Fault Monitoring and Restoration in Optical WDM Networks

Jian Wang[+], Laxman Sahasrabuddhe[⊕] and Biswanath Mukherjee[⊗]
[+]*Department of Applied Science, University of California, Davis, CA 95616, USA*
[Å]*SBC Service, Inc., 2600 Camino Ramon, San Ramon, CA 94583, USA*
[⊗]*Department of Computer Science, University of California, Davis, CA 95616, USA*
[+]*Correspondance Author: Tel: +1-530-752-5129, Fax: +1-530-752-4767, Email:* jnwang@ucdavis.edu

1. Introduction

Generalized multiprotocol label switching (GMPLS) is a set of protocol extensions to multiprotocol label switching (MPLS) that are essential for enabling next-generation IP-over-WDM networks [1-5]. Here, we consider an IP-over-WDM network in which the network nodes consist of GMPLS-capable optical crossconnects (OXCs) and GMPLS-capable IP routers (henceforth, the term "IP router" will mean a "GMPLS-capable IP router"). An IP router can employ GMPLS to setup an optical connection, called an optical label-switched path (optical-LSP), from itself to another IP router in the network.

In an IP-over-WDM network, the failure of a fiber link can lead to the failure of all the optical-LSPs that traverse the fiber. Each optical-LSP is expected to operate at a rate of a few Gbps (or a few tens of Gbps); hence, the network designer must provide an efficient fault-management technique that combats fiber failures. Fault monitoring and management techniques are essentially of two types: (a) *protection* and (b) *restoration* [6-8]. In *protection*, spare capacity is reserved during call setup. In *restoration*, the spare capacity that is available after the fault's occurrence is utilized for rerouting the disrupted connections. In this work, we study restoration techniques that operate at optical-LSP level.

Recently, there has been a considerable amount of standards activity within the Internet Engineering Task Force (IETF) towards establishing a fault-management framework for MPLS [9-11]. So far, the standardization efforts have focused on protection techniques, because protection techniques allow service providers to offer "hard" guarantees on the recovery time, e.g., 50 ms recovery time in SONET. However, many new data-centric services, such as virtual private network (VPN) services, may not require such "hard" guarantees on the recovery time. Moreover, restoration techniques have many advantages over protection techniques. For example, restoration utilizes bandwidth more efficiently than protection because, in restoration, the resources for the backup connections are not committed until the fault actually occurs. Another important advantage of restoration is that it can naturally handle multiple simultaneous fiber failures, whereas protection techniques are designed for handling a preset number of simultaneous failures, typically single fiber failures. Thus, by implementing restoration techniques, service providers can broaden their service portfolio to support varying degrees of service guarantees, based on the customer requirements. A recent IETF draft [12] proposed a hybrid fault-management approach.

In our current study, we focus on the implementation details and performance comparisons of different restoration techniques. We try to keep our implementations as close as possible to the existing Internet drafts on GMPLS and MPLS. No signaling extension is proposed. This paper will show how to handle each fiber failure by itself; however, it should be clear that our implementations of the restoration techniques can handle multiple independent fiber failures, whose occurrences may be asynchronous and uncorrelated, as well as node failures, which is a special case of multiple fiber failures. To the best of our knowledge, there has not been any work that compares different kinds of restoration techniques by performing a detailed simulation implementation of the signaling protocols. We have developed a detailed simulation platform for an IP-over-WDM network [13], so that we can comprehensively test and accurately compare the various restoration techniques and the protocol implementations.

2. Fault Monitoring and Restoration Techniques

Providing fault-monitoring functions within OXCs is attractive, especially for transparent (all-optical) switches. Today, fault-monitoring functions are usually provided by the optical-transmission systems. For the wavelength channels using SONET framing, the B1 bit in the SONET overhead can be used to measure the bit-error rate. For other formatted optical channels, the optical-power loss can be used to detect certain failures, such as a fiber cut. In today's market, since most OXCs use electronic switching fabric, optical-electrical-optical (OEO) conversion is used before each OXC port. Therefore, faults can be detected on link-by-link basis. Even for all-optical switches, both the end nodes of the failed link can detect the fiber cut.

2.1. Restoration Techniques

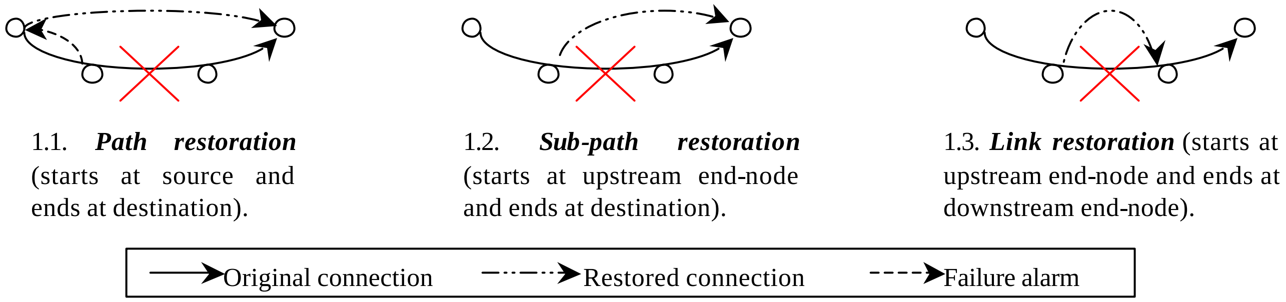
We describe three restoration techniques that can be built into GMPLS: (a) *path restoration*, (b) *sub-path restoration*, and (c) *link restoration*.

In *path restoration*, when a fiber fails, the upstream end-node of the failed fiber sends an *alarm* message to the source node of the disrupted optical-LSP, while the downstream end-node sends a *teardown* message to the destination node of the optical-LSP (the end-node of the failed fiber that is closer to the source (destination) node is called the upstream (downstream) end-node). After the source node receives the *alarm* message, it tries to re-establish the optical-LSP by sending a *setup* message to the destination node via an alternate path, as shown in Fig. 1.1.

In *sub-path restoration*, when a fiber fails, the upstream end-node *does not* send an *alarm* to the source node of the disrupted optical-LSP; instead, it tries to patch the optical-LSP by sending a *setup* message to the destination node. Meanwhile, the downstream end-node sends a *teardown* message to the destination node of the optical-LSP (see Fig. 1.2). Note that our definition of the term "sub-path" is a little different from other usage of this term in the recent literature [14, 15].

Finally, in *link restoration*, when a fiber fails, the upstream end-node does not send an *alarm* message to the source node; the downstream end-node *does not* send a *teardown* message to the destination; instead, the upstream end-node tries to reroute the optical-LSP around the failed fiber link by sending a *setup* message to the downstream end-node (see Fig. 1.3).

Note that, in all three restoration techniques, if the restoration attempt fails, then the optical-LSP is dropped.


1.1. ***Path restoration*** (starts at source and ends at destination).

1.2. ***Sub-path restoration*** (starts at upstream end-node and ends at destination).

1.3. ***Link restoration*** (starts at upstream end-node and ends at downstream end-node).

Figure 1. Three restoration schemes.

2.2. Implementation of the Restoration Techniques

We built a simulation platform based on ns-2 (http://www.isi.edu/nsnam/ns/). In our simulation platform, an optical-LSP is setup following the requirements in Reference [5]: downstream-on-demand label allocation and distribution, with an ingress-initiated ordered control. This platform also supports source-routing computation (source node computes the end-to-end path) and explicit-routed-LSP setup (label-distribution protocol or LDP pins down the LSP according to the path calculated by the source node).

2.2.1. Contention Resolution

Source-routing computation normally is based on the resource (e.g., wavelength) availability information collected by the routing protocol; however, link-state update takes time. If an LSP-setup request follows too closely to a change, such as the setup or teardown of another optical-LSP, then the up-to-date link-state information may not be available at the source node of the new request. Although simply ignoring the ongoing requests may not be a bad choice during normal operation, it can cause significant performance degradation in case of a fiber failure because lots of requests are generated nearly simultaneously since a large number of connections could have been traversing a fiber before it failed.

Part of this problem can be solved relatively easily. When all the near-simultaneous requests come to a single node, which is the typical post-failure scenario for sub-path and link restoration, the common "source" node (upstream end-node in case of sub-path and link restoration) can increase its routing-computation accuracy by incorporating information on the pending LSPs. Explicit-path information on the pending LSPs is available at the LDP module of the source node. In the simulation platform, the routing protocol (where the computation is performed) will always query the LDP module for information on the pending optical-LSPs before making any explicit-routing computation. This information is then combined with the link-state information to do routing computation. By doing the above, the chance of having contention is greatly reduced.

The other part of the problem is a little more difficult to solve. When the sources nodes are different, which is the typical post-failure scenario for path restoration, instant coordination among the source nodes is very difficult, if not impossible. Our solution is to give the source nodes additional chances to retry when one restoration effort fails. We will examine the system's performance improvement under such retries.

2.2.2. Loop Prevention

The MPLS architecture [4, 16] does not assume a single LDP. There are two main LDP standards, namely the constraint-based-routing label-distribution protocol (CR-LDP) and the resource reservation protocol (RSVP). In either protocol, an optical-LSP must have an unique identification. To eliminate possible confusion and to conform to existing Internet drafts [12, 17], we choose to use the original LSP identification for the restoration purpose.

A loop in an LSP is defined as the LSP traversing a node more than once [18]. In sub-path and link restoration techniques, since the restoration path is identified as part of the original LSP, we have to ensure that these two parts do not overlap with each other at any node.

For data switching/forwarding purpose, an intermediate node in an LSP does not need to keep the full path information. Neither does standard LDP require this support [16]. When an intermediate node does restoration-path computation, there is a chance that the computed path will overlap with the remaining segments of the original path. In our simulation platform, the loop-prevention mechanism of LDP can detect and drop all the problematic restoration paths. Although the loops will not disturb the correct behavior of our protocol, this choice can affect the performance of these two restoration schemes. We are working on implementing traffic-engineering extensions to LDP so that each node along the original path can "remember" the full path. Once this extension is done, we can expect that most loops will be eliminated at the explicit-routing-computation stage.

3. Restoration Performance Results

3.1. Performance Metrics and Simulation Network

We study the following two performance metrics: *availability and restoration time*.

- *Availability* is defined as the ratio of the total uptime for an optical-LSP to the total duration of the optical-LSP.

- *Restoration time* is defined as the average repair time (from the instant a connection is disrupted to the instant the connection is restored) for all successfully restored paths.

We simulated the NSFnet topology (Fig. 2) with 16 wavelengths on each fiber, where one wavelength per fiber was used for control signaling, i.e., for sending GMPLS signaling messages. All nodes are capable of performing wavelength conversion. (Although not reported here, the extension to wavelength-continuous networks is straightforward.) The length of the fiber link between a pair of cities is chosen to be equal to the corresponding driving distance.

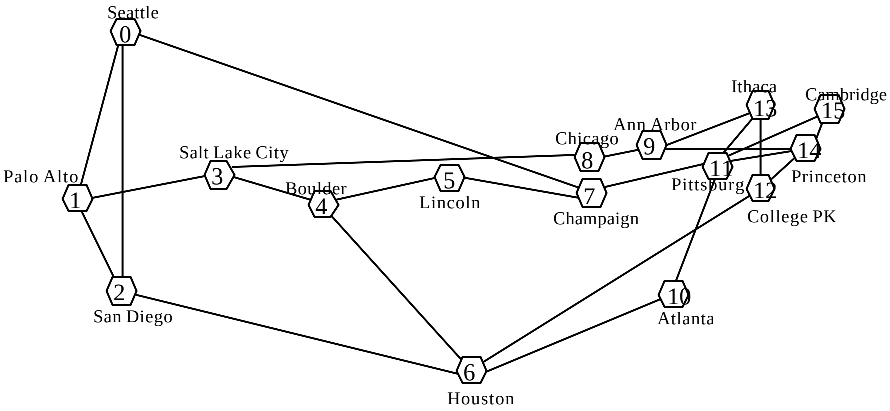


Figure 2. NSFnet topology used in our experiments.

In our simulation experiments reported here, the call inter-arrival time and the call-holding time are assumed to be exponentially distributed, while the source and destination nodes for a connection are uniformly distributed. Similarly, we assume that the inter-arrival time and the duration of the fiber faults are also exponentially distributed. The failed fiber is chosen randomly using an uniform distribution. Note that multiple fiber failures may occur in our experiments. In our experiments reported here, time is normalized to the average call duration, which is assumed to be unity (and equal to 50 seconds); average fault inter-arrival time is denoted by ***f*** normalized units; and the average duration of a fiber fault (or fault-repair time) is assumed to be 2 normalized units. The failure-detection time is assumed to be 1 ms. In our link-restoration experiments, the downstream segment of a failed connection is retained for 0.5 s after a failure is detected. If the request for the restoration path around the failed link comes to the downstream end-node within 0.5 s, then the restoration path will be connected to the original downstream segment. Otherwise, the downstream segment will be torn down by the downstream end-node of the failure.

We choose the parameters based on our projections about the future network with emerging technologies. We believe the parameters chosen here are close to realistic values but on the aggressive side (e.g., the failure arrival rate may be higher than that in a real network). If the failure rate is too low, limited by our current computation capacity, we are unable to get good statistics on the protocols' restoration behavior within a reasonable simulation time.

3.2. Performance Comparison of the Restoration Techniques

Figures 3.1 and 3.2 plot the performance metrics measured against the network load. In each figure, three curves are shown for path restoration, sub-path restoration, and link restoration. The value of ***f*** is chosen to be 10 normalized units. Note that, in this topology, a network load of 5 Erlangs corresponds to an average link load of about 3.5%, while a network load of 100 Erlangs corresponds to an average link load of about 70%. For the results reported below, we simulate about 60,000 calls for each simulation scenario (given a restoration technique and load, both metrics are measured in one run), and it takes 9 hours on a Pentium IV computer (1.45 GHz CPU, 512 MBytes memory) running the Linux operating system.

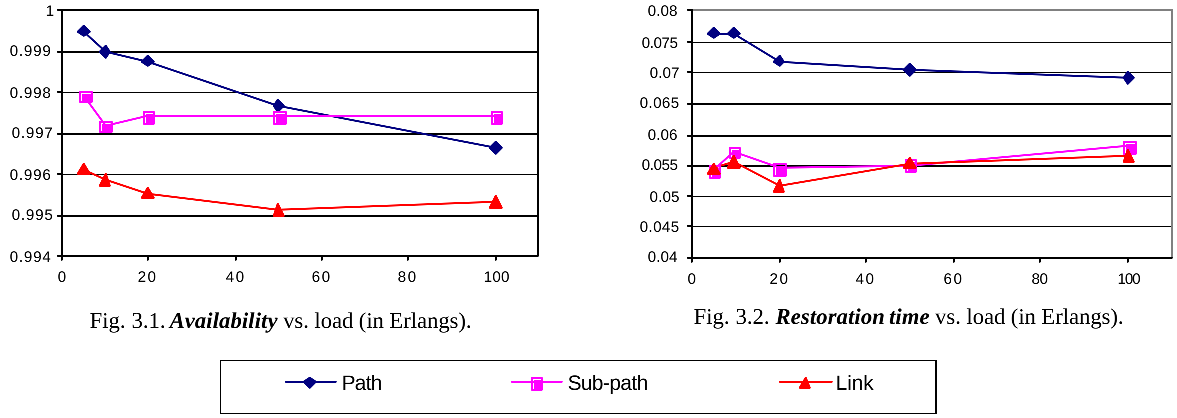


Fig. 3.1. ***Availability*** vs. load (in Erlangs).

Fig. 3.2. ***Restoration time*** vs. load (in Erlangs).

Figure 3. Performance comparison for different restoration techniques (***f***=10).

In Fig. 3.1, the network availability for the path-restoration technique is found to decrease with increasing network load (from about 99.95% at a load of 10 Erlangs to 99.65% at a load of 100 Erlangs). The service unavailable time mainly comes from those un-restored connections, since the down time for these connections are counted from when the disruption happens to the time when the request expires. With increasing network load, the chance of having a collision increases; and in turn, the total unavailable time also increases. The availability for sub-path and link restoration are around 99.8 and 99.6 percent, respectively, at light load, and increasing the network load does not seem to have any significant impact on these rates. For both of these techniques, the main reason for a restoration attempt to fail is the violation of the "no loop" rule (Section 2.2.2). The chance of having a loop does not increase with an increase of the network load. Link restoration has a lower availability than sub-path restoration because a restoration path may also form a loop with the downstream segment of the original LSP in link restoration, while in sub-path restoration, the downstream segment of the original path is torn down immediately after the failure. All the availability values we have measured are above 99.5%. When the load on the network is low, path restoration can achieve higher availability than both sub-path and link restoration. As the network load increases, path restoration gradually loses its competitive advantage to sub-path restoration due to intensified contentions.

Figure 3.2 shows the average restoration times. For path restoration, again, the curve goes down with increasing network load (from about 76 ms at 10 Erlangs to about 68 ms at 100 Erlangs). This is because the longer restoration paths have higher chance of being blocked. When contention intensifies, the survivors tend to be short. As expected, the restoration time for both sub-path and link restoration (both are around 55 ms) are shorter than that for path restoration. The simulation experiments show insignificant difference in restoration time between sub-path and link restoration on this NSF topology. The average hop distance for normal connections in this network is only about 2.2, so link restoration does not show much speed advantage over sub-path restoration. However, it is expected that this advantage can be greater in networks with larger node count and sparse connectivity.

3.3. Enhanced Path-Restoration Techniques

Performance of the path-restoration technique can be significantly improved by giving retrial opportunities to path restoration (referred to as enhanced path restoration hereafter). Recall that path-restoration failure is mainly caused by contention among restoration requests. By increasing the number of retrials allowed, we plot the performance metrics in Figs. 4.1and 4.2. The no-retrial curves are repeated from the Fig. 3. In this experiment, each retrial takes place 1 ms after the notification message, which is used to notify the failure of the previous restoration effort, reaches the source node. This delay has little impact on the restoration success rate (not shown) but an increase in this delay increases the restoration time.

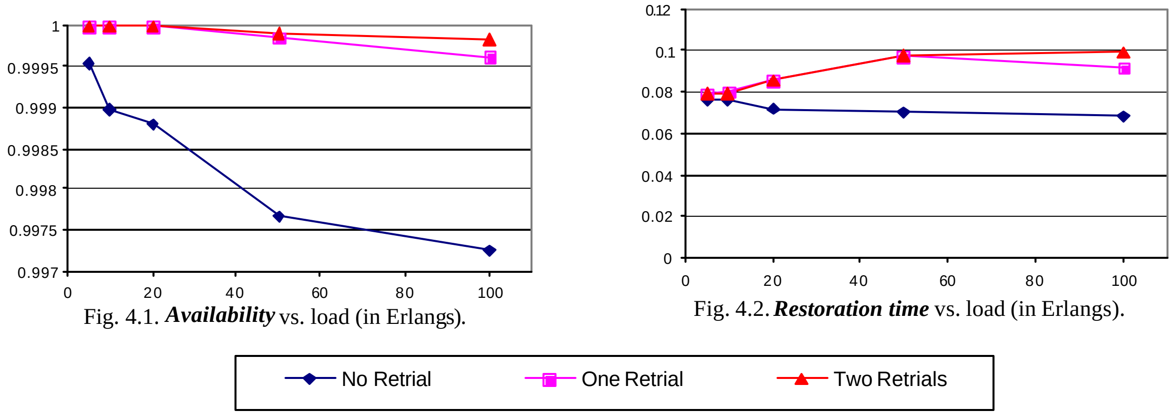


Fig. 4.1. ***Availability*** vs. load (in Erlangs).

Fig. 4.2. ***Restoration time*** vs. load (in Erlangs).

Figure 4. Performance comparison for enhanced path restoration with different retrial opportunities (***f***=10).

With just one retrial opportunity, the availability is found to jump close to 100% for most cases (Fig. 4.1). Even in the extreme case (network load equal to 100 Erlangs), the availability is still 99.97%. Increasing the number of retrials to two will further increase the availability at 100-Erlang load to about 99.98%. Additional experiments show that increasing the number of retrials beyond 2 will not yield much further improvement to the restoration success rate.

When the network load is low, say 5 Erlangs, the average restoration time for enhanced path restoration is about the same as that for path restoration (Fig. 4.2). When the network load approaches 100 Erlangs, the restoration times for one and two retrial cases increase to about 90 ms and 100 ms, respectively.

3.4. Effects of Fault Inter-Arrival Time

The failure frequency directly affects the network availability. Notice that the ***f***=10 value used in our experiments corresponds to a failure rate that is much more frequent than that in reality, so we are interested in seeing how these performance metrics will change when the failure rate decreases. Unfortunately, simulating real GMPLS signaling takes a lot of computation. Longer failure intervals (relative to average call-holding time) mean more call setup and teardown between two failures.

Unable to simulate for more realistic ***f*** values, we choose to illustrate our results by trying ***f***=20. Our simulations show that the availability is directly affected by the ***f*** value since the average unavailable time contributed by each fiber failure is largely unchanged. When the failures occur less frequently, availability increases accordingly.

4. Conclusion

In this study, we employed simulation to study three restoration techniques (path, sub-path, and link) using GMPLS signaling for fault management in an IP-over-WDM network. Then, we compared the performance of the three restoration techniques and an enhancement. A good restoration-technique design should deliver high availability for the entire network, as well as low restoration time for each disrupted connection. Our simulation experiments show that all restoration techniques can deliver availability higher than 99.5% and restoration time equal to a few tens of milliseconds under very frequent failures (***f***=10) in a nation-wide IP-over-WDM optical mesh network. By decreasing the failure frequency to more realistic values, we have observed significant increase in availability.

In general, our studies show that the enhanced path-restoration technique with an additional retrial for path restoration after a failure can achieve very good network availability, for the price of a little higher restoration time. For a non-mission-critical environment, the enhanced path-restoration technique has good potential to provide a simple, yet robust, optical-LSP service. Both sub-path and link restoration can provide restoration speed faster than path restoration; however, they have lower restoration success rate. We expect that this can be improved by employing some traffic-engineering extensions to LDP. The implementation of protection techniques using GMPLS signaling is also an important problem for future research.

Acknowledgement: This work has been supported in parts by Alcatel Research and Innovation (R&I); Bellsouth; Cisco Systems; UC Davis MICRO; and NSF Grant No. ANI-98-05285. We gratefully acknowledge the comments on and contributions to this research from Drs. Ljubisa Tancevski, David Elie-dit-Cosaque, and Maher Ali of Alcatel R&I.

References:

[1] A. Banerjee, et al., “Generalized multiprotocol label switching: an overview of signaling enhancements and recovery techniques,” IEEE Commun. Mag., vol. 39, no. 7, pp. 144-151, January 2001.

[2] N. Ghani, “Lambda-labeling: a framework for IP-over-WDM using MPLS,” Optical Networks Mag., vol. 1, no. 2, pp. 75-93, April 2000.

[3] C. Xin, T. Wang, Y. Ye, M. Yoo, S. Dixit, and C. Qiao, “On design and architecture of an IP over WDM optical network control plane,” Proc., IFIP TC6 Fifth Working Conference on Optical Network Design and Modelling, pp. 297-312, February 2001.

[4] E. Rosen, et al., “Multiprotocol label switching architecture,” RFC 3031, January 2001.

[5] P. Ashwood-Smith, et al., “Generalized muti-protocol label switching (GMPLS) architecture,” Work in progress, Internet draft, draft-ietf-ccamp-gmpls-architecture-01.txt, November 2001.

[6] S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks: Part I. Protection,” Proc., IEEE INFOCOM 1999, vol. 2, pp. 744 –751, March 1999; “Part II. Restoration,” *Proc., IEEE ICC’99*, vol. 3, pp. 2023-2030, June 1999.

[7] O. Gerstel and R. Ramaswami, “Optical layer survivability-an implementation perspective,” IEEE Journal on Selected Areas in Communications, vol. 18, no. 10, pp. 1885-1899, Oct. 2000.

[8] E. Modiano and A. Narula-Tam, “Survivable routing of logical topologies in WDM networks,” Proc. IEEE INFOCOM 2001, vol. 1, no. 3, pp. 348-357, April 2001.

[9] J. P. Lang, et al., “Generalized MPLS recovery mechanisms,” Work in progress, Internet draft, draft-lang-ccamp-recovery-01.txt, July 2001.

[10] V. Sharma, et al., “Framework for MPLS-based recovery,” Work in progress, Internet draft, draft-ietf-mpls-recovery-frmwrk-03.txt, July 2001.

[11] K. Owens, V. Sharma, V. Makam, and C. Huang, “Extensions to CR-LDP for MPLS path protection,” Work in progress, Internet Draft, draft-owens-crldp-path-protection-ext-02.txt, July 2001.

[12] D. Gan, et al., “A method for MPLS LSP fast-reroute using RSVP detours,” Work in progress, Internet Draft, draft-gan-fast-reroute-00.txt, April 2001.

[13] J. Wang, S. J. B. Yoo, and B. Mukherjee, “Design and development of an MPLambdaS simulator,” Proc. MPLScon’01, San Jose, pp. 139-146, March 2001.

[14] V. Anand, S. Chauhan, and C. Qiao, “Sub-path protection: A new framework for optical layer survivability and its quantitative evaluation,” Dept. of Computer Science and Engineering, State University of New York at Buffalo, Tech. Report 2002-01, Jan. 2002.

[15] C. Ou, H. Zang, and B. Mukherjee, “Sub-path protection for scalability and fast recovery in WDM mesh networks,” Proc. OFC 2002, March 2002.

[16] L. Andersson, P. Doolan, N. Feldman, A. Fredette, and B. Thomas, “LDP specification,” RFC 3036, January 2001.

[17] J. Ash, et al., “LSP modification using CR-LDP,” Work in progress, Internet draft, draft-ietf-mpls-crlsp-modification-03.txt, March 2001.

[18] Y. Ohba, Y. Katsube, E. Rosen, and P. Doolan, “MPLS loop prevention mechanism,” RFC 3063, February 2001.