Shared Protection Routing Algorithm for Optical Network

Shengli Yuan (e-mail: syuan@utdallas.edu)
Jason P. Jue (e-mail: jjue@utdallas.edu)
Department of Computer Science, University of Texas at Dallas
Richardson, TX 75083

ABSTRACT

In a connection-oriented network such as an all-optical transport network, shared protection provides the same level of protection against single path failures as dedicated protection, with potentially higher network utilization. This paper lists the requirements of path protection and proposes a heuristic routing algorithm for shared protection provisioning. Simulations were conducted to verify the algorithm and to compare network utilization of shared protection to that of dedicated protection.

1. INTRODUCTION

A connection-oriented network provides end-to-end paths such as Optical Light Paths (OLPs) in a DWDM network, and MPLS Label Switched Paths (LSPs) in a MPLS domain. To prevent traffic loss, a path may be protected by another path, or so-called protection path or backup path. The protection path has the same source and destination as the primary, or working path. When the primary path fails, the protection path is activated to continue carrying traffic. Statistically, the failure probabilities of the paths ought to be independent; thus if the failure probability of one path is $p_f < 1$, the probability of traffic loss is reduced to $p_f^2 < p_f$.

Based on whether the sharing of network resources is allowed, a protection scheme can be categorized as dedicated protection or shared protection. In dedicated protection, different protection paths do not share common resource, which may be a fiber line, a SONET channel, a WDM wavelength or a switch. Examples of dedicated protection are SONET 1+1 protection and SONET 1:1 protection [1]. In SONET 1+1 protection, traffic is transmitted

An earlier version of this paper was presented at the OptiComm 2001 Conference, 21 –23 Auguest 2001, Denver, USA simultaneously on two separate fibers from the source to the destination. In SONET 1:1 protection, traffic is transmitted over only one fiber at a time. For dedicated protection, the failure and activation of one protection path doesn't affect any other protection path. The provisioning of this type of protection is relatively simple, and its behavior is deterministic.

On the other hand, in shared protection, multiple protection paths may go through common resources as illustrated in Figure 1. In this example, the solid lines are primary paths and the dash lines are protection paths. The protection paths of primary path *ab* and *cd* share common nodes *e* and *f* as well as the link between theses nodes. SONET 1:N protection is a special case of shared protection, since all primary paths have the same source and destination. In shared protection, when one protection path is activated, other protection paths that share common resources with it may have to be rerouted. For example, when primary path *ab* in Figure 1 fails, protection path *aefb* is activated, so the protection path *cefd* for primary path *cd* has to be rerouted. On the other hand, when a common resource fails, all protection paths that share that resource need to be rerouted. In the same example, if link *ef* fails, new protection paths of both primary paths need to be reestablished. Therefore shared protection is more complex to provision and maintain.

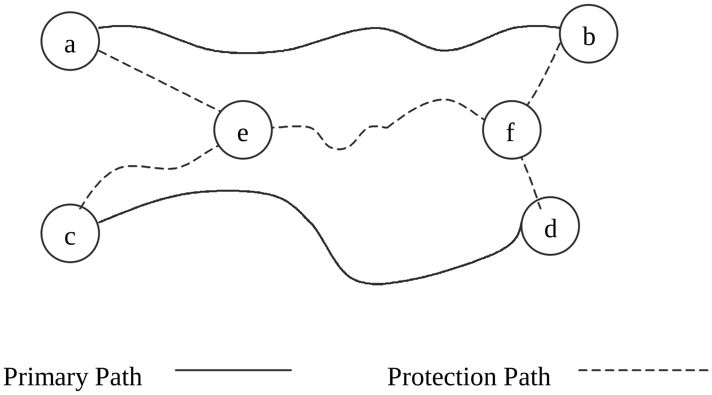


Figure 1. Example of Shared Protection

However, shared protection does offer one advantage over dedicated protection, i.e., it may offer higher network utilization. Assume that every path needs protection. In the dedicated case, the best network utilization would be 50%. On the other hand, for shared protection, since multiple paths share common resources, the total number of resources required for all the protection paths can potentially be much lower. If the failure probabilities of primary paths are statistically independent, we wouldn't expect multiple paths to fail simultaneously, in which case shared protection provides the same protection as dedicated protection. This concept can be illustrated in the following example.

Assume that there are 10 primary paths in a network, each with an independent failure probability of 0.01. At any moment the probability of path failure is 1- $(1\text{-}0.01)^{10}$ = 0.9562, of which single path failure probability is $10{*}0.01{*}(1\text{-}0.01)^{9}$ = 0.9135. Thus a single path failure counts for 95.534% of total path failures, for which shared protection performs as well as dedicated protection. For the remaining 4.466% failures, i.e., multiple path failures, the performance of shared protection would depend on how effectively the other protection paths are rerouted when one protection path is activated.

Shared protection provides decent protection with potentially much lower network resources; thus, the network can achieve higher utilization. Shared protection and dedicated protection schemes complement each other to offer more flexible solutions. Only the paths with the most strict protection requirement need to be dedicatedly protected. The remaining paths can be protected under shared protection and free up network resources, to either support more paths, or to protect paths that had no protection before.

The benefits of shared protection have attracted some research interests, especially for the emerging all optical DWDM network. [2] proposes a control protocol with various capabilities including shared protection routing for WDM mesh networks. [3] actually implements optical layer shared protection in a 5-node optical network. Additionally, research in [4] describes the properties and benefits of dedicated and shared protections rings, and how both types can contribute to a cost-efficient design for a stack of WDM rings.

In this paper, we investigate the routing problem in shared protection. The goal is to successfully route primary and protection paths with the minimum usage of network resources. This problem can be considered under two different traffic assumptions. Static routing applies to the case in which the set of connections is known in advance. In the dynamic case, connection requests arrive to the network dynamically. The static case is NP-complete. Studies in [5] and [6] fall into the first category. [5] develops an integer linear program (ILP) formulation which is quite effective for small network. For larger network, a heuristic algorithm proposed by [6] may be more suitable. Under dynamic routing, [7] suggests a “bucket-based” link state representation combined with shortest path algorithms. But this approach is most effective only in link-based protection, i.e., each link on the working path is protected by an alternative path connecting the two end nodes of the link.

This paper addresses the dynamic routing problem for shared protection. We first establish a generic framework to capture the information on risks and potential failures of a network. We then propose a heuristic routing algorithm which utilizes the information to satisfy dynamic connection request with end-to-end path protection. This approach is applicable to all connection-oriented networks including the all optical DWDM network. The next section describes the heuristic routing algorithm in details. Section 3 presents simulation results. Section 4 discusses other issues related to shared protection, and Section 5 concludes the paper.

2. ROUTING ALGORITHM

When implementing routing algorithm for shared protection, a number of requirements must be satisfied. We list these requirements below.

Requirement 1. A routing algorithm for path protection is subjected to the risk disjointing constraint, i.e., a primary path and its protection path must not undertake the same risk(s); otherwise the same failure may cause both paths to fail. This requirement applies to both dedicated and shared protection.

In order to implement this requirement mathematically, we assign a unique number, a **Risk ID** for each risk. If a network resource is subjected to multiple risks, the collection of the risk IDs describes all the risks for that resource, and the collection of the risk IDs of all of a path’s resources describes the path’s total risks. As we stated earlier, a resource can be a fiber link, a wavelength or a nodal equipment such as a switch. This collection of risk IDs is called the **Risk Vector**. For instance, in a DWDM network, a lightpath consists of two links, l_1 and l_2. l_1 runs across two bridges, *A* and *B*. l_2 crosses one bridge, *C*. The failure of any bridge can cause the lightpath to fail. If we assign risk ID 2 to *A*, risk ID 5 to *B* and risk ID 3 to *C*, then {2, 5} is l_1’s risk vector and {3} is l_2’s risk vector. The lightpath’s risk vector is then {2, 5, 3}. With the concept of risk vector, the risk disjointing constraint requires that there must not be any common risk IDs in the risk vectors of a primary path and its protection path.

Another example is illustrated in Figure 2. There are two optical lightpaths in a DWDM network, l_1 = *abc* and l_2 = *abdc*. Link *ab* of l_1 and link *ab* of l_2 are on the same fiber between node *a* and *b*, therefore both links are subjected to the same fiber failure. We can assign a common risk ID 2 to the fiber *ab*. Their common node *b* is also assigned a risk ID 10.

Link *bc* of l_1 and link *bd*, *dc* of l_2 are on different fibers in disjoint terrain. The risks of fiber failure are different here. We assign a risk ID of 4 to *bc*, 5 to *bd* and 6 to *dc*. Combining the risks of all resources, we have l_1*'s* risk vector {2, 10, 4} and link l_2*'s* risk vector {2, 10, 5, 6}. It is clear that l_1 and l_2 do not satisfy the risk disjointing constraint because of their common risk ID 2 and 10.

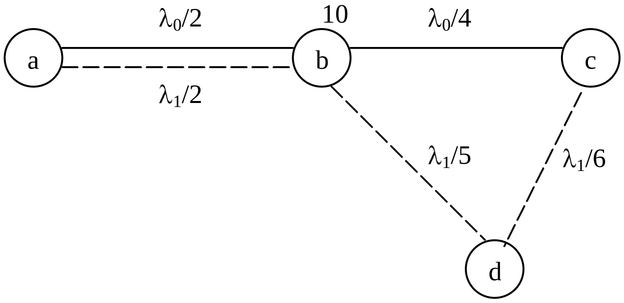


Figure 2

Requirement 2. Both the primary path and protection path must be routed in order to claim success. This requirement applies to dedicated protection as well. It is up to the network operator to handle the failure. For instance, the operator may decide that the request is blocked, or may make the primary path unprotected. Due to the risk disjointing constraint, if the risk IDs of the primary path's risk vector appear on many unused resources, those resources will have to be excluded from the protection path routing. Clearly, our routing algorithm should have a preference for the resources with uncommon risk IDs. This applies to the routing of both the primary path and protection path.

Requirement 3. If a resource is already taken by a protection path, that resource should be shared as much as possible by subsequent protection paths, up to the maximum number allowed on that resource. The purpose is to reduce the number of total resources taken by protection paths in the network. Therefore already shared resources should be given higher preference for routing the protection path.

Requirement 4. If multiple protection paths share common resources, those protection paths should not activate simultaneously. In order to achieve this, the routing algorithm must disallow protection paths from sharing common resources if their primary paths have common elements in their risk vectors.

Requirement 5. For multiple routing requests, we can process the requests either one at a time, or all at once. The latter is similar to static routing and has a higher chance of obtaining more optimal routes. But in a distributed network, routing requests often arrive at different nodes of the network. It is more practical and simpler to route requests one at a time. Once we develop the algorithm for a single request, we can handle the multi-request case by running iterations of the algorithm and choosing the most optimal routes.

Requirement 6. Specific networks may impose additional requirements. For instance, a DWDM network without wavelength conversion has the wavelength continuity constraint, in which case we may need to run iterations of the algorithm, one for each wavelength.

The heuristic routing algorithm we are proposing is a link state based shortest path routing algorithm. The development of such an algorithm is motivated by the wide deployment of link state routing algorithms in the internet today, i.e., OSPF and IS-IS. Every node has global network topology and complete information of every link in the network. In addition to generic link state information, all nodes have information on every link about,

1. The link’s risk IDs.
2. Whether the link is already taken by a primary path.
3. Whether the link is running a protection path. If so, the risk IDs of the primary paths are also known. If many protection paths share this link, the amount of data for this item is potentially large.
4. The maximum number of shared protection paths the link supports. By lowering this number, we can decrease the amount of data for item 3.

If it is desirable to cover node failure, network nodes can also be assigned risk IDs. The IDs are then included in the risk vectors of the links adjacent to the nodes.

Based on the above information, we will modify the link costs such that the shortest path algorithm generates the routes that meet all of the requirements. The algorithm is run at the source node and returns explicit routes.

We now describe the heuristic algorithm in C style pseudo code. First, to route a primary path:

```
Route_primary(s, d, net)  // s – source node, d – destination node, net – network topology
{
  for all link l in net
  {
    if (l is already taken by a primary or protect path)
      $c_l$ = infinity;  // step 1. $C_l$ is l’s cost
    else
    {
      $c_l = f(c_l, n_{rl})$; // step 2. Increase its cost if a link has risk ID with high occurrence
                         // $r_l$  - l’s risk ID
                         // $n_{rl}$ – the number of occurrences in the network of l’s risk  ID;
                         // f() may be either linear or non-linear.
      if (existing primary paths has $r_l$ )
        $c_l = g(c_l)$;  // step 3. Increase $c_l$ if the link has common risk ID of existing primary paths
    }
  }
  Dijkstra’s shortest path algorithm(s, d, net);
}
```

Both step 2 and 3 potentially increase the degree of sharing when routing the protection path, but step 3 requires the source node to be aware of the risk IDs of all existing primary paths, which may be a difficult task.

After routing the primary path, we now route its protection path:

```
Route_protection(s, d, net)
{
  for all link l in net
  {
    if (l is already taken by a primary path)
      $c_l$ = infinity;  // step 1
    else if (l is already taken by the maximum number of protection paths)
      $c_l$ = infinity;  // step 2
    else if (the primary path’s risk vector contains $r_l$)
    {
      if (risk disjoint constraint is strictly enforced)
        $c_l$ = infinity;  // step 3.1
      else if (risk disjoint constraint is not strictly enforced)
        $c_l$ = large number;  // step 3.2. There may be times when network operator allows
                                  // primary path and protection path to have common risk Ids
    }
    else if (there is common risk ID among the primary path’s risk vector and the primary
            path’s risk vector of a protection path on l )
    {
      if (Requirement 4 is strictly enforced)
        $c_l$ = infinity;  // step 4.1
      else if (Requirement 4 is not strictly enforced)
        $c_l$ = large number;  // step 4.2.
    }
    else
    {
      $c_l = f(c_l, n_{rl})$; // step 5. Adjust $c_l$ based on $r_l$’s occurrence. Increase its cost if a link has
                      // risk ID with high occurrence in the network
      if (l is running protection path )
        $c_l = g(c_l, n_{pl})$;  // step 6. Reduce $c_l$
    }
  }
  Dijkstra’s shortest path algorithm(s, d, net);
}
```

Now we illustrate the algorithm in an example with the network shown in Figure 3.

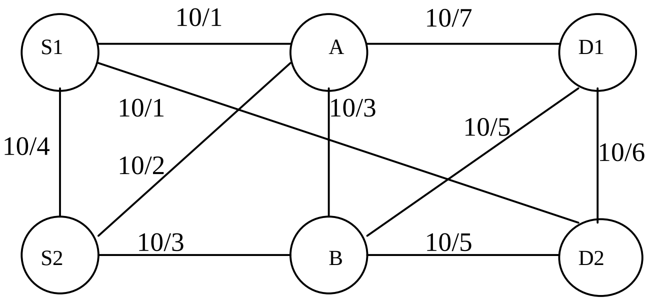


Figure 3

There are two routing requests. The first request asks for a path from node *S1* to node *D1*. The second request asks for a path from *S2* to *D2*. Each link has a cost of 10 and risk IDs as marked in the figure. All links and paths are bi-directional. There is no node disjoint requirement. With dedicated protection, one of the paths would have to be unprotected.

With shared protection, *S1-D1* is routed first. For the primary path, since it is the first path in the network, we only need to modify the link costs based on risk ID occurrence before running the shortest path algorithm. For each extra occurrence of a risk ID, we increase the link cost by 10%. The resulting network is shown in Figure 4. Running the shortest path algorithm yields a primary path *S1-A-D1*. Its risk vector is {1,7}. The total cost is 20.

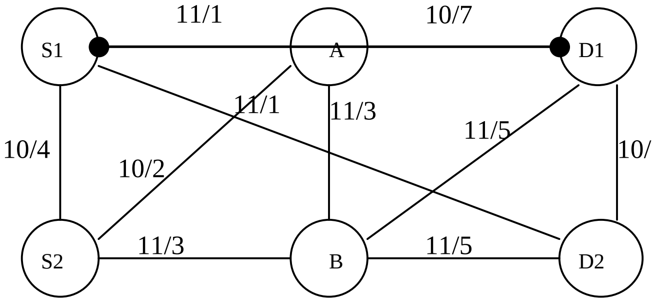


Figure 4

Next we route *S1-D1*'s protection path. Since it is the first protection path in the network, we only need to remove the primary path and the links with common risk IDs with the primary path from Figure 4. We then obtain the protection path *S1-S2-B-D1*, its risk vector is {3, 4, 5}. This path satisfies the risk disjointing constraint, and its total cost is 30.

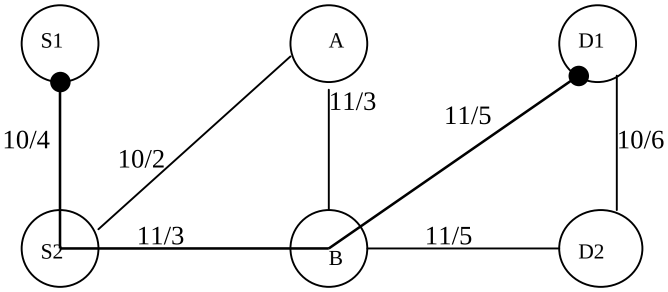


Figure 5

We now process the second request. To route the primary path from *S2* to *D*2, we need to remove the links on primary path *S1-A-D1* and its protection path *S1-S2-B-D1* from Figure 4. We also need to increase the cost on links that have common risk ID with the risk vector of primary path *S1-A-D1*, {1,7}. The resulting network topology is shown in Figure 6. the shortest path algorithm yield the second primary path *S2-A-B-D2*. Its risk vector is {2, 3, 5}, and its total cost is 30.

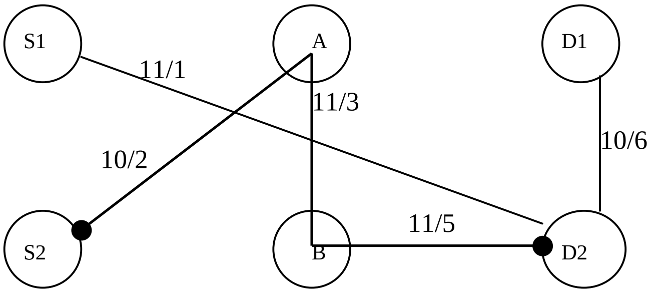


Figure 6

To route the protection path, we need to remove all links of the two primary paths from Figure 4 and links with risk ID 2 or 3 or 5. Then we decrease by 10% the cost on links that are running the first protection path. The resulting topology is shown in Figure 7. The protection path becomes *S2-S1-D2*. Its risk vector is {1, 4}, and its total cost is 20.

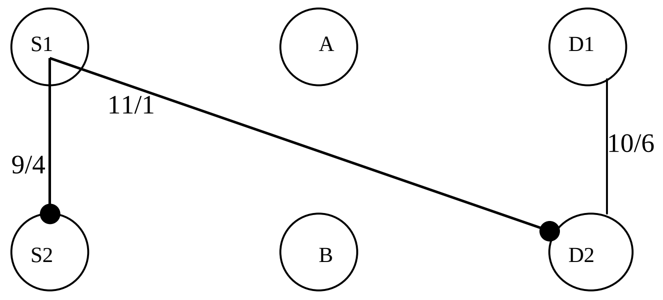


Figure 7

Finally we obtain the network with all paths routed as shown in Figure 8. Solid lines indicate the primary path; dash lines indicate the protection path. Link *S1*-*S2* is shared by two protection paths. With shared protection, both primary paths are protected, which is infeasible with dedicated protection.

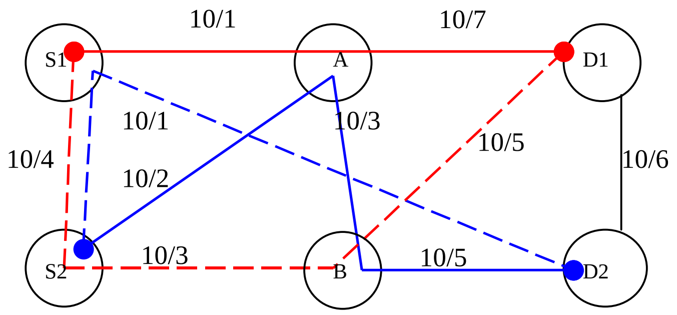


Figure 8

3. SIMULATIONS

To evaluate the performance of the shared protection routing algorithm, we develop a simulation. The simulation is run on networks of various connectivity to verify utilization improvement from our shared protection algorithm. As shown in Figure 9, network *a* is a six node ring. Network *b* has six nodes as well, and each node has a connectivity of three. Network *c* is a fully meshed six node network. It is assumed that each link has the same cost but unique risk ID. The capacity of each link in all the networks is 8, which means there are at most eight non-shared paths on a link. There is no node disjoint requirement.

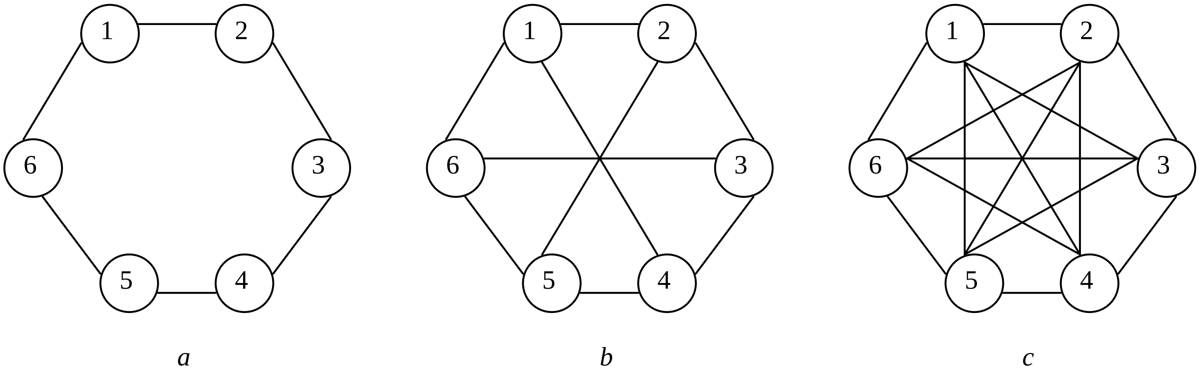


Figure 9. Networks with various connectivity

We randomly generated 500 source-destination pairs as the routing requests. Then we compared the number of successes for dedicated protection and shared protection. For shared protection, we also changed the maximum number of shared protection paths, M, allowed on each link. The results are plotted in Figure 10, and confirm our earlier assertion that shared protection offers higher network utilization than dedicated protection.

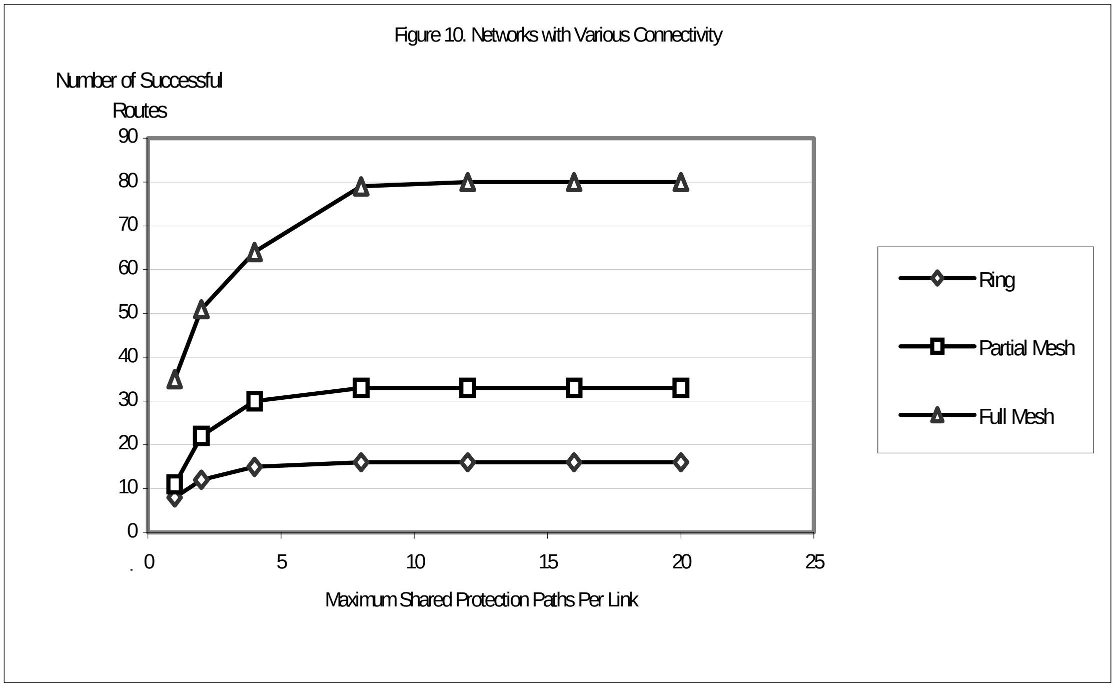


Figure 10. Networks with Various Connectivity

Next we run our simulations using the 16-node, 25-link NSFNET backbone topology as shown in Figure 11. The cost of every link is assumed to be 100, and the risk IDs are as marked in the figure. There is no node disjoint requirement.

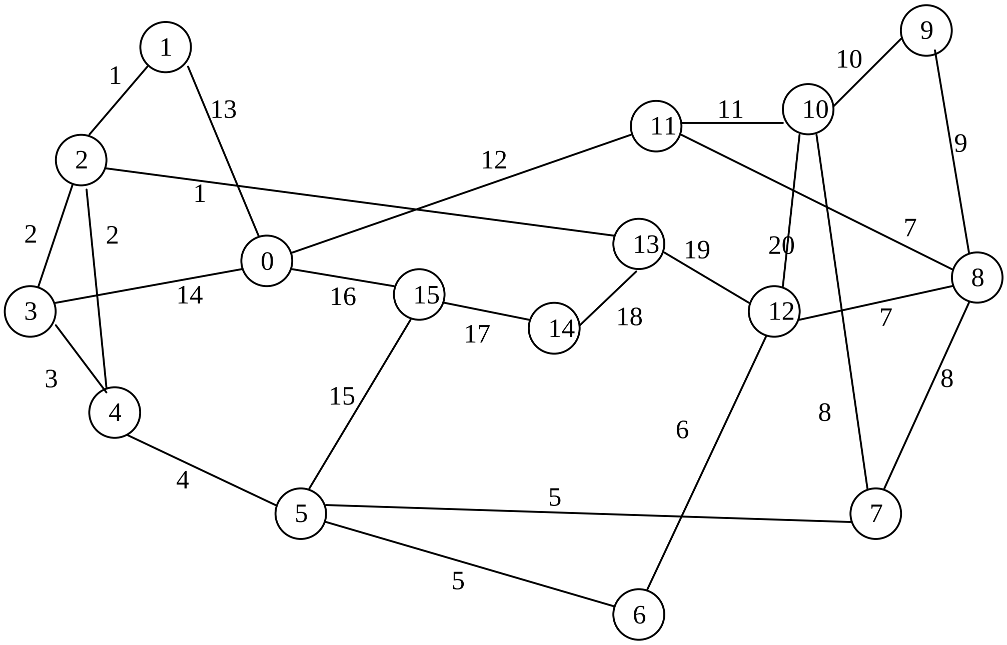


Figure 11. 16 Node NSFNET Backbone Network

Let the capacity on each link be C. A primary path takes one unit of capacity, as does a dedicated protection path. When multiple protection paths share a common link, they take one unit of capacity. We run simulations with various values of C.

We use standard Dijkstra’s Shortest Path algorithm for dedicated protection and the heuristic algorithm described earlier for shared protection.

For shared protection, we increase a link’s cost by 100% for each occurrence of its risk ID in the network when routing primary and protection paths. A link’s cost may also be decreased when routing a new protection path, if the link already has protection path running through it. We compare the effects of three functions:

- Function a. No modification to link cost
- Function b. Decrease link cost by 50% if it already has protection path
- Function c. Set link cost to zero if it already has protection path

We randomly generate 500 source-destination pairs as the routing requests, and we compare the number of successes for dedicated protection and shared protection. For shared protection, we also change the maximum number of shared protection paths, M, allowed on each link. We then run the simulations for 100 iterations.

We list the simulation results in Table 1. The numbers are the average numbers of successful routes for the 100 iterations. The first row of data contains the numbers of path pairs being successfully routed with dedicated protection, with various link capacities, C. The remaining rows contain the results for shared protection, with different link capacities, and sharing degrees, M.

Protection Type		Number of Path Pairs Routed											
		C = 2			C = 5			C=10			C=20		
Dedicated		7.11			17.6			35.81			71.76		
Shared	Cost Function	a	b	c	A	b	C	A	b	c	a	b	c
	M=2	9.81	9.71	9.32	26.21	26.1	24.55	52.6	52.63	49.97	102.87	103.54	99.23
	M=4	12.41	12.42	12.14	33.68	33.57	31.84	67.99	67.87	64.54	134.04	133.62	125.1
	M=8	12.67	12.92	13.06	34.78	35.37	35.79	70.44	72.36	72.65	137.77	141.57	141.18
	M=16	12.67	12.92	13.06	34.78	35.38	35.84	70.45	72.94	72.65	137.77	141.63	141.59
	M=32	12.67	12.92	13.06	34.78	35.38	35.84	70.45	72.43	72.94	137.77	141.63	141.59

Table 1. 16 Node NSFNET With Various Cost Functions

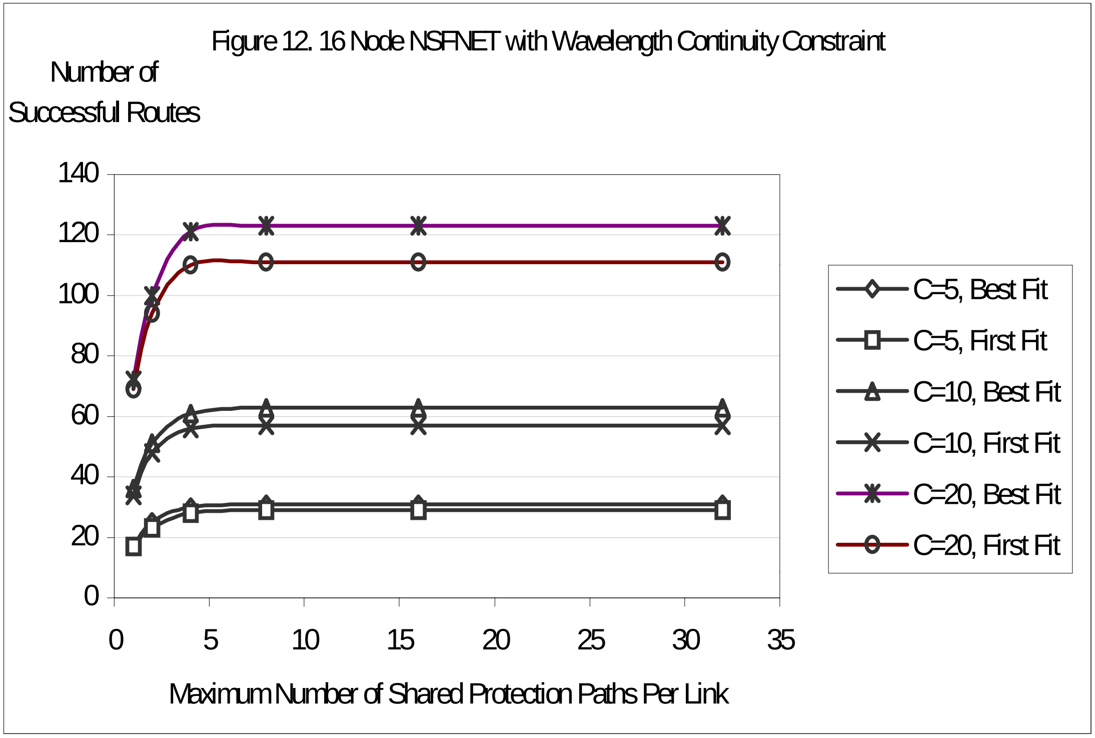
Function a. No Modification to Link Cost

Function b. Reduce Cost by 50% for Shared Links

Function c. Set Cost to 0 for Shared Links

Three observations can be made from the results. First, once again, our shared protection routing algorithm routes more requests than dedicated protection. It is also worth noting that network utilization can increase fairly significantly even with the minimum amount of sharing. For example, when the link capacity is 10, about 36 primary-protection path pairs are routed successfully under dedicated protection. But with only a maximum sharing of two per link, about 53 pairs can be routed under shared protection, an increase of nearly 50% in network utilization. Secondly, in shared protection, higher degrees of sharing beyond 8 do not provide significant additional gains of network utilization. Thirdly, various cost functions have similar effects on the utilization. Nevertheless, for lower values of M, it is better to have no modification of link cost (function a), while for higher values of M, setting link cost to 0 (function c) performs best.

We run the last simulation using the same NSFNET topology with the addition of wavelength continuity constraint, assuming that the network is a WDM network. For wavelength selection, we choose Best-Fit and First-Fit and compare their results. With Best-Fit, we choose a wavelength that gives us the lowest total path cost among all the wavelengths that have connectivity from the source to the destination. With First-Fit, we choose the first wavelength that gives a path from the source to the destination. We also choose cost function a. The results are plotted in Figure 12 and they are consistent with the results of previous simulation. The results also show that Best-Fit outperforms First-Fit because Best-Fit chooses wavelength that uses less network resource than First-Fit.



4. ISSUES AND IMPROVEMENTS

Additional Data and Computation Requirements. Compared with the algorithm for dedicated protection, the proposed algorithm for shared protection requires resources of protection paths to have the extra knowledge of the risk IDs of the related primary paths. If the maximum number of shared protection path allowed is M, and the number of nodes in the network is N, then on each link, the number of risk IDs is on the order of O(M*N). In order to find out whether the paths have any common risk IDs with the target primary path, the algorithm needs an extra O(NlogN + (M*N)log(M*N)) computations on each link.

Path Removal. When a primary path and its protection path are torn down, the resources that were once occupied are freed up. If we reroute the remaining paths, we may obtain more optimal routes [8]. This applies to both dedicated protection and shared protection.

With either type of protection, rerouting primary routes may cause traffic hits. It may be more practical to reroute only the protection paths from time to time.

Protection Activation. When multiple protection paths share common resources, only one can be activated at a time. In order to allow multiple activation, we can do one of the following after a protection path is activated:

- Reroute the failed primary path. Once the new primary path is established, move traffic onto it from the protection path, then deactivate the protection path. This approach requires one reroute, plus signaling for path deactivation. The probability of traffic hit is high when traffic is moved to the new primary path.
- Leave the traffic on the activated protection path and make it the new primary path. Establish a new protection path for it, as well as rerouting all other protection paths that shared common resource(s) with it. This approach doesn't introduce traffic hit, but it requires rerouting multiple protection paths as well as signaling associated with the rerouting.

The network operator should decide which option to take. If the end user has a high tolerance for traffic hits, the first approach is clearly more suitable, since, when the network utilization is relatively high, rerouting multiple protection paths has higher failure probability than rerouting only the primary path. On the other hand, if rerouting multiple protection paths is not an issue, then the second approach may be considered.

Signaling. Shared protection requires more signaling than dedicated protection. In addition to path establishment and removal, protection activation needs extra signaling as described above. Signaling can be done either in-band or out-of-band, depending on the network type.

Maximizing Resource Sharing. *Requirement 4* prohibits resource sharing among protection paths whose primary paths have common risk IDs. However, if we break each primary path into multiple segments, we may find risk disjointing segments of the primary paths. We can then establish segment-based protection instead of path based protection, and achieve higher sharing this way [9].

5. CONCLUSION

In this paper we proposed a shared protection routing algorithm for optical network and other connection-oriented networks. Simulations confirmed that, with this algorithm, shared protection provides a decent level of protection with less network resource than dedicated protection. Once deployed, shared protection complements the current two level protection of no protection or dedicated only protection.

This paper also addressed some of the issues that shared protection faces. The most obvious one is the extra signaling, data, computation, and path rerouting required by the proposed algorithm. Simulations suggest that practical implementations may use a small degree of sharing to reduce the extra expense while still achieving higher network utilization. More research is certainly needed to study the tradeoffs and the optimal implementations.

6. REFERENCES

[1] Rajiv Ramaswami, Kumar N. Sivarajan, “Optical Networks”, pp.430-434

[2] Hui Zang, Biswanath Mukherjee, “Connection management for survivable wavelength-routed WDM mesh networks”, Optical Networks Magazine, July/August 2001, pp.17-28

[3] Levy, D.S.; Sarathy, J.; Younghye Kwon; Al-Salameh, D.Y., “Optical layer shared protection using an IP-based optical control network “, Optical Fiber Communication Conference and Exhibit, 2001. Vol.2, 2001, pp.TuO8-1-TuO8-3

[4] Arijs, P.; Demeester, P. “The merit of shared and dedicated protection WDM rings in a hybrid network design”, Optical Fiber Communication Conference, 2000, Vol.4, pp.93 -95

[5] S. Ramamurthy, B. Mukherjee, “Survivable WDM Mesh Networks: Part I – Protection”, Proceedings, IEEE INFOCOM ’99, Vol. 2, 1999, pp.744-751

[6] Christian Mauz, “Allocation of Spare Capacity for Shared Protection of Optical Paths in Transport Networks”, Third International Workshop on Design of Reliable Communication Networks (DRCN2001)

[7] Ching-Fong Su, Xun Su, “Protection path routing on WDM networks”, OFC2001, TuO2-1.

[8] Vishal Anand, Chunming Qiao, “Dynamic Establishment of Protection Paths in WDM Networks: Part I”, Proceedings, 9th International Conference on Computer Communication and Networks (IC3N 300), pp.198-204

[9] Pin-Han Ho, H.T. Mouftah, “SLSP: A new path protection scheme for the optical Internet”, OFC2001, TuO1-1