

Path vs. Subpath vs. Link Restoration for Fault Management in IP-over-WDM Networks: Performance Comparisons Using GMPLS Control Signaling

Jian Wang, University of California, Davis

Laxman Sahasrabudde, SBC Service, Inc.

Biswanath Mukherjee, University of California, Davis

ABSTRACT

We investigate three restoration techniques (path, subpath, and link restoration) for fault management in an IP-over-WDM network. We have implemented all of these techniques on the ns-2 simulation platform using GMPLS control signaling. These techniques can handle practical situations such as simultaneous multiple fiber failures, which are difficult to design for and recover from by nonrestoration techniques. We then present performance measurement results for the three restoration techniques by applying them to a typical nationwide mesh network running IP over WDM. We investigate interesting trade-offs in the performance of the restoration techniques on restoration success rate, average restoration time, availability, and blocking probability.

INTRODUCTION

Generalized multiprotocol label switching (GMPLS) is a set of protocol extensions to MPLS that are essential for enabling next-generation IP-over-WDM networks [1, 2]. Here, we consider an IP-over-WDM network in which the network nodes consist of GMPLS-capable optical cross-connects and GMPLS-capable IP routers (henceforth, the term IP router will mean a GMPLS-capable IP router). An IP router can employ GMPLS to set up 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, failure of a fiber link can lead to failure of all the optical LSPs that traverse the fiber. Each optical LSP is expected to operate at a rate of a few (or a few tens) of gigabits per second; hence, the network designer must provide an efficient fault-management technique that combats fiber failures. Fault management techniques are essentially of two types:

- *Protection*
- *Restoration* [3, 4]

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 the optical LSP level.

Recently, there has been a considerable amount of standards activity within the Internet Engineering Task Force (IETF) toward establishing a fault management framework for MPLS [5]. So far, the standardization efforts have focused on protection techniques, because protection techniques allow service providers to offer "hard" guarantees on recovery time, for example, 50 ms recovery time in synchronous optical network (SONET). However, many new data-centric services, such as virtual private network (VPN) services, may not require such hard guarantees on recovery time. Moreover, restoration techniques have many advantages over protection techniques. For example, restoration utilizes bandwidth more efficiently than protection because, in restoration, the backup route is computed specifically for the failure, and the resources for the backup route are not committed until the fault actually occurs. Although protection may incorporate sophisticated preplanning methods, these methods usually try to protect against all possible failures. Another important advantage of restoration is that it can naturally handle simultaneous multiple fiber failures, whereas protection techniques are designed to handle 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 customer requirements. A recent IETF draft by D. Gan proposed a hybrid

fault management approach (please see the Appendix).

In our current study, we focus on 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, and only necessary changes to the state transaction machine are made. This article 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 failure, a special case of multiple fiber failures. We have developed a detailed simulation platform for an IP-over-WDM network, so we can comprehensively test and accurately compare the various restoration techniques and protocol implementations. By experimenting, we can be confident about the correctness of the protocol implementations, particularly the timing-related aspects of the distributed signaling protocols. Hence, our work is an important step toward the standardization and eventual deployment of GMPLS.

In [6], the authors provide a detailed comparison of several fault recovery strategies that employ preplanned path restoration to recover from single-link failures. In [7], the authors present experimental results for preplanned path restoration based on their implementation of GMPLS fast restoration.

RESTORATION TECHNIQUES

THREE RESTORATION TECHNIQUES

We describe three restoration techniques that can be built into GMPLS:

- *Path restoration*
- *Subpath restoration*
- *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 reestablish the optical LSP by sending a *setup* message to the destination node via an alternate path, as shown in Fig. 1a.

In *subpath 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 (Fig. 1b).¹

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 (Fig. 1c).

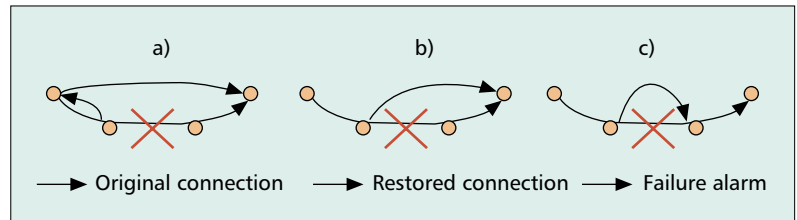


Figure 1. Three restoration schemes studied in this report: a) path restoration (starts at source and ends in destination); b) subpath restoration (starts at upstream end node and ends at destination); c) link restoration (starts at upstream end node and ends at downstream end node).

In all three restoration techniques, restoration optical LSPs are computed by the same algorithm that is used for the primary optical-LSPs. If the restoration attempt fails, then the optical-LSP is dropped.

SIMULATION PLATFORM

Our simulation platform is based on ns-2 (<http://www.isi.edu/nsnam/ns/>), which is a discrete-event-driven simulation platform targeted at networking research. Although ns-2 provides substantial support for simulating IP networks, it lacks the necessary modules for simulating next-generation IP-over-WDM networks. In order to simulate IP-over-WDM networks, we have extended ns-2 by:

- Adding an optical cross-connect (OXC) module
- Adding a WDM link module
- Extending the label distribution protocol (LDP) module (as specified in [10])
- Adding WDM-specific traffic-engineering (TE) extensions to the link state routing protocol module (as specified in [11])
- Adding the explicit route computation module for computing routes in wavelength-continuous and wavelength-convertible WDM networks

In addition, we have enhanced various modules in ns-2 to simulate the restoration techniques and protocols.

We tried to build our simulation model as close to real networks as possible. Our implementations and enhancements allow us to accurately simulate the transient phenomena in the network. For example, in path restoration, after a fiber failure, the source nodes of all the failed optical LSPs try to reestablish their connections, which may lead to several near-simultaneous events. In a heavily loaded network, this may cause contention among the various restoration requests. As will be shown later, this contention is captured by our detailed protocol implementation. Our implementation also helps us to check whether our proposed protocol extension can gracefully handle various corner cases, for example, the restoration protocols must handle the cases where a second fiber failure occurs (e.g., loss of an *alarm* message) while the network is recovering from the first fiber failure.

The original ns-2 code consisted of approximately 70,000 lines of C++ code and 20,000 lines of OTcl code. Our extensions to ns-2 consist of approximately 4500 lines of C++ code and 2000 lines of OTcl code.

¹ Note that our definition of the term subpath is a little different from other usage of this term in the recent literature. Two examples of alternate usage of the term are outlined below. In [8], a path is fragmented into nearly-equal-length parts to form a sequence of subpaths, each protected separately. At one extreme, when a path consists of only one subpath, this approach is equivalent to path protection; at the other extreme, when each subpath is a link, this approach is the same as link protection. In [9], a large mesh network is fragmented into a set of nonoverlapping areas, and a path traversing a sequence of such areas is fragmented into a sequence of subpaths, one per area. Each such subpath is protected independently within its own area. Both of the approaches in [8, 9] use subpaths with protection to reduce the protection switching time when a fiber failure occurs.

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.

RESTORATION TECHNIQUES: GENERAL PHILOSOPHIES AND IMPLEMENTATION

In this section we focus on several issues that affect restoration performance. In our simulation platform, the way an optical LSP is set up follows the requirement in reference [2]: 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 (LDP pins down the LSP according to the path calculated by the source node). Source routing is important for implementing various traffic engineering policies. In our IP-over-WDM network, for example, source routing can automatically circumvent the congested links by choosing a longer path (given the assumption that the link state information possessed by the source node is correct). Source routing is used throughout the following discussions.

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, 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 subpath and link restoration, the common source node (upstream end node in case of subpath 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 first queries the LDP module for information on the pending optical LSPs before making any explicit routing computation. Then the routing protocol computes the explicit route by treating the pending optical LSPs as if they existed in the network. 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 source nodes are different, which is the typical post-failure scenario for path restoration, coordination among the source nodes (so they would not compete for the capacity on the bottleneck links) is difficult, if not impossible. One solution is to allow the source nodes to retry when the initial restoration effort fails. We will examine the system's performance improvement, if any, under such retries.

Identification of the Restoration Optical LSPs — MPLS architecture [2] does not assume a single label distribution protocol. There are two

mainstream LDP standards, namely the Constraint-Based-Routing Label-Distribution Protocol (CR-LDP) and Resource Reservation Protocol (RSVP). No matter which is used, an optical LSP must have a unique identification. In RSVP, the identification is the combination of a SESSION object and a SENDER_TEMPLATE object; in CR-LDP, the identification is LSPID.

A question arises when assigning identification to a restoration LSP. Should the restoration path use the same identification as the original one or not? A new identification may cause confusion, especially when subpath and link restoration techniques are used. For example, the link restoration technique can create a LSP that consists of three segments. The first and last segments use the same old identification while the middle one uses a new identification. In both CR-LDP and RSVP, part of the identification is the source node identification. The semantics of path identification may be used by some protocol implementation, so leaving some LSP segments unattached to their original source is risky. Besides, concatenation of different optical LSPs needs major extensions to the existing protocols. We choose to use the original LSP identification for the restoration purpose. Our choice conforms to several related IETF drafts.

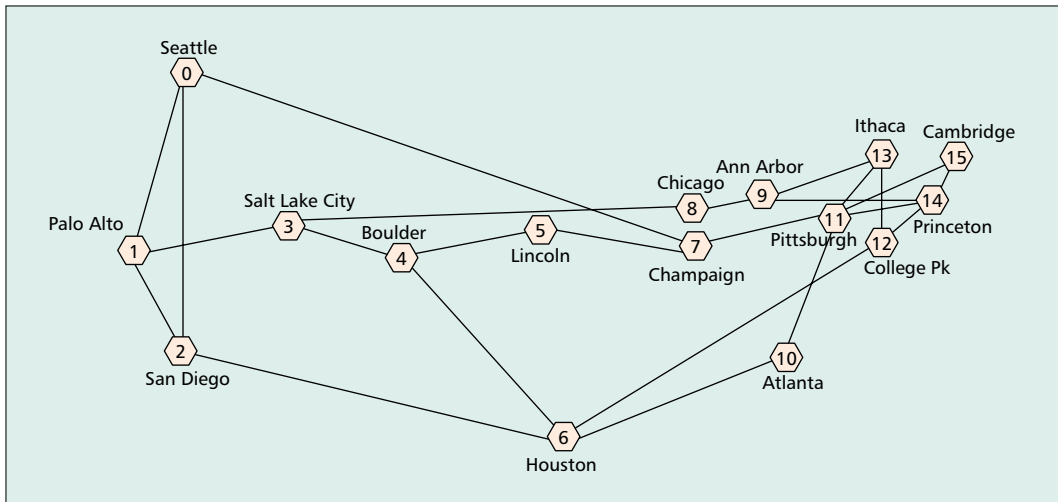
Loop Prevention — A loop in an LSP is defined as the LSP traversing a node for more than once [12]. In subpath and link restoration techniques, since the restoration path is identified as part of the original LSP, we have to ensure that these two parts do not overlap with each other at any node.

For data switching/forwarding purposes, an intermediate node in an LSP does not need to keep the full path information. Neither does standard LDP require this support [2, 10]. 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.

RESTORATION PERFORMANCE RESULTS PERFORMANCE METRICS AND SIMULATION NETWORK

We study the following four performance metrics: *restoration success rate*, *restoration time*, *availability*, and *blocking probability*:

- *Restoration success rate* is defined as the ratio of the total number of successfully restored connections to the total number of disrupted connections.



■ **Figure 2.** The NSFnet topology used in our experiments.

- *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.
- *Availability* is defined as the ratio of the total uptime for an optical LSP to the total duration of the optical LSP.
- *Blocking probability* is the ratio of the number of unsuccessful connection requests to the total number of connection requests in a network.

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. (We believe that the extension of our simulation platform to include wavelength-continuous networks is straightforward, and we expect the results of the simulation to be similar to those presented in this work.) The length of the fiber link between a pair of cities is chosen to be equal to the corresponding driving distance.

In our simulation experiments reported here, the call interarrival time and 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 interarrival time and duration of the fiber faults are also exponentially distributed. The failed fiber is chosen randomly using a 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 s); average fault interarrival 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, 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. In this work, all the restoration paths are nonreversible, that is, the connections will not be switched back to the original path when the failed links are physically repaired.

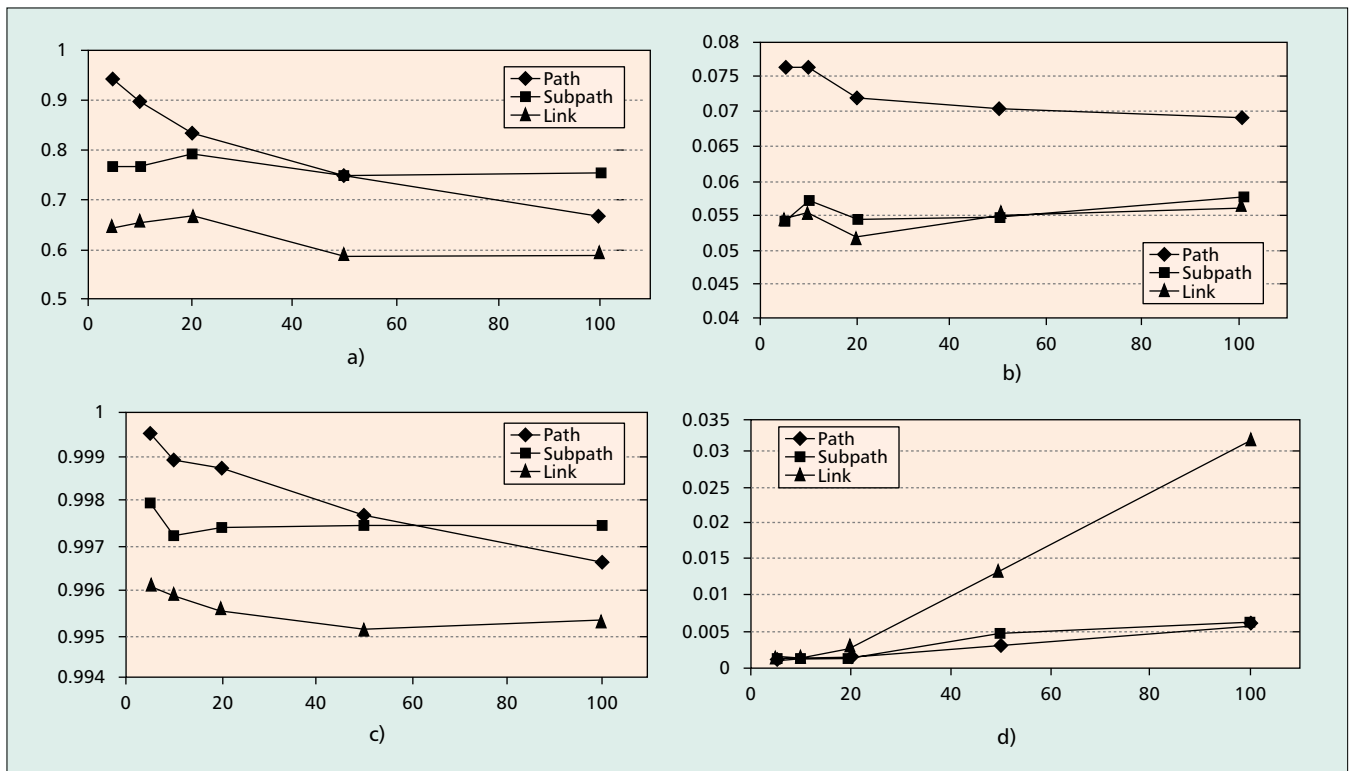
We wish to choose the network parameters based on our projections about the future network with emerging technologies. However, limited by our current computation capability (even with modern computers), we are unable to get good statistics on the protocols' restoration behavior within a reasonable time. Hence, we have chosen the related parameters such as call duration of a lightpath, failure repair time, and so on, to be a few orders of magnitude on the aggressive side. Also, relatively speaking, the failure arrival rate is chosen to be much more aggressive than in reality to test the robustness of our approaches as well as to obtain decent performance results within a reasonable amount of simulation time.

PERFORMANCE COMPARISON OF THE RESTORATION TECHNIQUES

Figure 3a–d plots the four performance metrics measured against the network load. In each figure, three curves are shown for path restoration, subpath 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 percent, while a network load of 100 Erlangs corresponds to an average link load of about 70 percent. For the results reported below, we simulate about 60,000 calls for each simulation scenario (given a restoration technique and load, all four metrics are measured in one run), and it takes 9 h on a Pentium IV computer (1.45 GHz CPU, 512 Mbytes memory) running the Linux operating system.

In Fig. 3a, the restoration success rate for the path restoration technique is found to decrease monotonically with increasing network load (from about 94 percent at a load of 10 Erlangs to 65 percent at a load of 100 Erlangs). The major cause for a path restoration attempt to fail is contention among restoration requests. With

The failure arrival rate is chosen to be much more aggressive than in reality to test the robustness of our approaches as well as to obtain decent performance results within a reasonable amount of simulation time.



■ **Figure 3.** Performance comparison for different restoration techniques ($f = 10$): a) restoration success rate vs. load (in Erlangs); b) restoration time vs. load (in Erlangs); c) availability vs. load (in Erlangs); d) blocking probability vs. load (in Erlangs).

the increase of network load, the chance of having a collision increases.

The restoration success rates for subpath and link restoration are around 76 and 63 percent, respectively, at light load. Increasing the network load does not seem to have any significant impact on these rates. For both subpath and link restoration techniques, the main reason for a restoration attempt to fail is violation of the “no loop” rule. Note that, in these two approaches, since only a segment is restored, the restored segment may intersect with the upstream segment that is not being restored, thus forming a loop. Link restoration has a lower success rate than subpath restoration because the restored segment may also intersect and form a loop with the downstream segment of the original LSP in link restoration, while in subpath restoration, the restored segment may intersect and form a loop with the upstream segment only. Subpath restoration outperforms path restoration in terms of success rate when the network load is high, because there is no contention among restoration efforts.

Figure 3b shows the average restoration times. For path restoration, again, the curve goes down with the increase of network load (from about 76 ms at a load of 10 Erlangs to about 68 ms at a load of 100 Erlangs). This is because the longer restoration paths have a higher chance of being blocked. When contention intensifies, the survivors tend to be short. As expected, the restoration time for both subpath and link restoration (both around 55 ms) are shorter than that for path restoration. The simulation experiments show insignificant difference in restoration time between

subpath 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 subpath restoration. However, it is expected that this advantage can be greater in networks with larger node count and sparse connectivity.

To carriers and users, the network availability is a very important quality of service (QoS) metric. In Fig. 3c, our experiments show that all restoration schemes can deliver availabilities higher than 95 percent. Because the unavailable time is mainly contributed by unrestored failures, availability is largely determined by the restoration success rate. When the load on the network is low, path restoration can achieve higher availability than both subpath and link restoration. As the network load increases, path restoration gradually loses its competitive advantage over subpath restoration due to intensified contentions.

Besides the post-failure performance, we are also interested in how the restoration schemes affect normal network operation. The blocking probability (which does not count the blocked restoration requests) performance shown in Fig. 3d gives us a good idea on this characteristic. Besides link restoration, the other two techniques give us similar blocking performance; however, link restoration results in a significant leap in blocking probability when the network load is high. Paths restored by link restoration tend to be further away from the optimal (with respect to path length), and they tend to take up more extra resources. This causes more likelihood for blocking for later connection requests.

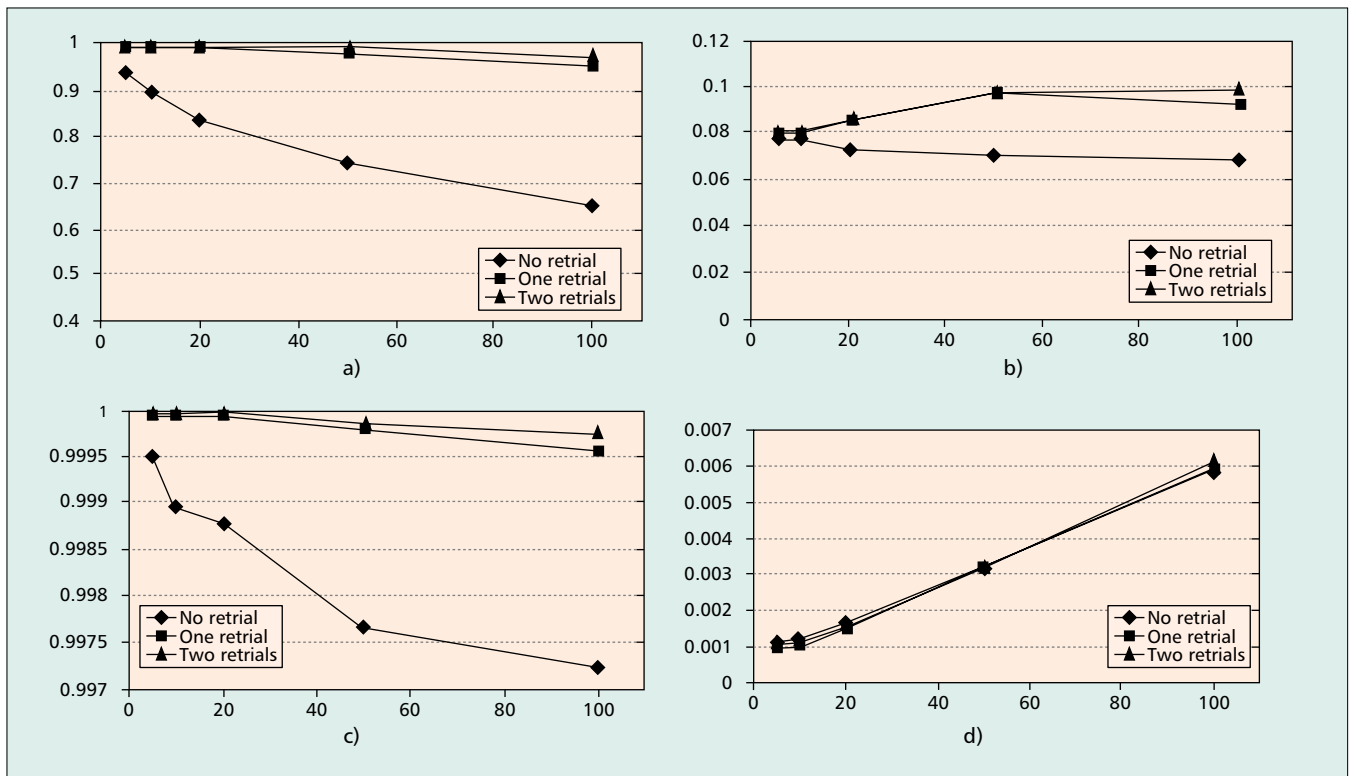


Figure 4. Performance comparison for enhanced path restoration with different retrial opportunities ($f = 10$): a) restoration success rate vs. load (in Erlangs); b) restoration time vs. load (in Erlangs); c) availability vs. load (in Erlangs); d) blocking probability vs. load (in Erlangs).

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 the contention among restoration requests. Note that when the source node retries the restoration, it can employ the latest link state information, thus reducing the chances of contention. We plot the performance metrics of the path-restoration method vs. the number of retrials in Fig. 4a–d. The no-retrial curves are repeated from 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, but an increase in this delay increases the restoration time (not shown).

With just one retrial opportunity, the restoration success rate is found to jump close to 100 percent for most cases (Fig. 4a). Even in the extreme case (network load equal to 100 Erlangs), the restoration success rate is still 96 percent. Increasing the number of retrials to two will further increase the restoration success rate at 100 Erlang load to about 98 percent. Additional experiments show that increasing the number of retrials to numbers larger than two will hardly yield any 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. 4b). 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.

As a result of effectively resolving contentions, two retrials can raise the availability of enhanced path restoration to higher than 99.99 percent for network load up to 50 Erlangs. Even for a network load of 100 Erlangs, the availability is still 99.98 percent. One point worth mentioning is how the availability result should be interpreted. When the restoration success rate is not very close to 1 (< 99.5 percent with the parameter settings in our experiments), the unavailable time is mainly contributed by the un-restored failures. Since the average un-restored time changes with the average call holding time, the availability is not sensitive to changes in the average call holding time. However, when the restoration success rate is so high that the restoration time (which does not change with the average call holding time) is the dominant part of the unavailable time, longer average call holding time will result in higher availability. Although none of the network-availability figures shown here meets some industrial requirements (e.g., five 9s or 99.999 percent), recall that our average call holding time is only 50 s, so this availability can be improved by using higher (i.e., closer to reality) call holding time. For example, if we use an average call holding time of 500 s, for a network load up to 50 Erlangs, number of retrials equal to 2, and $f = 10$, the availability resulting from enhanced path restoration will be above 99.999 percent. Also, the failure frequency

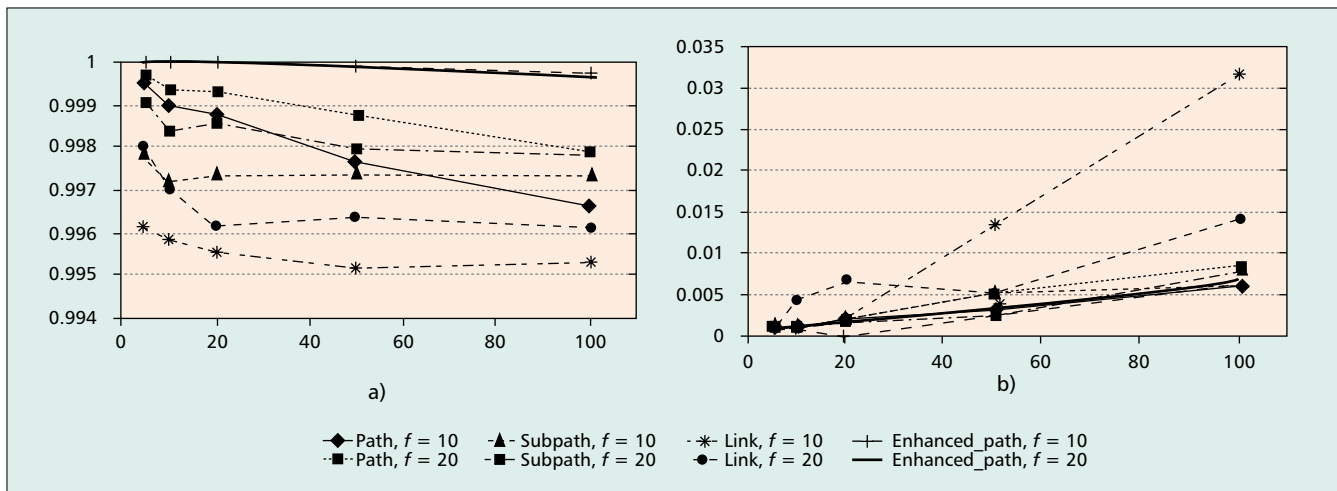


Figure 5. Performance comparison for different restoration techniques under different failure rates: a) availability vs. load (in Erlangs); b) blocking probability vs. load (in Erlangs).

chosen here is quite high, and more discussion on this parameter is provided in the following subsection. For more realistic failure frequencies, our restoration schemes are expected to achieve much higher availability values, possibly five 9s or higher. Audiences should adjust this average call holding time and failure arrival time according to the characteristics of their own networks when considering high restoration success rate cases.

Retrials do not seem to have any significant effect on blocking probability.

THE EFFECTS OF FAULT INTERARRIVAL TIME

Failure frequency directly affects network availability, independent of whether the restoration success rate is high or low. Notice that the $f = 10$ value used in our experiments corresponds to a failure rate much more frequent than 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 time. Longer failure interval (relative to average call holding time) means more call setup and teardown between two failures (on average, two failures can happen simultaneously). Unable to simulate for more realistic f values, we choose to illustrate our results by trying $f = 20$.

In Fig. 5a and b, we superimpose availability and blocking probability performance of a lower failure rate experiment ($f = 20$) to the $f = 10$ results obtained earlier. The restoration success rate and restoration time comparisons are not shown here because our experiments indicate that these two parameters are not affected much by the failure arrival rate.

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. Our experiments also indicate that the blocking probability under link restoration is clearly affected by the value of f , while other blocking probability curves are not affected much. The blocking probability for link restora-

tion decreases with decrease of failure rate because the local congestion caused by link restoration happens less frequently.

CONCLUSION

In this study, we employed the ns-2 simulation platform to implement three restoration techniques (path, subpath, 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. In our simulation experiments, all restoration techniques can deliver availability higher than 99.5 percent and restoration time equal to a few tens of milliseconds under very frequent failures ($f = 10$) in a nationwide (NSFNET) 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 subpath and link restoration can provide restoration speed faster than path restoration; however, they have a 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.

ACKNOWLEDGMENT

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, Jan. 2001, pp. 144–51.
- [2] P. Ashwood-Smith *et al.*, "Generalized Multi-protocol Label Switching (GMPLS) Architecture," work in progress, Internet draft, draft-ietf-ccamp-gmpls-architecture-01.txt, Nov. 2001.
- [3] S. Ramamurthy and B. Mukherjee, "Survivable WDM Mesh Networks: Part I. Protection," *Proc., IEEE INFOCOM 1999*, vol. 2, Mar. 1999, pp. 744–51, "Part II. Restoration," *Proc., IEEE ICC '99*, vol. 3, June 1999, pp. 2023–30.
- [4] O. Gerstel and R. Ramaswami, "Optical Layer Survivability—an Implementation Perspective," *IEEE JSAC*, vol. 18, no. 10, Oct. 2000, pp. 1885–99.
- [5] J. P. Lang *et al.*, "Generalized MPLS Recovery Mechanisms," work in progress, Internet draft, draft-lang-ccamp-recovery-01.txt, July 2001.
- [6] G. Sahin, S. Subramaniam, and M. Azizoglu, "Signaling and Capacity Assignment for Mesh-based Restoration Schemes in Optical Networks," *J. Op. Net.*, vol. 1, no. 5, May, 2002, pp 188–206.
- [7] G. Li *et al.*, "Experiments in Fast Restoration Using GMPLS in Optical/electronic Mesh Networks," *Proc. OFC 2001*, vol. 4, Mar. 2001, pp PD34: 1–3.
- [8] V. Anand, S. Chauhan, and C. Qiao, "Subpath Protection: A New Framework for Optical Layer Survivability and its Quantitative Evaluation," Dept. Comp. Science Eng., SUNY Buffalo, Tech. rep. 2002-01, Jan. 2002.
- [9] C. Ou, H. Zang, and B. Mukherjee, "Sub-path Protection for Scalability and Fast Recovery in WDM Mesh Networks," *Proc. OFC 2002*, Mar. 2002.
- [10] P. Ashwood-Smith *et al.*, "Generalized MPLS Signaling — CR-LDP Extensions," work in progress, Internet draft, draft-ietf-mpls-generalized-cr-ldp-06.txt, Apr. 2002.
- [11] K. Kompella *et al.*, "OSPF Extensions in Support of Generalized MPLS," Internet draft, draft-ietf-ccamp-ospf-gmpls-extensions-07.txt, May 2002, work in progress.
- [12] Y. Ohba *et al.*, "MPLS Loop Prevention Mechanism," RFC 3063, Feb. 2001.

BIOGRAPHIES

JIAN WANG (jnwang@ucdavis.edu) graduated from the University of California, Davis with a Ph.D. in engineering applied science. His major at UC Davis was optical networking. He received an M.S. degree in biomedical engineering from Shandong Medical University, P.R. China, and another M.S. degree in engineering applied science from UC Davis. He received his B.S. degree in physics from Shandong University, P.R. China. Currently, he is an assistant professor at Florida International University.

LAXMAN SAHASRABUDDHE received his B.Tech. degree from the Indian Institute of Technology, Kanpur, in 1992, an M.Tech. degree from the Indian Institute of Technology, Madras, in 1994, and a Ph.D. degree from the University of California, Davis (USA) in 1999. He is the recipient of the Best Doctoral Dissertation Award from the College of Engineering, UC Davis, for his research on WDM optical networks. From 1999 to 2000 he was an embedded software engineer at Amber Networks, which was acquired by Nokia in July 2001. Currently, he is a principal member of technical staff at SBC Services, Inc.

BISWANATH MUKHERJEE received his B.Tech. (Hons.) degree from Indian Institute of Technology, Kharagpur (India) in 1980 and a Ph.D. degree from the University of Washington, Seattle, in June 1987. At Washington, he held a GTE Teaching Fellowship and a General Electric Foundation Fellowship. In July 1987 he joined UC Davis, where he has been professor of computer science since July 1995, and chairman of computer science since September 1997. He is co-winner of paper awards presented at the 1991 and 1994 National Computer Security Conferences. He serves on the editorial boards of *IEEE/ACM Transactions on Networking*, *IEEE Network*, *ACM/Baltzer Wireless Information Networks*, *Journal of High-Speed Networks*, *Photonic Network Communications*, and *Optical Network Magazine*. He also served as editor-at-large for optical networking and communications for the IEEE Communications Society. He served as the technical program chair of the IEEE INFOCOM '96 conference. He is author of the textbook *Optical Communication Networks* (McGraw-Hill, 1997), which received the Association of American Publishers, Inc.'s 1997 Honorable Mention in Computer Science. His research interests include lightwave networks, network security, and wireless networks.

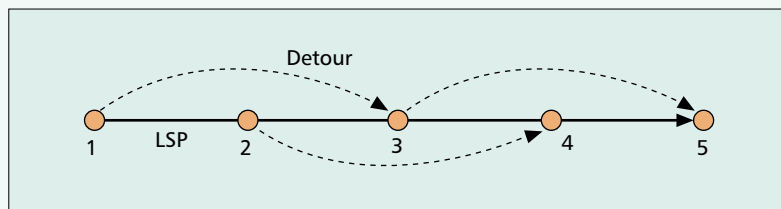
In general, our studies show that the enhanced path restoration technique with an additional retrial for path restoration after a failure can achieve very good network availability, for the price of a little higher restoration time.

APPENDIX

An Internet draft by D. Gan, which has been implemented by some router vendors in their products, proposed a hybrid fault management mechanism. During the setup phase of an LSP, no backup resource is reserved. Once the LSP setup process is done, the source node and the intermediate nodes try to reserve "detours" toward some downstream nodes. If an intermediate node cannot find a detour temporarily, the node will silently retry periodically. A typical path protected by detours is illustrated in Fig. 6. An attractive feature of this mechanism is that it can protect against failure scenarios such as multiple fiber failures on the original LSP.

Because backup resources are prereserved (if detours do exist), this detouring mechanism shares some commonality with typical protection techniques, such as fast recovery

speed and high resource consumption, which are especially true in GMPLS networks. Detouring is also similar to restoration techniques in such a way that the existence of a detour is not guaranteed. Some extensions to MPLS signaling (i.e., new signal objects) are needed to support this mechanism.



■ **Figure 6.** Typical backup resources reserved by detouring (a fault management technique proposed in by D. Gan).