Resilience in GMPLS Path Management: Model and Mechanism

Jong T. Park, Kyungpook National University

ABSTRACT

The recent advent of converging the IP and optical networks has necessitated the development of a generalized multiprotocol label switching framework. Resilience becomes more important than ever before in a GMPLS network since a single cut of an optical fiber may generate hundreds of link and node failures at high layers of the GMPLS architecture. In this article we briefly survey the current work regarding GMPLS recovery management, and present a new resilience-based dynamic GMPLS path management strategy. We present a simple model to represent the resilience requirements in GMPLS path management, and propose fast path management algorithms. The salient feature of the proposed approach is that it enables the paths to be dynamically selected under multiple simultaneous failure occurrences while satisfying the resilience requirement. Backup path design rules are developed, and the condition for backup path availability is derived for the special mesh-type GMPLS network. Finally, a simple example is shown to illustrate the effectiveness of the proposed resilience model and path management mechanism.

INTRODUCTION

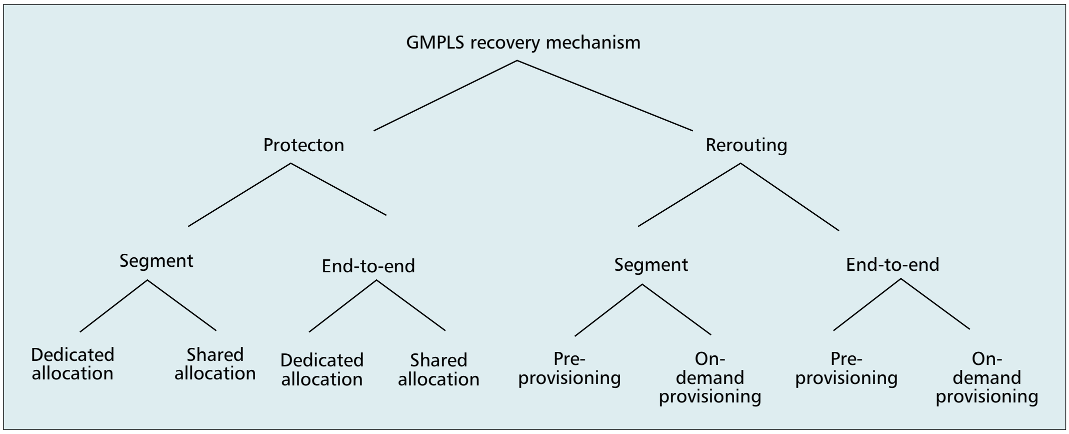
The recent advent of converging the IP and optical networks has necessitated the development of generalized multiprotocol label switching (MPLS). Generalized MPLS (GMPLS) differs from MPLS in that it supports multiple types of switching such as time-division multiplexing (TDM), lambda, and fiber (port) switching [1]. The functions of the control plane of MPLS are extended and modified to support for these additional switching types. These extensions and modifications in turn impact the basic properties of the label switched path (LSP) of MPLS with respect to label setup, traffic direction, error propagation, and end node synchronization. GMPLS utilizes IP-based control (i.e., signaling and routing) protocols to provide and maintain network connections.

The concept of resilience will become more important than ever in the GMPLS network since a single cut of a dense wavelength-division multiplexing (DWDM) optical fiber may generate hundreds of link and node failures at high layers of the GMPLS architecture, which may in turn incur a tremendous amount of significant data loss if the failures are not recovered properly. Resilience is a capability of recovery from network component failures. In GMPLS, the recovery methodology would generally consist of the following steps: failure detection, failure localization and isolation, failure notification, recovery (protection and/or restoration), and reversion (returning the traffic to the original working LSP or to a new one).

The Internet Engineering Task Force (IETF) is currently actively working on standardization of a GMPLS recovery mechanism [1–4], which can be roughly classified into protection and rerouting. In protection switching, often called fast reroute, alternate LSPs are preprovisioned to minimize disruption of service if the network component fails. In the rerouting scheme, recovery LSPs are established after failures occur. Both mechanisms have some advantages and disadvantages with respect to recovery time, resource usage, and survivability. Both constraint-based routing and signaling methods are utilized to establish the recovery path. We analyze the features of these mechanisms subsequently. Recovery management should be fast enough to not affect delay-sensitive applications, and efficient regarding resource utilization.

Li *et al.* [5] recently showed that the performance of the GMPLS restoration technique for a mesh-type optical network was fast enough to be competitive with synchronous optical network (SONET) ring recovery time. Ho and Mouftah [6] presented a framework for a shared protection mechanism in WDM mesh networks. Li *et al.* [7] proposed a path selection algorithm for the restoration of LSPs over a shared bandwidth in a fully distributed GMPLS architecture. Autenrieth and Kirstadter [8] presented a scheme of integrating IP resilience with the traditional QoS requirement in MPLS for a single failure. Four resilient classes were defined to differentiate the various recovery mechanisms. Although it may be one of the first attempts to employ the resilience requirement of MPLS path management, it does not show how to specifically design the recovery path for a given resilience requirement.

0163-6804/04/$20.00 © 2004 IEEE



■ **Figure 1.** *Classification of the GMPLS recovery mechanisms.*

In a restoration mechanism, the backup path is established after failures in a primary path occur, and then the data traffic is switched to the backup path. Both mechanisms are different in terms of time scale and required resources.

In this article we briefly survey current work on GMPLS recovery management, and investigate specific features of various standard approaches. We present a simple formal model to represent the resilience requirement for GMPLS path management, and propose a fast path selection mechanism. The salient feature of the proposed mechanism is that it enables the paths to be dynamically selected *under multiple failure occurrences*, while satisfying the given resilience requirements. Most of the previous work regarding optical network recovery focused on the management of a single link failure.

Rules are developed for efficient design of available backup paths. Using these backup path design rules, the condition for the existence of backup paths under multiple simultaneous link failures has been developed for the special mesh-type GMPLS network where every node in the path has the same degree. This condition empowers the proposed dynamic path management mechanism to be fast enough to find optimal backup paths that satisfy the resilience constraints. Illustrative examples are drawn to show the operation of the approach. Finally, a simple example is shown to illustrate the effectiveness of the proposed resilience model and path management mechanism. The approach is generic enough to be applicable consistently to resilient path management in various layers of GMPLS, ranging from higher GMPLS path management to the lowest optical path management. It can facilitate automatic service provisioning in a future optical network employing the GMPLS framework.

GMPLS Recovery Mechanisms

There has been much research on the GMPLS recovery mechanism in the IETF [2]. A primary path is the main traffic route between the end nodes in a GMPLS network; it is sometimes referred to as a *working path* in standards documents. A backup path, sometimes called an *alternate path*, is defined as an alternative path for when the primary path is unavailable due to failures of links or nodes within it. A component of the GMPLS network implies either a node or a link. The GMPLS recovery mechanism, like the MPLS recovery mechanism [9], is roughly classified into two techniques: protection switching and restoration/rerouting. In protection switching, a set of backup links and/or nodes is always preselected, and the required resources are also preallocated. Therefore, when a failure occurs, the primary path can be switched immediately to the backup path. In a restoration mechanism, the backup path is established after failures in a primary path occur; then the data traffic is switched to the backup path. Both mechanisms are different in terms of timescale and required resources.

In Fig. 1 we classify the various GMPLS recovery mechanisms in terms of protection scope and resource usage. Both protection and rerouting mechanisms can be categorized into local link or segment protection (or rerouting), or end-to-end protection (or rerouting) based on the scope of the recovery domain. Protection switching can be further classified in several ways: 1+1 and 1:1, 1:*N*, *M*:*N*, and split path protection [9]. The 1:1 approach allows traffic to be transmitted on a separate backup path if the primary path fails. Protection LSP can be established at either the end or intermediate nodes. In the 1+1 approach, traffic is transmitted simultaneously on two links, and the best traffic is selected at the receiving node. In *M*:*N*, *M* backup paths are pre-established to protect *N* primary paths, and the data traffic is switched to the backup paths only after the primary paths fail.

Both the 1:1 and 1:*N* protection schemes can be regarded as special cases of the *M*:*N* approach. In split path protection, multiple recovery paths are allowed to carry the traffic of a primary path based on a certain load splitting ratio. In both the 1+1 and 1:1 approaches, the backup path is dedicated to the primary path. In both the 1:*N* and *M*:*N* approaches, the backup paths may be shared among the primary paths. In *M*:*N*, the high reserved resource of the *M*:*N* (or 1:1) approach may be used by other low-priority traffic, called *extra traffic* in the IETF standard [10]. The restoration mechanism is sometimes called a rerouting mechanism, and usually adopts the make-before-break principle. The make-before-break concept allows an old path to still be used during the setup of a new path to avoid double resource reservation. After the setup process is completed, the node performing rerouting may switch to the new path and close the old path. This feature is supported

Recovery mechanism	Recovery domain/resource	Recovery time	Resilience to multiple failures	Resource consumption
Protection	Local (dedicated/shared)	Low	Low	High
	End-to-end (dedicated/shared)	Low	Low	High
Restoration (rerouting)	Local/preprovisioned	Medium	Medium	Medium
	Local/on-demand provisioned	High	High	Low
	End-to-end/preprovisioned	Medium	Medium	Medium
	End-to-end/on-demand provisioned	High	High	Low

Table 1. *A comparison of various GMPLS recovery mechanisms.*

A hybrid method, where the backup paths are planned in advance, but without resources being allocated, could be a good compromise as a fast and efficient mechanism for GMPLS recovery.

with Resource Reservation Protocol with Traffic Engineering (RSVP-TE) using shared explicit filters and Constraint-Based Routing Label Distribution Protocol (CR-LDP) using the action indicator flag [1].

In rerouting, backup path (or subpath or link) construction consists of three phases: precomputing for path selection, signaling, and resource reservation [3]. Preprovisioning implies that any one or combinations of these phases are provisioned in advance. For example, one case would be that only path selection is done, but neither signaling nor bandwidth is allocated. In the other case, all three phases may be performed in advance. On-demand provisioning implies that any of the three phases is only performed after failure occurs and not in advance. In addition to these criteria, we can also classify the mechanism based on the hierarchy of the GMPLS control plane. GMPLS also defines a protection mechanism called *enhance protection* in which protection rings or other methods based on a pre-established topology of protection resources are used to enhance the level of protection, possibly for multiple link failures within a span [1].

In Table 1 we summarize the characteristics of various mechanisms with respect to recovery time, resilience to multiple failures, and resource consumption. Here, resilience to multiple failures is closely related to the survivability of the network. When multiple failures occur unpredictably, they may affect both the primary and backup paths together, even though both paths belong to different shared risk link groups (SRLGs). The protection scheme may not be able to recover the path in this case, but the rerouting scheme with on-demand provisioning may be successful. In general, both resilience and recovery time increase with less preplanning and resource reservation, so there is usually a trade-off between resilience and recovery time with respect to the amount of preplanning and resource reservation/allocation.

The protection mechanism is generally found to be effective in coping with local link failures at lower layers of the GMPLS hierarchy. Rerouting is effective in providing survivability for best effort IP traffic. The protection mechanism, even though it is fast, is weak in handling multiple failures, and generally needs spare resources. The protection mechanism is not robust when both the primary and backup paths fail. Rerouting is generally robust to multiple failures and resource-stringent since it tries to find a new path after failures occur, as in the conventional IP routing protocol. However, rerouting is usually slow. A hybrid method, where the backup paths are planned in advance but without resources allocated, could be a good compromise as a fast efficient mechanism for GMPLS recovery. In this article we propose an efficient hybrid recovery mechanism that could handle multiple failure occurrences effectively.

A Resilience Model for GMPLS Path Management

In GMPLS, as in MPLS traffic engineering requirements [11], a basic resilience attribute determines whether the failed LSP is to be rerouted when segments of its path fail. Extended resilience attributes are used to specify specific recovery mechanisms and policies that govern the relative preference of each specified backup path. The user service level with regard to reliability specified in service level agreements (SLAs) between customers and service providers may be mapped to these resilience attributes of the LSPs [1]. We present a simple model to represent this resilience attribute for the GMPLS recovery mechanism below.

Definition: A path resilience in the GMPLS network is defined as a real-valued function such that

$$\text{Path Resilience} = \sum_{ProtectionSet} \frac{1}{m} \cdot \frac{\text{Number of Protected Components}}{\text{Total Number of Components}},$$

where m is the multiplicity factor of a primary path, and *ProtectionSet* denotes the set of all the backup paths and/or segments that protect the primary path. The multiplicity factor m of a path defines the number of total primary paths that share a backup path, or a segment. It is used to represent the sharing of a backup path in the GMPLS recovery mechanism.

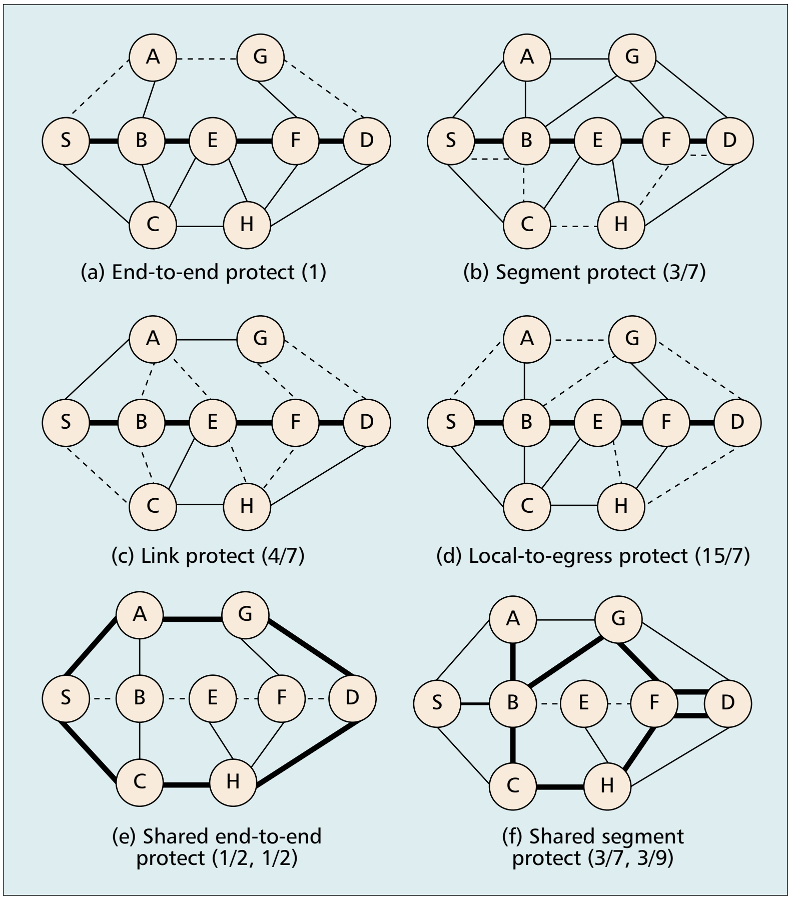
The *Total Number of Components* implies the total number of components in a path, without including the end nodes of the GMPLS network since they are always shared among the primary and backup paths. The *Total Number of Protected Components* implies the total number of components in the path that are protected by backup paths. Path resilience is zero if there is no backup path, or all the components of the backup path are shared with the primary path. For brevity of expression, we simply use resilience instead of path resilience. The resilience model enables service providers or network operators to automatically support a range of different service levels in order to optimize their service revenue with respect to available network resources.

In Fig. 2 we show various protection modes used in both the GMPLS protection and restoration mechanisms. A bold line indicates a primary path, and a dotted line indicates a backup path. In Fig. 2a a separate end-to-end backup path is established to protect the primary path, while in Fig. 2b a segment <B, E, F> of the primary path is protected by the backup path. In Fig. 2b nodes S and D are equivalent to ingress and egress routers in MPLS, and nodes B and F are equivalent to the path switch and path merge label switched routers (LSRs) of MPLS defined in [9], respectively. The path switch LSR is responsible for switching or replicating the traffic between the primary and backup paths, and the path merge LSR is responsible for receiving the backup path traffic, and merging it back onto the primary path.

In Fig. 2c and d we show the protect mode for link protection and local-to-egress protection. In Fig. 2e two primary paths share one backup path, and in Fig. 2f the segments of primary paths are protected by backup paths. In Fig. 2f the backup path for primary path <A,B,G,F,D> is <A,B,E,F,D> and that for primary path <S,B,C,H,F,D> is <S,B,E,F,D>. Segment <B, E, F> is shared by two backup paths. In this case, a bypass tunnel described in [9, 12] can be constructed for segment <B, E, F>. A bypass tunnel is an LSP that could support a number of recovery backup paths using MPLS label stacking. In this case, node B stacks the labels for LSP <A,B,G,F,D> and LSP <S,B,C,H,F,D> by creating and attaching a new label for segment <B,E,F>. Node F, the penultimate node, pops up the attached label to retrieve the original traffic flow.

In Fig. 2a–d the multiplicity factor is 1. For Fig. 2a the path resilience is 1, which is shown in parenthesis since all the components are protected. For Fig. 2b the path resilience is 3/7 since the number of protected components is 3 while the number of total components is 7. For Fig. 2c there exist four alternate paths in *ProtectionSet*, and the resilience is 4/7 since the resilience of each alternate path covering a link is 1/7. For Fig. 2d the resilience is 15/7 since there are overlapping protect paths. For Fig. 2e and f the multiplicity factor is 2 because the backup path is shared by two primary paths. For Fig. 2e the resilience of each primary path is 1/2, and for Fig. 2f the resilience is 3/9 for path <S, B, C, H, E, D> and 3/7 for path <A, B, G, F, D>.

The recovery mechanism in GMPLS should be fast, able to handle multiple failures, and efficient in resource utilization. All of these factors are contingent on what percentage of the path is protected. The resilience model, defined to be *the normalized ratio of the fraction of the protection region of a path*, can effectively represent various protective modes used in both the protection and restoration mechanisms in GMPLS. The fraction is normalized to the number of alternative backup paths. The resilience requirement specified in the SLAs can be mapped to a resilience value when the specific protection mode applied to the resilience model is determined.



(a) End-to-end protect (1)
(b) Segment protect (3/7)
(c) Link protect (4/7)
(d) Local-to-egress protect (15/7)
(e) Shared end-to-end protect (1/2, 1/2)
(f) Shared segment protect (3/7, 3/9)

Figure 2. *Various protection modes.*

Backup Path Design Rules and Availability Testing Condition

Backup Path Design Rules

In this article the assumptions made for GMPLS path management are as follows:

- Each link of an LSP is bidirectional, and one logical link exists between the two adjacent nodes.
- Multiple failures can be detected by lower-layer mechanisms such as loss of light. Failures can be localized, and the notification message can be sent rapidly to the target GMPLS end nodes using either GMPLS LMP [12] or RSVP-TE extensions. Explicit source routing is preferred for path setup, and rerouting of the LSP may use the techniques employed in MPLS-TE.
- Multiple component failures will occur unpredictably, affecting both primary and backup paths simultaneously, even though they do not share a fate (i.e., belong to disjoint SRLGs).

The construction of the backup paths needs to provide feasible backup paths which satisfy the given resilience constraints. The bandwidth resource for the selected backup paths may be reserved but not allocated.

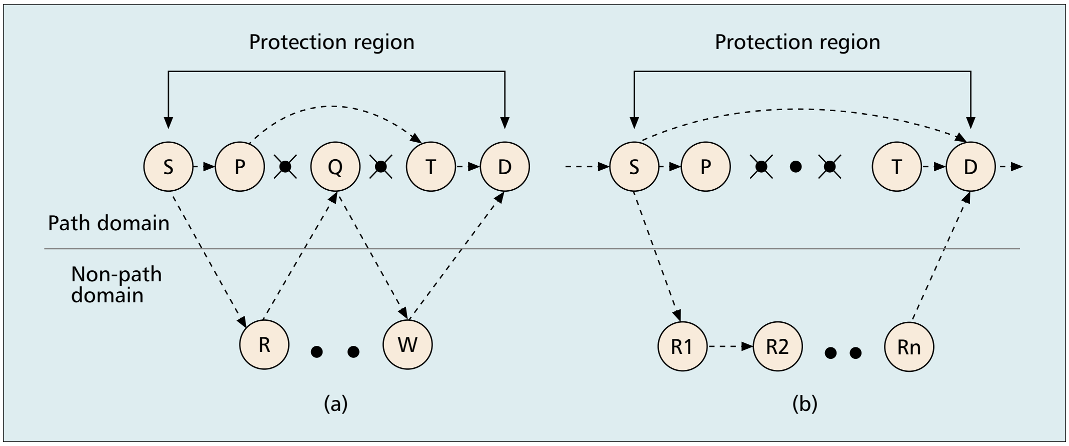


Figure 3. *Feasible and infeasible candidates for backup paths: a) a case violating rule 3; b) feasible backup path candidates.*

The problem of path selection is stated as follows: For a given GMPLS network consisting of a set of nodes, links, and resilience constraints, establish and maintain the primary and backup paths such that disruption of service is minimized under multiple component failures while satisfying the resilience constraints. The construction of backup paths needs to provide feasible backup paths that satisfy given resilience constraints. The bandwidth resource for the selected backup paths may be reserved but not allocated. We present guidelines for design of backup paths below.

- **Rule 1:** No node in the path should be trespassed more than once.
- **Rule 2:** The backup paths for a given primary path should follow the same sequence of nodes and links of the primary path in the subpaths that are not protected.
- **Rule 3:** No component in the protection region of a primary path may be used for the construction of backup paths, except for the beginning and end nodes of the protected segment.

Rule 1 implies that there should be no cycles in the path, which is usual practice in network design. Rule 2 is applied only for the design of backup paths. The motivation of rule 2 is to minimize resource utilization for backup path construction, so both primary and backup paths use the same nodes and links in the shared region. Rule 3 implies that nodes and links in the primary path should not be used for the generation of backup path candidates.

In Fig. 3 the *protection region* of a primary path is defined as a segment of the primary path protected by the backup paths. *Path domain* implies the set of nodes in the primary path, and *non-path domain* implies the set of nodes not included in the primary path. Figure 3a shows the violating cases of rule 3 where subpath <S, …, D> of the primary path needs to be protected. Neither <S, R, Q, W, D> nor <S, P, T, D> should be used as a candidate segment for backup path construction since according to rule 3, no link emanating from the intermediate between end nodes S and T should be used for backup path construction. Figure 3b shows two possible candidates for backup path construction: subpaths <S, D> and <S, R1, R2, …, R*n*, D>. No other types may be considered for construction of the backup paths. Path design rules 1, 2, and 3 guide the generation of the candidates for a primary path so that they are used for the construction of backup paths that satisfy the resilience constraints.

Before proceeding, we define the concept of a protection region. A primary path is said to have *k-protection* if any segment of the path consisting of (k – 1) adjacent nodes and k links connecting these nodes is protected by the backup paths. In this case, the primary path is said to have a protection region of length k. For example, 1-protection path says that a link in a path is protected, and 2-protection path says that one node and 2 links incident to the node in the path are protected. In order to design a primary path with k-protection, we need to consider backup paths that are only disjoint with the primary path in a protection region of length k. Given a resilience requirement from customers, a k-protection backup path associated with the requirement can be chosen. In other words, the resilience constraint can determine the length of the protection region (i.e., k-protection).

The Condition for Backup Path Availability

In this subsection we derive a condition for the existence of backup paths that satisfies the resilience constraints in the GMPLS network, by applying the backup path design rules from an earlier section. We present a solution for a GMPLS network of the mesh type where every node in the primary path is assumed to have the same degree (i.e., the same number of links). As explained earlier, the LSPs are assumed to be bidirectional, so the data can be input to or output from the node through the same link. Let **P** and **R** denote the path domain and non-path domain, respectively. Let <P_1, P_2, …, P_n> be a primary path in a GMPLS network with N nodes. The condition for the availability of the backup paths satisfying the resilience constraints is presented below. As described previously, the resilience constraint here specifies k-protection.

```
Procedure Backup_Path_Design (Total Number of Link Failures, Resilience Constraint)
/* Test the availability of the k-protection backup path. If available, construct the backup path with minimal cost. */
Begin
Step 1: Upon receiving the multiple failure notifications, update the GMPLS configuration database;
Step 2: Calculate $\zeta$ for the construction of the k-protection backup path availability;
Step 3: If the Total Number of Multiple Link Failures < $\zeta$
        Then {Find all the feasible backup paths running the algorithm Fast_Backup_Path_Construct;}
Step 4: Select the k-protection backup path with minimal cost from the set of feasible backup paths.
End
```

Algorithm 1. *Procedure Backup_Path_Design.*

In order to achieve fast rerouting, it is necessary to decide rapidly whether the construction of a proper backup path is feasible or not under multiple failure occurrences, while satisfying the resilience requirement.

The Condition for Backup Path Availability — For a mesh-type GMPLS network, let us assume that the nodes in the primary path $<P_1, P_2, \ldots, P_n>$ for the links from path domain **P** to non-path **R** have the same degree, γ, and the subgraph consisting of nodes in **R** is connected without passing through a node in the path domain. Also, there are no direct links between any two nonadjacent nodes in the primary path. Then the primary path with n nodes has *k*-protection even though (ζ – 1) number of links from **P** to **R** fail, where

$$\xi = \begin{cases} mk\gamma & j \le k \\ ((m-1)k + j)\gamma & j > k \end{cases}$$

for $n = 2km + j, j = 0, 1, \ldots, (2k - 1)$. The proof is shown in the Appendix.

In the proof of the above availability condition, the path design rules and bidirectional property of the LSPs of the GMPLS network are applied to the construction of the backup paths. Where the number of component failures in the GMPLS network is large, the proper backup path satisfying the resilience requirement may not be found. Since the general path finding problem with constraint is known to be NP-complete, the search time for finding out the proper backup path that satisfies the resilience constraint would be intolerably large for some failure conditions. In order to achieve fast rerouting, it is necessary to decide rapidly whether the construction of a proper backup path is feasible or not under multiple failure occurrences, while satisfying the resilience requirement. The testing condition for availability of backup paths enables checking whether there would be available backup paths for a given resilience constraint or not. Actually, the number (ζ – 1) gives the maximal number of allowable link failures in the GMPLS network that will guarantee the existence of a primary path with *k*-protection. If the testing condition is satisfied, it is always feasible to find a backup path that satisfies the resilience constraint (i.e., *k*-protection).

An Optimal Backup Path Design Methodology

We present a *k*-protection backup path design methodology for a special mesh-type GMPLS network in which nodes apart from each other with *k*-links in the primary path have the same number of degrees. It consists of four steps: configuration database update, evaluation of a testing condition, feasible backup path construction, and selection of the optimal one with minimal cost. The outline of the backup path design methodology is described in Algorithm 1, and detailed algorithms are presented in Algorithms 2 and 3.

The optimal backup path can be obtained at step 4 of the Procedure Backup_Path_Design by selecting the least cost backup path from the set of *k*-protection backup path candidates. Every end node may run this algorithm independent of other nodes, so the path management for the overall GMPLS network can be done in a distributed way. In Algorithm Fast_Backup_Path_Construct, the beginning and ending nodes of a path or segment are labeled P_x and P_y, respectively, and the guidelines for backup path design, rules 1, 2, and 3, are applied. Fast_Backup_Path_Construct can rapidly find feasible backup paths if the number of component failures is less than ζ. A recursive procedure, Find_Path, searches for feasible backup paths recursively. Details of the operations are explained in the comment. The backup path construction algorithm is basically a breadth first search algorithm with computational complexity of $O(mn)$ where m is the cardinality of the non-path domain and n is the length of the path. Path design rules 1, 2, and 3 are applied in order to find feasible backup paths.

Example — In Fig. 4 we show an example of testing the availability condition. Let us assume that the resilient constraint specifies the construction of a 2-protection backup path for primary path <N1, N3, N5, N7, N9> with end nodes N1 and N9. The multiple link failures of GMPLS LSPs occur as shown in Fig. 4a, indicated by the cross marks. In Fig. 4 there are two independent non-path domains: {N4, N8} and {N2, N6}. No nodes in {N4, N8} can be connected to a node in {N2, N6} without passing through a node in the path domain, and vice versa. Therefore, testing of the availability condition should be done independent to each non-path domain. The total number of ζ for the primary path can be obtained by summing up the ζ values from the two non-path domains.

By applying the test condition in step 2 of Procedure Backup_Path_Design for the construction of 2-protection backup paths, ζ is found to be 2 for each of the non-path domains {N4, N8} and {N2, N6}. Thus, the value of ζ for the 2-protection backup path is 4 for the configuration in Fig. 4, which is obtained by summing up the ζ values of the two non-path domains.

During the rerouting phase of our approach, the alternative backup path can, even under multiple failure occurrences, rapidly be found by using the test condition of backup path availability.

```
Algorithm Fast_Backup_Path_Construct;
/* Find all the feasible backup paths along the primary path, by applying the design rules Rule 1, Rule 2,
and Rule 3. Let <$P_1$, $P_2$, …, $P_n$> be a primary path in a GMPLS network. */
Begin
For i = n Down To 1
    $P_y = P_i$;
   Unmark all the marked nodes;
    Select an unmarked node P where P is directly connected to $P_y$;
    If P = Px Then Return (P, $P_y$); /* direct backup link found between $P_x$ and $P_y$ */
            Do {Select an unmarked node R where R is an element of a non-path domain and R is directly
            connected to Py;
              Mark R;
              Call Find_Path (R, $P_y$);
`            UNTIL (All nodes which are directly connected to $P_y$ are marked);}}}
End
```

Algorithm 2. *Fast_Backup_Path_Construct.*

```
Recursive Procedure Find_Path (X, Y)
/* Find a backup path < $P_x$, $R_1$, $R_2$, …, $R_m$, $P_y$> recursively using a sequence of R nodes in a non-path
domain. */
Begin
    Select an unmarked node Z which is directly connected to X;
    If Z = $P_x$
    Then Return (Z, Y);
    Else {Do
            If Z is an element of a non-path domain
              Then {Mark Z; Call Find_Path (Z, X);}
          UNTIL (All nodes which are directly connected to X are marked);}
End
```

Algorithm 3. *Find_Path.*

Thus, we know that for some scenarios of four concurrent link failures, there may be no way to construct any 2-protection backup path, but any scenario of concurrent 3-link failures that are not in the primary path would always guarantee the construction of a 2-protection backup path.

Figure 4a shows the case with four link failures where there is no way to construct any 2-protection backup path, while Fig. 4b shows the case where a 2-protection backup path <N1, N3, N4, N8, N7, N9> is available, even though three (i.e., ζ –1) links fail concurrently. Consequently, the testing condition for backup availability effectively enables us to know in advance of the existence of any feasible backup path that satisfies the resilience constraint, under multiple failure occurrences. Fast_Backup_Path_Construct quickly finds out the available 2-protection backup path, and in step 4 of Procedure Backup_Path_Design, the candidate with minimal cost is selected if there are more than one 2-protection backup paths available.

The proposed approach is similar to the protection mechanism, but with additional capability to handle multiple failure occurrences under a specified resilience constraint. In other words, when backup paths are damaged by multiple failure occurrences, alternative backup paths are constructed, similar to the rerouting mechanism, while satisfying the resilience constraint. Therefore, the proposed approach is the hybrid one that combines the merits of conventional protection and rerouting mechanisms. During the rerouting phase of our approach, the alternative backup path can, even under multiple failure occurrences, rapidly be found using the test condition of backup path availability.

Conclusion

In this article we briefly compare the characteristics of the various GMPLS recovery mechanisms currently under active research by the IETF. We then present a simple model to represent resilience for GMPLS path management. Based on the resilience model, we propose an efficient dynamic path recovery mechanism that could efficiently handle multiple failure occurrences in the GMPLS framework. A condition to test the availability of backup paths that satisfies the resilience constraint is derived for a specific GMPLS network configuration in which all the nodes in the primary path are assumed to have the same degree. Using the testing condition for backup path availability, the dynamic path recovery mechanism guarantees finding with computational time $O(N2)$ the optimal backup path under multiple failure occurrences, satisfying the resilience constraints. A further research area may include finding the optimality that considers other design factors such as quality of service and security combined with resilience constraints.

References

[1] E. Mannie *et al.*, "Generalized MPLS Architecture," Internet draft, draft-ietf-ccamp-gmpls-architecture-07.txt, May 2003.

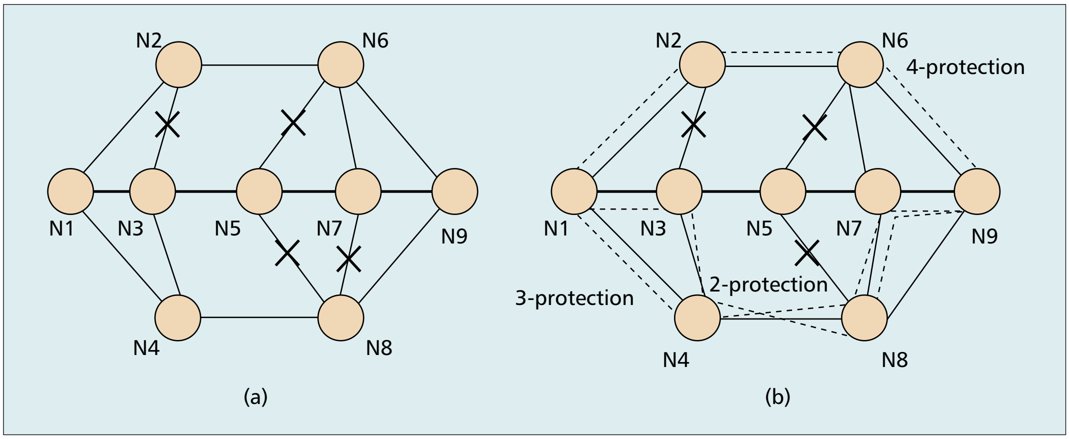


Figure 4. *An example to test the availability condition and the backup path design: a) any 2-protection backup path not available; b) a 2-protection backup path available.*

A condition to test the availability of backup paths that satisfies the resilience constraint is derived for a specific GMPLS network configuration in which all the nodes in the primary path are assumed to have the same degree.

[2] A. Banerjee *et al.*, "Generalized Multiprotocol Label Switching: An Overview of Signaling Enhancements and Recovery Techniques," *IEEE Commun. Mag.*, vol. 39, no. 7, July 2001, pp. 144–51.

[3] D. Papadimitriou and E. Mannie, "Analysis of Generalized MPLS-Based Recovery Mechanisms (Including Protection and Restoration)," draft-ietf-ccamp-gmpls-recovery-analysis-01.txt, May 2003.

[4] J. P. Lang and B. Rajagopalan, "Generalized MPLS Recovery Functional Specification," Internet draft, draft-ietf-ccamp-gmpls-recovery-functional-00.txt, Jan. 2003.

[5] G. Li *et al.*, "Experiments in Fast Restoration Using GMPLS in Optical/Electronic Mesh Networks," *Postdeadline Papers Digest, OFC 2001*, Anaheim, CA, Mar. 2001.

[6] P. Ho and H. T. Mouftah, "A Framework for Service-Guaranteed Shared Protection in WDM Mesh Networks," *IEEE Commun. Mag.*, vol. 40, no. 2, Feb. 2002.

[7] G. Li *et al.*, "Efficient Distributed Path Selection for Shared Restoration Connections," *IEEE INFOCOM 2002*, pp. 140–49.

[8] A. Autenrieth and A. Kirstadter, "Engineering End-to-End IP Resilience Using Resilience-Differentiated QoS," *IEEE Commun. Mag.*, vol. 40, no. 1, Jan. 2002.

[9] V. Sharma *et al.*, "Framework for Multi-Protocol Label Switching (MPLS)-Based Recovery," RFC 3469, Feb. 2003.

[10] L. Berger, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description," RFC 3471, Jan. 2003.

[11] D. Awduche *et al.*, "Requirements for Traffic Engineering Over MPLS" RFC 2702, Sept. 1999.

[12] J. Lang, "Link Management Protocol (LMP)," Internet draft, draft-ietf-ccamp-lmp-08.txt, Mar. 2003.

Biography

JONG T. PARK [SM] (jtpark@ee.knu.ac.kr) is a professor at the School of Electrical Engineering and Computer Science, Kyungpook National University, Korea. He received a Ph.D. degree in computer science and engineering from the University of Michigan and previously worked at AT&T Bell Labs in the United States. He founded the Korean Network Operations and Management Committee (KNOM) of the Korean Institute of Communication Sciences and was one of the founding members of the Asia-Pacific Symposium on Network Operations and Management (APNOMS). He served as chair of the Technical Committee on Information Infrastructure of IEEE Communication Society. He is currently on the editorial board of *International Journal on Network and Systems Management*, and a standing committee member for APNOMS. He was general chair of APNOMS '97, general chair of ICC 2002 Symposium and a co-chair of the GLOBECOM 2002 Symposium on Global Service Portability and Infrastructure. He has also served as a committee member or advisory board member for IEEE/IFIP NOMS and IM. His current research interests include issues related to the management of next-generation information networks including MPLS, GMPLS, B3G/4G, IPv6, and ubiquitous computing and home networks. He is also interested in the development of mobile software and the wireless Internet.

Appendix

The Proof of Condition for Backup Path Availability

Let us first consider the case in which the primary path $<P_1, P_2, \ldots, P_n>$ consists of n nodes where $n = 2km + j$ and j $(2k - 1)$. Let us decompose the set of n nodes into m blocks of size $2k$ nodes and one block of size j nodes, $<P_1, P_2, \ldots, P_{2k}>$, $<P_{2k+1}, P_2, \ldots, P_{4k}>$, …, $<P_{(2k(m-1)}, \ldots, P_{2km}>$, $<P_{2km+1}, \ldots, P_{2km+j}>$. If all the links from **P** to **R** of the latter half of the nodes of the m blocks fail, there is no way, using the path design rules, to construct a k-protection backup path from the m blocks. For the last j nodes, it is also not possible to construct a k-protection backup path since $j \leq (2k - 1)$. In this case, the total number of failures of the links from **P** to **R**, ζ, which would not allow the construction of a k-protection backup path, is $mk\ \gamma$ since the number of link failures in each block is $k\ \gamma$ and there are m blocks. In fact, ζ is a minimal number since each node in the primary path has the same number of links γ from **P** to **R**. Thus, $(\zeta - 1)$ number of link failures from **P** to **R** will guarantee the existence of a k-protection backup path. Next, for the case with $j > k$, there are $j - k$ ways of constructing k-protection backup paths. Therefore, if all the links of the latter half of nodes of each block and the additional j-k nodes from **P** to **R** fail, there is no way to construct a k-protection backup path. Thus, $\zeta = mk\ \gamma + (j - k)\gamma$.