* created upon routing the current primary connection, inclusive of the primary. For any potential backup path which is disjoint from the primary, it is required that the path has atleast C wavelengths that are either available or used as backup. Note that, as explained earlier in the section, this depends on the kind of combine function used. Now suppose if we cannot find C backup wavelengths, then assign one wavelength continuous path without any sharing. It is a fairly simple exercise to see that in both the above cases, 100% protection guarantee is maintained.

We will illustrate this with an example. Figure 2 shows the snapshot of a part of the network.

Assume that there are two wavelengths λ_1, λ_2 on each link. There are two primary paths P1 and P2 which use, as part of its route, links 2-3 and 2-1 respectively. Assume that a primary path for the new request between (3,1) has been selected to be 3-2-1 (Pnew) on λ_2. Note that the primary wavelengths cannot be shared. The maximum conflict created by Pnew is $C = 2$. Now the claim is that as long as there are C wavelength-continuous backups assigned on the selected backup path for this new request, 100% protection guarantee can be ensured. The other case is to assign one wavelength continuous path on 3-4-1 without any sharing. It is trivial to see that in the latter case, since one dedicated wavelength continuous path has been assigned for the new request, 100% guarantee is ensured.
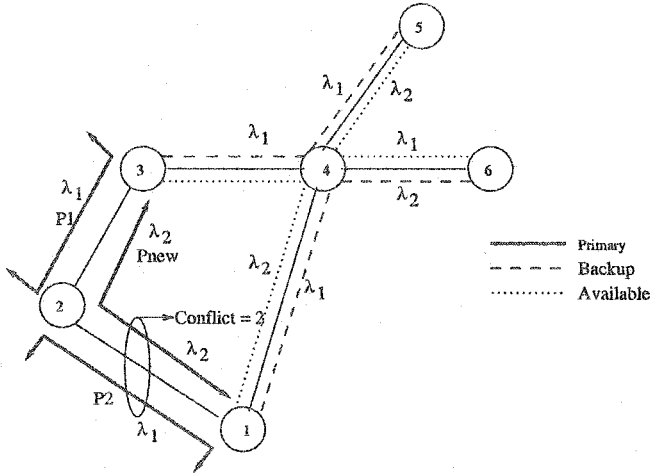
Figure 2.   Snapshot of a part of the network to illustrate protection guarantee.

Let us examine the first case of the claim that there are C wavelength-continuous backup paths assigned. Suppose, if it is not the case then it would result in the violation of 100% guarantee. To understand this better consider the example shown in Figure 2. Suppose that we were to assign one wavelength continuous path on 3-4-1 using λ_1. Here on both 3-4 and 4-1 λ_1 has been already reserved as backup for some earlier request(s). Assume that the backup path for P2 had been assigned λ_1 on link 3-4. Suppose of link 1-2 fails, both Pnew and P2 contend for the same wavelength λ_1, thereby violating 100% protection guarantee. Note here that instead of λ_1, if we were to assign the backup path for the new request on 3-4-1 using λ_2 which is available on both links, then this becomes a dedicated backup for Pnew and as shown earlier is a trivial case.

It is clear that in some cases, due to lack of information, more backup wavelengths may be reserved than required. In the above example, if P2 were not using any of links 3-4 or 4-1 for backup, we could have assigned the backup path for the new request on 3-4-1, sharing the already reserved wavelength λ_1. The inherent assumption here is that the algorithm had ensured protection guarantee for all earlier requests before the arrival of the new request. This is the best decision that any routing algorithm can make to provide 100% protection guarantee given the partial information scenario. So we need to exploit the excess reservations that may be made in some cases to our advantage by reusing such links where backup wavelength chunks are already available.

It is to be noted that, we need to ensure C wavelength continuous paths, instead of merely checking for C wavelength capacity to be either available or

reserved as backup on all links on the backup path. This is because when the backup has to be actually assigned for this request, we may not find a common wavelength that is available on all paths, although the required capacity is present. Although we provide C possible backup wavelengths for some arriving requests, we need to switch the connection onto one specific wavelength when a failure occurs. There are some issues as to the order in which the affected requests pick the backup wavelength to switch to. A wrong assignment here may result in some requests being blocked. This can be avoided by some extra processing to ensure that the switch to the backup path is done without blocking any requests.

The optimality metric for our dynamic algorithm is stated as follows: route the primary on the shortest path, and try to route backup on links where backup wavelengths are already reserved by earlier requests, at the same time ensuring 100% protection guarantee. We develop routing algorithms based on our optimality definition and all the observations discussed earlier in the paper. All the algorithms are designed to provide 100% protection guarantee. Each of the algorithms have a metric for path selection and a wavelength assignment algorithm. The algorithms are classified based on the path selection approach used for the primary path, as listed below.

Shortest Cycle (SC): This approach uses fixed alternate path routing. For path selection, the shortest vertex disjoint cycle which minimizes the sum of path lengths is chosen. The two vertex disjoint paths are pre-computed using the algorithm in [3]. The primary is routed on the shortest path and the backup on the alternate path. Wavelength sharing is allowed on the backup path.

Shortest Path Unconstrained Primary (SPUP): In this approach, the primary is routed without any constraints on the shortest path. The metric used for routing the backup path is to maximize the minimum backup capacity (MAXMINBACKUP) available. This metric attempts to find the a path for which minimum backup capacity available on the path is maximum. This can be achieved using the EDSP framework, by using a minimum metric in the link-state vector combine function for backup capacity, and choosing the maximum value when comparing path vectors to update the Dijkstra cost. The backup is constrained by the fact that the primary has been routed on the shortest path and C conflicts have been created. Now suppose it is not possible to find a backup path that can satisfy the conflict, the algorithm tries to accommodate the request using the SC approach.

Least Conflict Path Unconstrained Primary (LCPUP): The primary is routed using the least conflict path, i.e., minimize the maximum conflict created on the primary path. This is achieved in the EDSP framework, by using a maximum metric in the link-state vector combine function for primary capacity, and choosing the minimum value when comparing path vectors to update the

Dijkstra cost. The backup path as in the SPUP approach is chosen using the MAXMINBACKUP metric.

K Shortest Paths (KSP): In this approach, K shortest paths are pre-computed for each node pair. The routing algorithm employs SPUP approach for each of the K paths, and chooses the one that minimizes some chosen network metric. Unlike the SPUP approach, if for any path ik on which the primary is assumed to be routed, if it cannot find C conflict backups, it merely skips and proceeds with the next shortest path. Some good network metrics include minimizing the total wavelength miles consumed, or minimizing the maximum link utilization. Since this approach tries to minimize a global network metric, and has a choice of paths to choose from, it is expected to work the best. We use this algorithm mainly to draw comparisons to other approaches.

The key aspect in all these algorithms is to exploit the excess reservations that may be made in some cases to our advantage by reusing the already reserved backup capacity. We use FirstFit wavelength assignment policy. Primary paths use FirstFitDescending, by arranging wavelengths in descending order and picking the first wavelength that is available. Backup paths use FirstFitAscending, by arranging wavelengths in ascending order and picking the first wavelength that is available.

Each approach works on different aspects of the optimality definition. For example SC would try to assign requests on shortest cycle, but may not be able to perform efficient sharing of backup wavelengths. A metric like SPUP may gain initially by providing primary on shortest path and share a backup on some longer path. However, as requests which share wavelengths on the longer paths leave, it might end up using extra resources. Thus it is clear that each algorithm tries to arrive at a tradeoff on optimal routing. In the next section, we will evaluate and compare the performance of these different approaches.

3. Performance Evaluation

The performance of the algorithms described in the previous section are evaluated on the 14 node 22 link NSFNet network topology, shown in Figure 3. When a request arrives at a node, primary and backup paths are chosen using the above mentioned routing algorithms. FirstFit wavelength allocation policy is used for both primary and backup paths, with primary choosing from wavelengths in descending order, and backup in ascending order.

3.1. Experimental setup

The experimental setup for the simulation is based on the following assumptions: (1)The arrival of requests at a node follows a Poisson process with rate λ and are equally likely to be destined to any other node; (2) The holding time of the requests follow an exponential distribution with unit mean; (3) The capacity
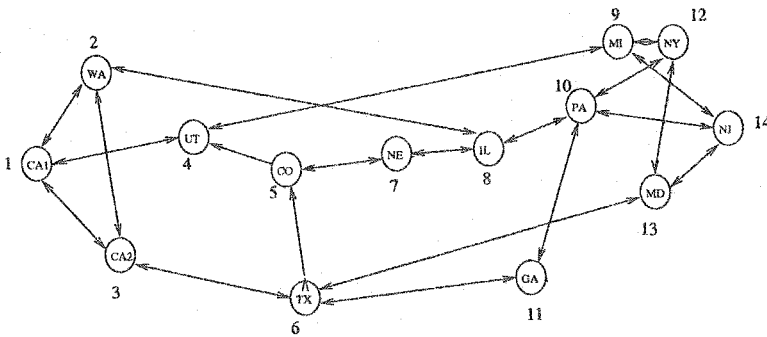
Figure 3.   14 node 22 link NSFNet topology.

requirement of a request is restricted to a full wavelength; and (4) Every link has 32 wavelengths.

The requests are generated independently at a rate of $N\lambda$, where N denotes the number of nodes in the network. The requests are equally likely to have any of the N nodes as its source. The generated requests are fed to the different networks running in parallel and their performances are measured. A total of 5×10^5 requests were generated with performance metrics being measured in batches of 10^5 requests. The average of the performance metrics over observed five set of values are reported in the results.

3.2.   Performance metrics

The performance metrics measured are the request blocking probability, average path length of an accepted connection(Z), redundancy, and network utilization (η). Since every request is assigned a primary and a backup path, and since the network revenue is gained as a result of accepting more primaries, and the backup capacity is idle until a failure occurs, all the measurements and performance comparisons are made specific to the primary paths.

The blocking probability is computed as the ratio of the number of blocked requests to the total number of requests generated. Z is computed as the average of the length of the primary paths assigned to the accepted requests by a specific routing algorithm. Redundancy is defined as the ratio of total spare to working capacity in the network.

The *network utilization* is computed by assigning an effective network capacity requirement for a request. A request r for capacity b from source s to destination d has an effective capacity requirement of $b \times h_s$, where h_s is the shortest path length from the source to the destination. This effective capacity requirement of a request is the minimum capacity that is required in the network to support the request, irrespective of the routing algorithm. If a routing

algorithm selects a path of length h for the connection, $b(h - h_s)$ denotes the additional capacity used by the network to support the connection.

The effective network capacity utilized at an instant of time, denoted by U, is defined as the sum of the effective network primary capacity requirement of all the connections that are active at that instant. The value of U at any instant of time is bounded by $L \times C$, where L is the total number of links in the network and C is the capacity on each link. The network utilization is then computed as the ratio of the effective used capacity to the maximum capacity of the network as $\eta = \frac{U}{L \times C}$.

3.3. Results and discussion

The blocking performance of various routing algorithms versus the total arrival rate of requests for the NSFNet network with 32 wavelengths is shown in Figure 4. It can be observed that LCPUP performs the best. The SC algorithm performs the worst. Although the SC algorithm allocates the primary on the shortest path, it does not have much choice on the selection of the backup path. As a result it is constrained, hence performs poorly. The SPUP algorithm which fixes the primary to the shortest path but has a choice of selecting the best backup path, still performs very close to SC. The gain is small, with less than 2% improvement in blocking performance. This is because both these algorithms restrict the primary connection to the shortest path. As a result both these approaches create similar conflict values, and since SPUP has a choice of selecting the best backup path available, it performs slightly better than SC. LCPUP which routes the primary on the least conflict path and selects the backup based on the MAXMINBACKUP metric, performs the best. However, as the load increases longer calls are blocked and primary paths are restricted to being close to shortest path lengths, the performance of all the algorithms converge.

Figure 5 shows the average primary path length of the accepted connections. The average primary path length of LCPUP algorithm is greater than those of SC and SPUP. The average primary path lengths of SC and SPUP are almost the same. The reason they are not exactly the same, although both route primary on the shortest path. This is because there is a slight difference in the number of requests admitted by both routing algorithms and the average is computed for only accepted connections. This trend confirms the fact seen in Figure 4. Since LCPUP is able to route primary path through least conflict longer paths, it is able to improve the overall blocking. Average primary path length of LCPUP is almost 17% greater than those of SC and SPUP.

These trends are similar under the effective network utilization performance metric as can be seen in Figure 6. Note that although there is significant improvement in blocking for LCPUP, the improvement in effective network utilization
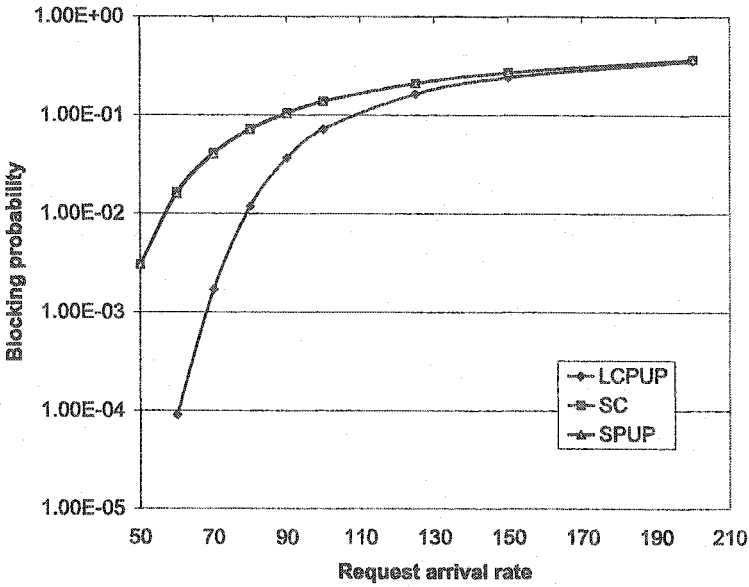
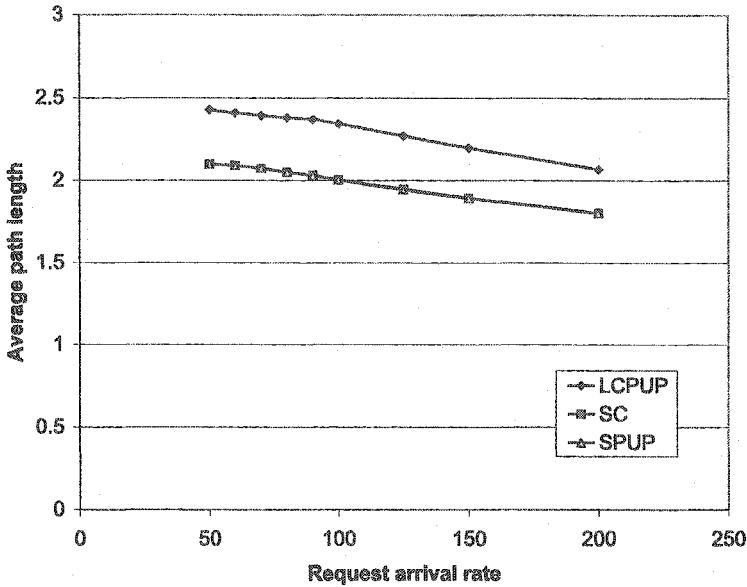*Figure 4.*   Blocking performance of various routing algorithms.

Figure 5.   Average primary path length of accepted requests under different routing algorithms.

is only 8%. As explained earlier, the effective capacity requirement for any request is computed based on the minimum capacity that is required to support
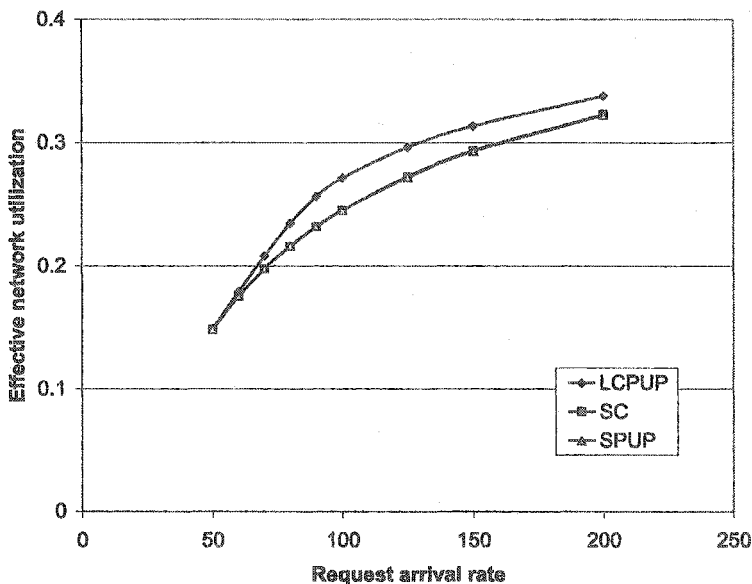
Figure 6. Network utilization under different routing algorithms.

the request (based on shortest path length), irrespective of the routing algorithm. As a result LCPUP, which routes on a longer paths, consumes additional capacity than required and hence does not gain much in this metric.

The redundancy, which is a ratio of the total spare to total working capacity in the network, is plotted for various routing algorithms in Figure 7. The more the redundancy the worse the blocking performance, since this capacity can otherwise be used to accommodate more requests. SC is almost similar to dedicated protection. This is because although we allow backup wavelength sharing for SC, the paths are still restricted. As a result it performs the worst in redundancy and hence worst in blocking performance as seen earlier. SPUP which is an improvement over SC still performs poorly for the same reasons stated above. LCPUP has the best redundancy over all. One interesting aspect to note here is that even for LCPUP, redundancy is still greater than 100%. Typical path protection algorithms perform very efficiently with almost 50% redundancy. But they work with complete per-flow information. The routing algorithms developed in this paper work under the partial information scenario, and so the performance is expected to be worse when compared to algorithms which have complete information.

To confirm the trends observed and to demonstrate the effectiveness of the LCPUP algorithm, we compare these algorithms with KSP routing algorithm. The results comparing SPUP and LCPUP with KSP routing algorithm for different values of K are shown in Figure 8. In the KSP approach, we minimize
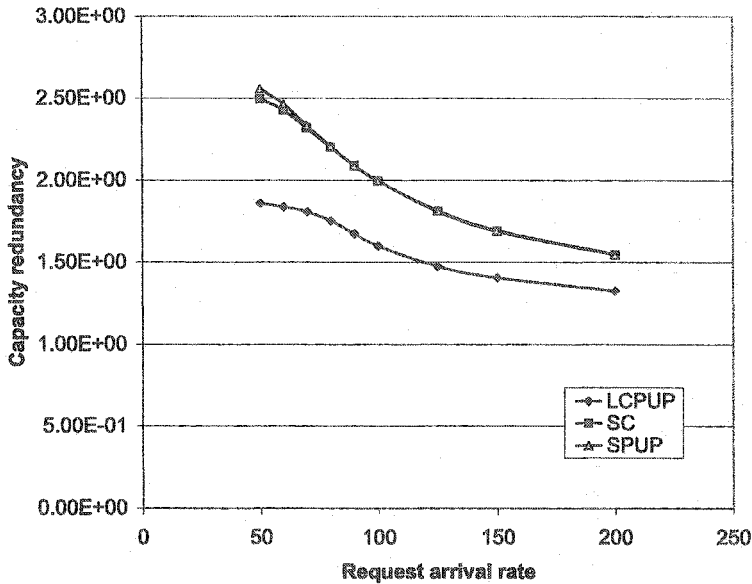
Figure 7.   Redundancy of various routing algorithms.

the total wavelength miles. Surprisingly LCPUP still performs the best in terms of blocking performance, although under heavy loads, KSP with K=10 outperforms LCPUP slightly. This is because, under heavy loads longer paths are typically blocked, and given that path selection has to be made using shorter paths, KSP, which explores over all possible K paths and chooses based on minimum wavelength usage, performs better. Thus the dynamic algorithm LCPUP performs better as compared to static algorithms using pre-computed paths, even when the path selection is based on current network state.

Note that SPUP is provided with more paths with increasing values of K. This results in better performance. This effect is seen for values of K as small as 3 or 5. Although the set of 3 or 5 pre-computed paths might look restrictive, it is interesting to note that even for such small values the performance improvements are fairly significant.

4.   Conclusions

In this paper, we develop source based algorithms with partial information which can be easily obtained from traffic engineering extensions to routing protocols. The algorithms differ on the path selection approach used for the primary path. We compare the performance of various routing algorithms through simulation studies based on metrics such as the call blocking probability, aver-
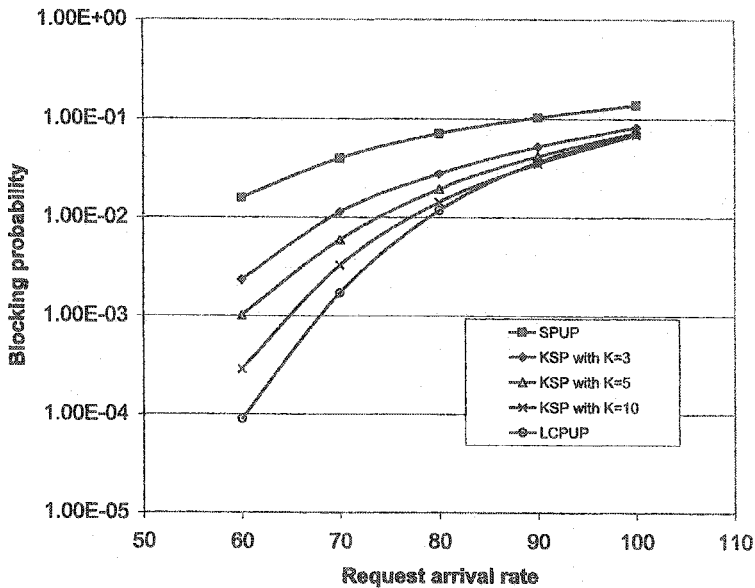
Figure 8. Blocking performance improvements for KSP routing algorithm.

age path length of an accepted connections, capacity redundancy, and effective network utilization.

Our studies show that dynamic routing algorithms perform better than static routing algorithms using pre-computed paths, even when the path selection in static algorithms is based on current network state. The other interesting observation we make is that the performance improvement of dynamic routing algorithms using K pre-computed paths is significant even for small values of K.

Currently, we are working on an integer programming based formulation for capacity optimization in the partial information scenario to provide theoretical basis for the simulation studies carried out in this paper.

Notes

1. Stub release refers to the release of capacity along the surviving upstream and downstream portions of a failed primary path, and making those links available for the restoration process.

References

[1] Alanyali, M. and E. Ayanoglu: 1998, 'Provisioning Algorithms for WDM Optical Networks'. *IEEE INFOCOM* 2, 910–918.

[2] Banerjee, D. and B. Mukherjee: 1996, 'A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks'. *IEEE Journal of Selected Areas in Communications* 14(5), 903–908.

[3] Bhandari, R.: 1999, *Survivable Networks: Algorithms for Diverse Routing*. Kluwer Academic Publishers.

[4] Doshi, B., S. Dravida, P. Harshavardhana, O. Hauser, and Y. Wang: 1999, 'Optical Network Design and Restoration'. *Bell Labs Technical Journal* pp. 58–83.

[5] Fonseca, P.: 1998, 'Pan-European Multi-Wavelength Transport Networks: Network Design, Architecture, Survivability and SDH Networking'. *Proceedings of the 1st International Workshop on Reliable Communication Networks*.

[6] Iraschko, R., M. MacGregor, and W. Grover: 1998, 'Optimal Capacity Placement for Path Restoration in STM or ATM Mesh-Survivable Networks'. *IEEE/ACM Transactions on Networking* 6(3), 325–336.

[7] Iraschko, R. R. and W. D. Grover: 2000, 'A highly efficient path-restoration protocol for management of optical network transport integrity'. *IEEE Journal of Selected Areas in Communications* 18(5), 779–794.

[8] Jue, J. and G. Xiao: 2000, 'An adaptive routing algorithm for wavelength-routed optical networks with a distributed control scheme'. In: *Proceedings of the Nineth International Conference on Computer Communications and Networks*. pp. 192–197.

[9] Kodialam, M. and T. Lakshman: 2000, 'Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration'. In: *Proceedings of IEEE INFOCOM'00*. pp. 902–911.

[10] Li, L. and A. K. Somani: 1999, 'Dynamic Wavelength Routing Using Congestion and Neighbourhood Information'. *IEEE/ACM Transactions on Networking* 7(5), 779–786.

[11] Lowe, E. D. and D. K. Hunter: 1997, 'Performance of dynamic path optical networks'. In: *IEE-Proceedings of Optoelectronics*. pp. 235–239.

[12] Miyao, Y. and H. Saito: 1998, 'Optimal Design and Evaluation of Survivable WDM Transport Networks'. *IEEE Journal of Selected Areas in Communications* 16(7), 1190–1198.

[13] Mohan, G., C. Murthy, and A. K. Somani: 2001, 'Efficient Algorithms for Routing Dependable Connections in WDM Optical Networks'. *IEEE/ACM Transactions on Networking* 9(5), 553–566.

[14] Mohan, G. and A. K. Somani: 2000, 'Routing Dependable Connections With Specified Failure Restoration Guarantees in WDM Networks'. *IEEE INFOCOM* pp. 1761–1770.

[15] Ramamurthy, R., Z. Bogdanowicz, S. Samieian, D. Saha, B. Rajagopalan, S. Sengupta, and S. Chaudhuri: 2001, 'Capacity Performance of Dynamic Provisioning in Optical Networks'. *Journal of Lightwave Technology* 19(1), 40–48.

[16] Ramamurthy, S. and B. Mukherjee: 1998, 'Fixed alternate routing and wavelength conversion in wavelength-routed optical networks'. In: *Proceedings of the Global Telecommunications Conference, GLOBECOM'98*. pp. 2295–2303.

[17] Ramamurthy, S. and B. Mukherjee: 1999, 'Survivable WDM Mesh Networks, Part I:Protection'. *IEEE INFOCOM* 2, 744–751.

[18] Ramaswamy, R. and K. Sivarajan: 1995, 'Routing and Wavelength Assignment in All-Optical Networks'. *IEEE/ACM Transactions on Networking* 3(5), 489–500.

[19] Sridharan, M., M. Salapaka, and A. K. Somani: Expected publication date : 1st Quarter 2002, 'A Practical Approach to Operating Survivable WDM Networks'. Accepted for

publication in Spl. Issue on WDM Based Network Architectures, Journal of Selected Areas in Communication.

[20] Sridharan, M., A. K. Somani, and M. Salapaka: 2001, 'Approaches for Capacity and Revenue Optimization in Survivable WDM Networks'. *Journal of High Speed Networks* 10(2), 109 – 125.

[21] Srinivasan, R. and A. K. Somani : 2001, 'Dynamic Routing in WDM Grooming Networks'. *Technical Report (DCNL-ON-2001-001), Dependable Computing & Networking Laboratory, Department of Electrical and Computer Engineering, Iowa State University.*

[22] Thiagarajan, S. and A. K. Somani: 2001, 'Traffic Grooming for Survivable WDM Mesh Networks'. *Opticomm: Optical Networking and Communications.*

[23] Wu, T.: 1992, *Fiber Network Service Survivability.* Norwood, MA: Artech House.

[24] Zang, H. and B. Mukherjee: 2001, 'Connection Management for Survivable Wavelength-routed WDM Mesh Networks'. *Optical Networks Magazine* 2(4), 17–28.