A Scheme for Fast Restoration in MPLS Networks

Radim Bartoš and *Mythilikanth Raman*
Department of Computer Science, University of New Hampshire
Durham, NH 03824, USA. Phone: (603) 862-3792, Fax: (603) 862-3493
E-mail: {rbartos,mraman}@cs.unh.edu

Abstract: *MultiProtocol Label Switching (MPLS) has emerged as the technology to address the rapidly growing demands on routing in the Internet. The increase in mission-critical and real-time traffic carried by the Internet has created a need to significantly improve the reliability of IP/MPLS-based networks. Existing link and network layer protection mechanisms are either limited in scope or have higher convergence times. MPLS layer appears to be a suitable level at which fast, network-wide service restoration can be achieved.*

This paper proposes an MPLS restoration scheme that aims at protection of an entire domain as a way to reduce the overall cost of providing the protection. The Two Path algorithm is proposed to place the protection paths such that every node has two separate protection paths. It is shown that no single link failure would result in simultaneous loss of connectivity on both protection paths. The algorithm has a low polynomial complexity and therefore is scalable even for larger networks. The results of simulation study show that the proposed scheme can provide domain protection at a lower cost than the recently introduced Fast Reroute scheme. Furthermore, it is shown that the proposed scheme exhibits lower increase in delay that the traffic experiences after it has been switched on the protection paths.

Keywords: Fault-tolerant design, network design, MPLS, service restoration.

1. Introduction

The bandwidth of the transport networks has continued to increase while the latency introduced by the routers acts as a bottleneck in the Internet. The current high-speed backbone networks run at terabit rates, carrying high volume of data traffic. Recently proposed MultiProtocol Label Switching (MPLS) technology [1, 2] reduces the routing time, thus facilitating further increases in the data rates, and supports traffic engineering. Failure in the networks and interruption of service is becoming less acceptable for many applications. Future networks will require robust mechanisms to recover from failure.

Service restoration consists of the various mechanisms used to detect failure, reroute traffic, recover from the failure, and revert back to the state before the failure. Service restoration may involve setting up of the backup protection paths and the resource allocation for those paths. Traditional SONET-layer protections provide fast restoration times at the expense of inefficient use of the bandwidth and are limited to the ring topologies. Network layer restoration schemes have typically poor recovery times. For example, current IP networks facilitate re-routing based on the convergence of routing protocols which may take seconds to minutes. During this transient period, large volumes of data may be dropped, as well as, loops may occur in the network. It is generally accepted that desirable failure recovery time should be of the order of tens of milliseconds [3].

2. MPLS Restoration

MPLS networks use the label-switching paradigm to establish *Label Switched Paths* (LSPs) to route data traffic. LSPs are unidirectional in nature and are created by concatenating one or more label switched hops, allowing packets to be forwarded from one *Label Switching Router* (LSR) to another LSR across the MPLS domain. Note that the terms *path* and *LSP* will be used interchangeably throughout this paper. Similarly, the terms *node*, *router*, and *LSP* will refer to the same concept.

MPLS appears to be a suitable place in the layer stack to perform network-wide service restoration. Moreover, with the provision for fast switching using labels, it is intended that MPLS should also provide the fast restoration of the LSPs to be widely accepted for deployment [4]. With an MPLS restoration, IP traffic can be directly carried over the WDM optical channels without the need for traditional SONET layer protection mechanisms.

Current MPLS restoration methods are based on *path protection*. This paper introduces the concept of *domain protection* in MPLS. Two restoration mechanisms, *Fast Reroute* scheme [5] and *RSVP Backup Tunnels* [6], have recently been proposed. RSVP Backup Tunnels is an extension to the existing RSVP protocol for establishing LSP tunnels. It is used to shift traffic when better paths are available. The same concept can be applied to protection as well. The rest of this section outlines the Fast Reroute scheme.

A method for setting up alternate label switched paths to handle the fast rerouting of the traffic has been proposed in an Internet Engineering Task Force

(IETF) draft [5]. Fast Reroute (FR) mechanism involves setting up of two path segments. The *reverse segment* runs in the opposite direction of the working path from the egress to the ingress. The *alternate segment* runs from the ingress to the egress through nodes that are node- and link-disjoint with the working path. These two segments put together form the backup path for a LSP.

In the case of a failure of a link along the working LSP, the immediate predecessor LSR reroutes the traffic along the reverse segment towards the ingress. The ingress links the alternate segment of the backup path with the reverse segment, thereby transmitting traffic on the backup path towards the destination.

The major disadvantage of FR is the number of alternate paths that has to be established for every ingress and egress pair. This number grows as the number of the ingress and egress increases: for a network of n nodes and considering all the nodes to act as ingress and egress, the number of backup LSPs required is $O(n^2)$. Furthermore, sending traffic in the reverse direction may result in the reordering of packets.

3. Two Path Algorithm

The proposed algorithm attempts to reduce the overall cost of the protection by calculating the protection path placement for the entire domain as opposed to separate protection for each individual ingress to egress router path. During a single execution of the algorithm, protection paths are established for traffic originating in any of the ingress routers destined to one particular egress router. A full domain protection is achieved by executing the algorithm for every egress router

The standard goal of MPLS protection schemes is to protect the domain against a single link failure. Provided that the links are not placed in shared conduits, a multiple link failure is a relatively unlikely scenario. Furthermore, most schemes provide with a high probability full protection even in the case of multiple faults. Node faults are not considered in MPLS protection schemes since routers in backbone networks are typically highly reliable devices with multiple layers of built-in fault protection.

An MPLS domain D can be modeled as a quadruple $D = (N, L, I, E)$, where N is the set of n nodes, L the set of l links between the nodes, I the set of ingress routers $i_0, i_1, \ldots, i_{|I|-1}$ in the domain, $I \subseteq N$, and E the set of egress routers in the domain, $e_0, e_1, \ldots, e_{|E|-1}$, $E \subseteq N$.

The Algorithm

Input: The MPLS domain $D(N, L, I, E)$ and egress router $e \in E$. For simplicity we consider the worst case scenario where $I = N$ (the algorithm can be easily modified to handle the more general cases). Furthermore, it is assumed that graph $G(N, L)$, that represents the topology of the domain, has edge connectivity $e_c(G) \geq 2$. This condition guarantees that the domain can be protected against any single link failure.

Output: The two collections of protection paths running towards egress router e.

Step 1:

1.1 Using well-known Dijkstra's algorithm, calculate the shortest path spanning tree (SPST) of graph G rooted in egress router e.

1.2 Let P be the set of nodes for which the protection paths have been established. Initially it contains the egress router: $P = \{e\}$.

Step 2:

2.1 Select one of the branches of the SPST attached to the egress node and mark all its nodes except for the egress node.

2.2 Examine all marked nodes to find node i that has a link to an unmarked node[1] j. Such a node can always be found: If there was no link connecting the marked nodes of the branch of the spanning tree to unmarked nodes, removal of the link that connects the egress node to the branch would be sufficient to break the graph into two components. This would violate the above assumption that the network has edge connectivity $e_c(G) \geq 2$ and no single edge removal could break the graph into two components.

2.3 Consider a ring path consisting of the links of the SPST leading from e to i, the link between i and j, and the links of the SPST between j and e (note that this segment of the ring is empty in the case $j = e$). No link of the resulting path is visited more than once since the links of the first and last segments belong to different branches of the SPST and the link that connects the two branches is by definition not a part of the SPST. Similarly, the path visits every node exactly once, since paths from e to i and from e to j are two distinct branches of the SPST and therefore cannot share nodes. Hence, it can be concluded that the path created is a ring passing through the egress node.

2.4 Construct two protection paths along the ring: one clockwise, the other in a counterclockwise direction. The paths originate in the two nodes of the ring that are adjacent to the egress router and follow the ring all the way to the egress node as shown in Figure 1. Merge the created protection paths with the protection paths established in the previous iterations of the algorithm for the protected nodes that are now a part of the egress node.

[1] The concept of finding a link that connects two branches of a spanning tree have been independently proposed in the context of failure protection in ATM networks [7].

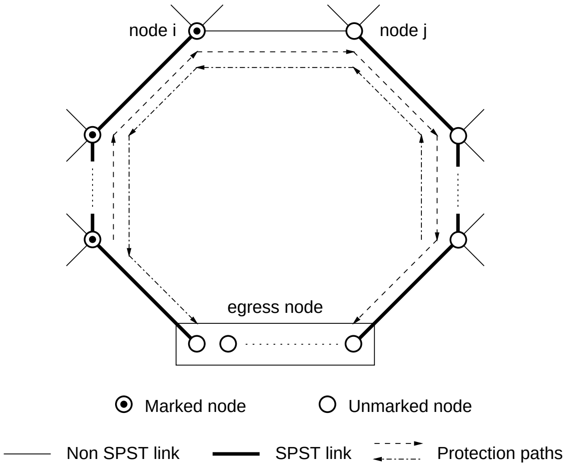

Figure 1: Protection path placement.

2.5 Since all nodes on the ring are now connected to both protection paths, they are considered protected and therefore added to the set of protected nodes P.

2.6 Construct a new graph by collapsing all nodes in P into a single node that becomes the egress node and by removing all the edges that connect two protected nodes. Note that nodes on the ring are, with the exception of link (i, j), linked with the edges of the SPST. It can be easily shown that a graph G' created by the collapse of a SPST link of graph G has (a) edges that are on the SPST in G are also edges of a SPST in G' and (b) $e_c(G) \leq e_c(G')$. By repeated application of these observations, it can be concluded that links of the SPST of the original graph can be considered as links of a SPST of the new graph and the new graph has edge connectivity $e_c(G) \geq 2$.
Repeat Step 2 for the new graph or stop if there are no unprotected nodes left in the network.

Theorem 1 *The above algorithm generates protection path placement that guarantees that no single link failure will cause a node to be simultaneously disconnected from its working path, as well as, both of the protection paths.*

Proof outline: No link failure can disconnect the egress node from itself. In a subsequent step of the execution, a ring originating at a protected node and terminating at a protected node is identified. Protection paths are placed along the ring and merged with the already existing protection paths of the protected nodes. In the case of a failure of a link on the ring, the nodes on the ring are split into two groups. Nodes in one of the groups are protected by one of the protection paths while the other protection path is disconnected by the failure. Nodes in the other group are protected by the alternate protection path. No other link failure in the network could affect the already existing protection paths among the protected nodes and could have an impact on the nodes on the ring. Therefore it can be concluded that the nodes on the ring are protected. Subsequent steps of execution connect more nodes to protection paths until a full domain protection with respect to the given egress node is achieved. □

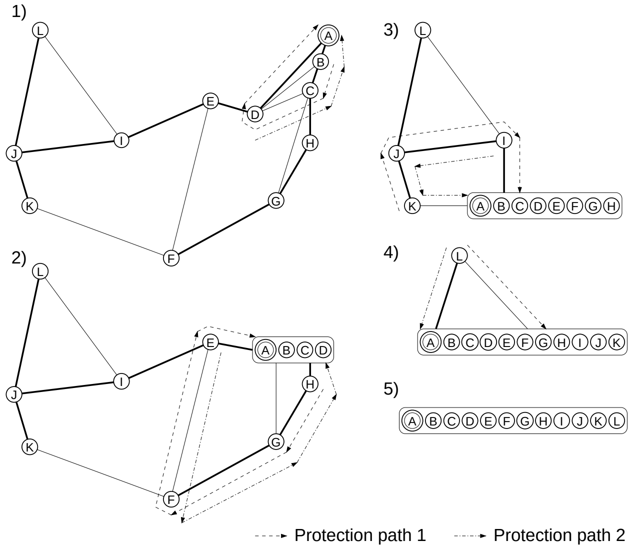

Figure 2: An example of protection path construction: vBNS topology.

The algorithmic complexity of Step 1 is $O(n^2)$. In the worst case, each iteration of Step 2 adds only one node to the set of protected nodes P and therefore Step 2 is executed no more than n-times. Each iteration of Step 2 involves visiting every neighbor of every unprotected node in the network. Hence, it can be concluded that the complexity of the algorithm is no worse than $O(nl)$, where l is the number of links in the domain. For a fully-interconnected graph, where $O(l) = O(n^2)$, the complexity of the algorithm is $O(n^3)$. However, typical backbone network topologies consist of bounded degree nodes and in such cases the resulting complexity remains $O(n^2)$.

In order to achieve a complete protection of the entire domain, the algorithm has to be executed for every egress router. Given the fact that the calculation of protection paths for individual routers is completely independent, it is expected that every egress router will execute the algorithm for itself. Such distributed calculations are commonly used, for example, to calculate routing tables in the Internet. The domain topology information required for the algorithm can be obtained from the standard routing protocols such as OSPF without a need for a new protocol. It has to be noted that the algorithm does not attempt to find an optimal protection path placement. Currently, the algorithm makes arbitrary choices when a branch of SPST is chosen in Step 2.1 and when a link to an

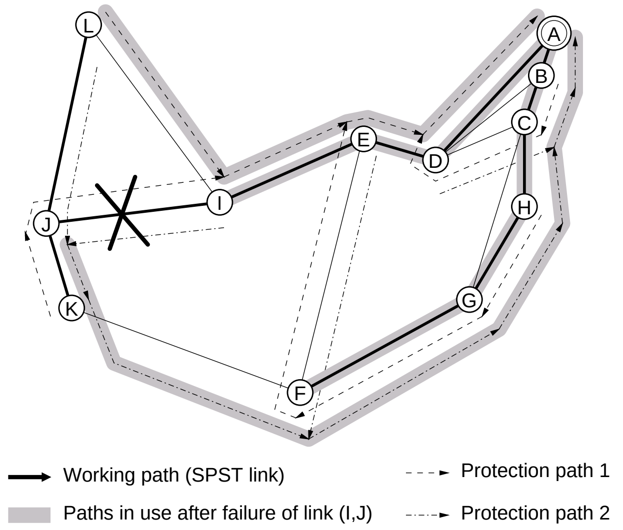

Figure 3: Protection path placement and path utilization after failure of link (I, J).

unmarked node is selected in Step 2.2. Several heuristics are currently being evaluated by the authors to improve the quality of the protection path placement.

Figure 2 shows the iterations of the algorithm calculating protection path placement for a network with topology of vBNS backbone [8] with unprotectable leaf nodes removed. The algorithm is executed with node A as the egress router and all other nodes of the network acting as ingress routers. Figure 3 gives the resulting protection path placement, as well as, the utilization of protection paths after a link failure.

Protection Mechanism Overview

The two paths calculated using the proposed protection path construction algorithm are setup using traffic engineering. These paths have the egress router as their root and span all the nodes in the domain. Once the paths are established, apart from the working path, every node will have two alternate paths to reach the egress node. When a link failure occurs, every node in the domain will be served by at least one of the two protection paths. The nodes on the path have to detect the failure of a working path and immediately switch over to the backup paths.

The signaling of the link failure is done using the recently proposed *Failure Indication Signal* (FIS) [3]. The node connected by the failed link transmits the FIS signal. The FIS propagates hop by hop towards all the ingress routers. The propagation flows through a *Reverse Notification Tree* (RNT) which is created along with the working path. Assuming that the working path forms a multipoint-to-point tree with the ingress nodes as the leaves and the egress node as the root, the RNT is the point-to-multipoint tree in the opposite direction. Signaling data flowing along the RNT will be given higher priority over other traffic

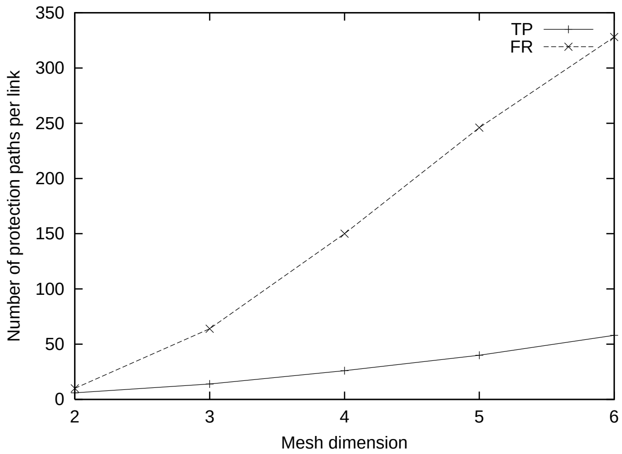

Figure 4: Maximum number of protection paths per link for meshes of varying dimensions.

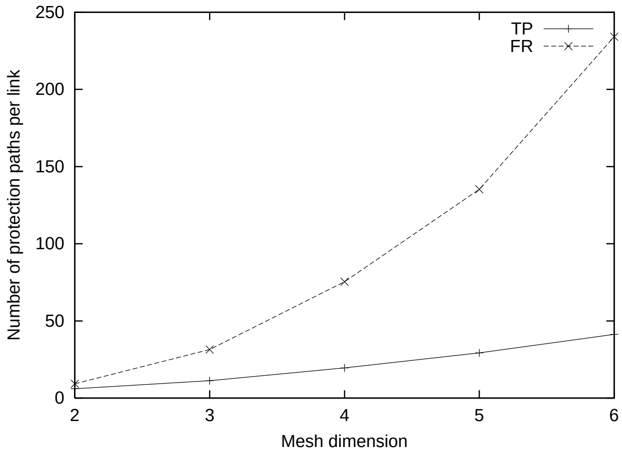

Figure 5: Average number of protection paths per link for meshes of varying dimensions.

to ensure the minimum delay of the FIS. When the nodes on the RNT receive the FIS, they stop transmitting data on the working path and the transmission is switched over to the backup paths.

4. Performance Evaluation

The proposed two path algorithm (TP) and fast reroute scheme (FR)[2] were simulated using a simulator written in C++. *Qt*, *Gtool*, and *GTL* libraries[3] were used for representation and visualization of the networks.

The quality of the protection path placement is evaluated using two measures. The first measure considers the length of protection paths as an indication of the delay the traffic will experience after a link failure. The average of all protection path lengths for the entire domain approximates the overall impact of a fault on the

[2]This paper considers 1:1 variant of FR, 1:n option is currently studied by the authors.

[3]Qt: *"http://www.trolltech.com/products/download/freelicense/qtfree-dl.html"*,
Gtool: *"http://erwin.math.lakeheadu.ca/gtool/"*,
GTL: *"http://infosun.fmi.uni-passau.de/GTL/"*.

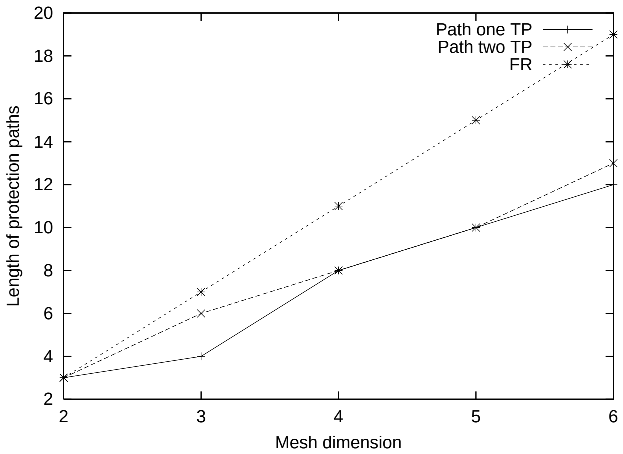

Figure 6: Maximum length of protection paths per node for meshes of varying dimensions.

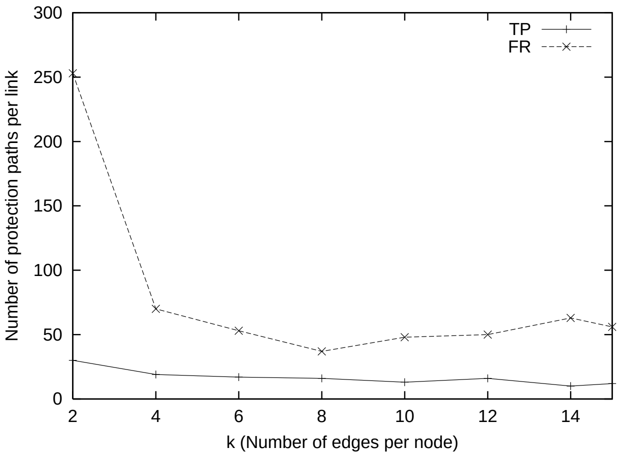

Figure 8: Maximum number of protection paths per link for $R_{16}(k)$.

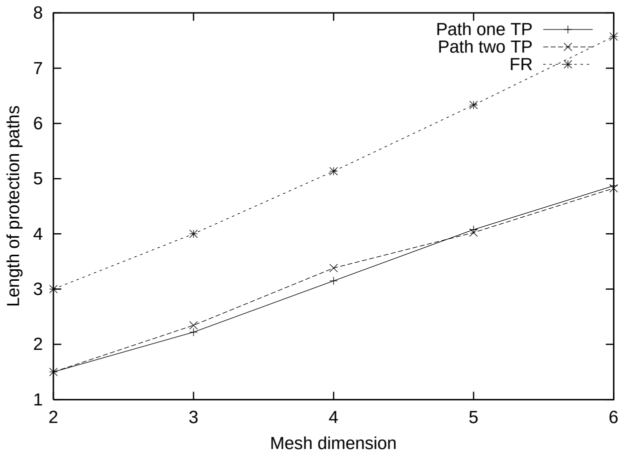

Figure 7: Average length of protection paths per node for meshes of varying dimensions.

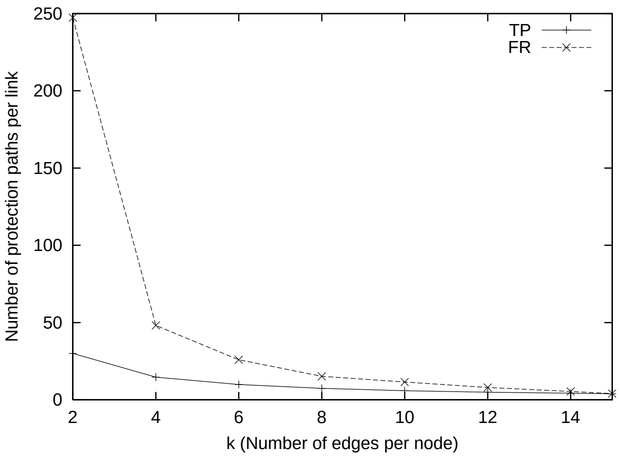

Figure 9: Average number of protection paths per link for $R_{16}(k)$.

traffic streams. The maximum protection path length gives the worst case scenario, an important measure for real-time traffic provisioning. In addition to the introduced delay, the length of protection paths reflects the amount of resources required.

The second measure considers the number of protection paths that pass through a link as an indication of the amount of resources that are required to setup and maintain the protection. The average and the maximum of the number of protection paths per link is also an indication of how well the protection paths are distributed throughout the network. A good protection path placement algorithm should generate paths with small average lengths and a maximum that is close to average. Additionally, the protection paths should be evenly distributed over the links of the domain.

Mesh topologies with varying dimensions were selected for one set of simulations since they resemble the backbone topologies and can be scaled. Varying the mesh dimensions served the purpose of visualizing trends as the number of nodes in a network increases. Figure 4 provides a comparison of the maximum number of protection paths that were placed on a link using the two methods. Figure 5 shows the results for the average number of paths per link. It can be inferred from the results that the two path algorithm improves as the network grows larger. The maximum and average lengths of the protection paths for the schemes are shown in Figures 6 and 7. It can be seen that TP performs better than FR in both cases and the improvement is more pronounced as the number of nodes increases.

The following topology was chosen to study the effects of varying connectivity of the network: Let $R_n(k)$, $k = 2, 4, ..., n-1$, represent a network with n nodes, labeled $0, 1, ..., n-1$, where node i, $0 \le i < n$, has links to nodes $i \ominus \lceil \frac{k}{2} \rceil$,..., $i \ominus 2$, $i \ominus 1$, $i \oplus 1$, $i \oplus 2$,..., $i \oplus \lceil \frac{k}{2} \rceil$ ($\oplus, \ominus$ represent addition/subtraction modulo n). From Figures 8, 9, 10, and 11, on simulation of a network with 16 nodes, it can be seen that except for the cases where the network is almost or fully interconnected, TP performs better than FR. The average length of the protection paths are less than that of FR. The number of links per node is also compar-

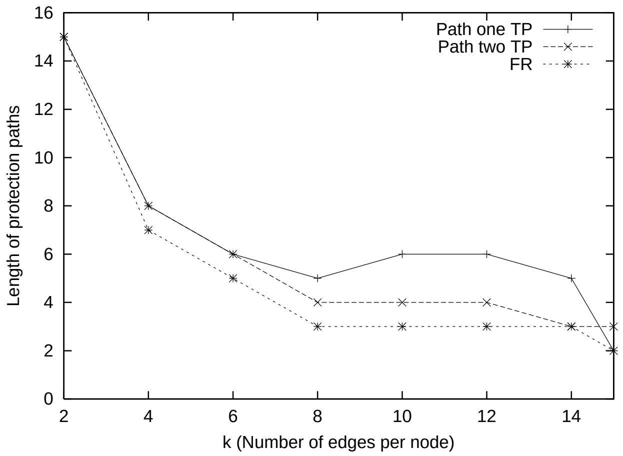

Figure 10: Maximum length of protection paths per node for $R_{16}(k)$.

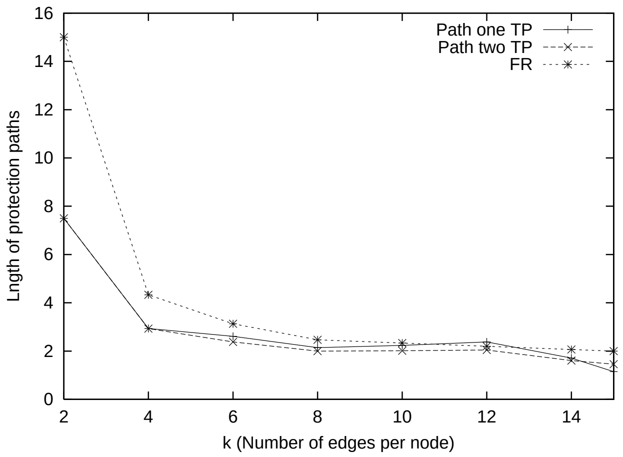

Figure 11: Average length of protection paths per node for $R_{16}(k)$.

atively less, especially in the cases of networks with lesser connectivity.

Two additional topologies shown in Figure 12 were simulated and the results are given in Table 1. It can be seen that with the exception of the maximum path length on Topology 2, the Two Path algorithm performs better than the Fast Reroute scheme for all other parameters and topologies.

5. Conclusion

This paper has proposed a novel scheme for protection of MPLS domains. Protection of an entire domain is considered as opposed to traditional single path protection. A new algorithm is proposed to calculate the protection path placement. The algorithm has low polynomial complexity and therefore can be used even for larger networks. The results of the simulation study show that the proposed algorithm can provide domain protection that uses less resources and introduces smaller delay in the case of a failure than recently proposed Fast Reroute scheme.

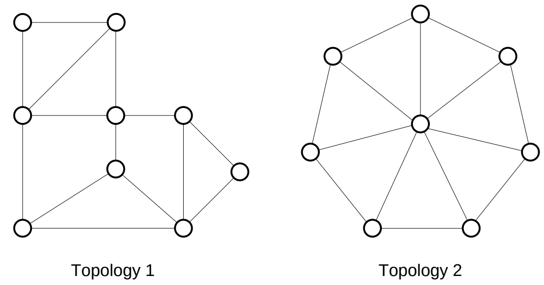

Figure 12: Network topologies used in simulations.

	Topology 1		Topology 2	
	TP	FR	TP	FR
Max. PPs per link	14	37	11	26
Avg. PPs per link	9.6	23.3	7.6	17.5
Max. PP length	6/6	7	5/4	4
Avg. PP length	2.2/2.3	3.5	2.5/1.9	2.8

Table 1: Simulation results for topologies shown in Figure 12 (PP - protection path).

References

[1] E. C. Rosen, A. Viswanathan, and R. Callon, "MultiProtocol Label Switching Architecture," *Work in progress [draft-ietf-mpls-arch-07.txt]*, July 2000.

[2] T. Li, "MPLS and the Evolving Internet Architecture," *IEEE Communications Magazine*, Volume 37, pp. 38-41, December 1999.

[3] S. Makam, et al, "Framework for MPLS-based Recovery," *Work in progress [draft-makam-mpls-recovery-frmwrk-01.txt]*, July 2000.

[4] G. Swallow, "MPLS Advantages for Traffic Engineering," *IEEE Communications Magazine*, Volume 37, pp. 54-57, December 1999.

[5] D. Haskin and R. Krishnan, "A Method for Setting an Alternative Label Switched Paths to Handle Fast Reroute," *Work in progress [draft-haskin-mpls-fast-reroute-04.txt]*, May 2000.

[6] D. O. Awduche, L. Berger, D. Gan, T. Li, G. Swallow, and V. Srinivasan, "Extensions to RSVP for LSP Tunnels," *Work in progress [draft-ietf-mpls-rsvp-lsp-tunnel-04.txt]*, September 1999.

[7] R. Slosiar and D. Latin, "A Polynomial-Time Algorithm for the Establishment of Primary and Alternate Paths in ATM Networks," *Proceedings of the IEEE Infocom*, Tel-Aviv, Israel, March 2000.

[8] J. Bannister, J. Touch, A. Willner, and S. Suryaputra, "How many wavelengths do we really need? A study of the performance limits of packets over wavelengths," *Optical Networks Magazine*, Volume 1, No 2, pp. 17-27, April 2000.