A Rerouting Scheme with Dynamic Control of Restoration Scope for Survivable MPLS Network*

Daniel Won-Kyu Hong[1] and Choong Seon Hong[2]

[1] Operations Support System Lab., KT
62-1 Hwaam-Dong Yuseong-Gu, Daejeon 305-718 KOREA
wkhong@kt.co.kr
[2] School of Electronics and Information, Kyung Hee Univerity
1 Seocheon Giheung Yongin, Gyeonggi 449-701 KOREA
cshong@khu.ac.kr

Abstract. This paper proposes a rerouting scheme that can be applied to the restoration of working Label Switched Paths (LSPs) and pre-provisioned backup LSPs, which consists of two subsequent algorithms. The first algorithm for the dynamic deterimination of the restoration scope (RS)increases the restoration speed by minimizing the complexity of the network topology and by maximizing the reusability of the existing working LSP. The second newly proposed concept of RS extension minimizes the probability of restoration failure by dynamically widening the restoration scope until the RS is equal to the whole network topology. Through simulation, we evaluate the performance of our restoration scheme and the existing protection schemes in terms of the restoration speed, packet loss, network resource utilization, and resource reusability of the existing working LSP.

1 Introduction

The rerouting method for traffic engineering in IP networks became the driving force behind MPLS. The ability to protect traffic against failure or congestion in an LSP can be important in mission-critical MPLS networks [6,7,9]. Restoration is necessary for two different reasons: one is fast restoration and the other is optimized restoration. Fast restoration minimizes service disruptions for the flows affected by an outage using a backup path [1,9]. Optimized restoration serves to alternatively optimized traffic flow in line with a changed network topology [2,9]. However, fast restoration cannot provide fast change of traffic flows any more when the backup and working paths go down simultaneously and cannot increase network resource utilization. In this paper, we propose a LSP rerouting scheme that dynamically aligns the restoration scope in MPLS network. Because this scheme dynamically adjusts the restoration scope depending on the fault and

* This work was supported by University ITRC Project of MIC. Dr. C.S. Hong is the corresponding author.

C. Kim (Ed.): ICOIN 2005, LNCS 3391, pp. 233–243, 2005.

congested location and the overall network status, we may provide the most reasonable bypassing path for avoiding congestion or for restoring fault. This paper proposes a rerouting algorithm to determine the restoration scope taking into account the bandwidth, delay, and hop count. This algorithm can be a scalable one because it expands the restoration scope when it fails to calculate the reasonable bypassing path within the found restoration scope. Our model can expand the restoration scope until the overall network is a restoration scope. Our restoration scheme focuses on the maximization of network resource utilization better than the restoration speed. However, it does not prominently degrade the performance of restoration speed compared with the existing backup path driven approaches [4,5], which establishes the working and backup paths simultaneously for fast restoration and lacks resource utilization and backup path protection schemes. Our model may provide a moderated restoration speed compared with the existing backup path approaches [4,5], maximize network resource utilization and protect the backup path without requiring much longer restoration time. We define a rerouting model that dynamically establishes a transient backup path, taking into account the current network status and topology when the node or link goes down and automatically releases it when the node or links goes up. We propose a algorithm to determine the restoration scope, an algorithm to find reasonable bypassing paths, and demonstrate a procedure that restores the faulty working path without prominent performance degradation while maximizing network resource utilization better than the existing backup path driven restoration methods [4,5]. Through simulation, the performance of the proposed restoration scheme is measured and compared with the existing schemes in terms of packet loss, restoration speed, resource utilization and the reusability of the existing working LSP.

2 The Provision Process of the Alternative LSPs

To meet the requirements of the maximization of network resource utilization and the minimization of the restoration speed, we propose a rerouting model that dynamically provides an alternative path bypassing the fault location with the concept of dynamic RS arrangement. The process of dynamic RS arrangement is composed of two subsequent steps: (1) the generation of the Intermediate Weighted Network Graph (IWNG) from the Network Graph (NG) taking into account the fault location and the traffic metrics, such as requested bandwidth (B_{REQ}), traffic class (T), and hop count (H); and (2) the determination of intermediate ingress node for confine the restoration scope.

Fig. 1 shows the overall process to create an alternative LSP. The detail procedure for the creation of an alternative LSP avoiding the fault location is as follows:

`Preconditions:`

– A network graph, $NG(N, L, B_{AVA}, R)$, where N is a set of node, L is a set of link, B_{AVA} is avariable bandwidth of link, and R is the rerouting option.

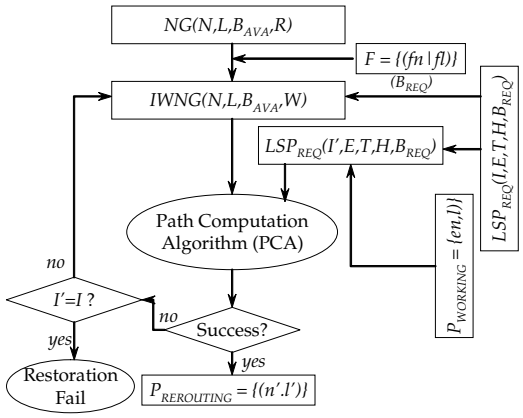


Fig. 1. The overall process to create an alternative LSP

- A list of fault locations, $F = (f_n|f_l)$, which is a set of abnormal nodes (n) or abnormal links (l)
- An LSP request, $LSP_{REQ}(I, E, T, H, B_{REQ})$, where I is the ingress LSR, E is the egress LSR, T is the traffic class (gold, silver, and bronze), H is the hop count that can hopefully be the end-to-end delay if the delay between each link is constant, and B_{REQ} corresponds to the bandwidth requirements.
- A working LSP, $P_{WORKING} = (n, l)$, where n is node and l is link traversing the working LSP

`Procedure:`

[Step 1] First, we create an Intermediate Weighted Network Graph (IWNG) with such information as F, $LSP_{REQ}(I, E, T, H, B_{REQ})$, and $P_{WORKING}$. We will propose the algorithm for creation of IWNG in next section. The weight (W) of IWNG is assigned by the IWNG creation algorithm.

[Step 2] We determine intermediate ingress node (I') to generate an alternative LSP, that is to say, we confine the restoration scope that will be the most reasonable boundary to create an alternative path for the restoration of the occurred faults. In other words, we create $LSP_{REQ}(I', E, T, H, B_{REQ})$. The I' should be one of the nodes in $P_{WORKING}$.

[Step 3] We find the most optimal alternative path from $LSP_{REQ}(I')$ to $LSP_{REQ}(E)$ with the routing constraints such as T, H, and B_{REQ}.

[Step 4] If the path computation algorithm generates an optimal path between $LSP_{REQ}(I')$ and $LSP_{REQ}(E)$, we select it as an alternative path $(P_{REROUTING})$ for the restoration of the fault and stop the procedure.

[Step 5] However, if there is no any reasonable alternative path between $LSP_{REQ}(I')$ and $LSP_{REQ}(E)$, we compare the original ingress node (I) with I'. If I is the same node as I', we stop the procedure because there can be no more wide restoration scope.

[Step 6] If I is not the same node as I', we define I' as an implicit abnormal node ($F \leftarrow LSP_{REQ}(I')$), which results in the extension of restoration scope. As I' is newly added to F, we iterate the above procedure from Step 1 until we find an optimal alternative path for restoration (Step 4) or there is no alternative path (Step 5).

`Output:`

An optimal alternative path, $P_{REROUTING} = (n', l')$.

2.1 A Dynamic RS Determination

The purpose of our restoration model is to achieve fast restoration and high resource utilization. However, fast restoration comes into conflict with high resource utilization. Therefore, this paper proposes a leverage algorithm that reconciles these two factors. In our restoration algorithm, we narrow down the restoration scope as soon as possible for the rapid path computation. However, the wide restoration scope is better than the narrow restoration scope in terms of resource utilization.

To determine the reasonable restoration scope based on the fault location, this paper proposes an algorithm for dynamic RS arrangement. This algorithm generates the WNG for the working LSP provision and the IWNG for the alternative LSP provision.

`Preconditions or Definitions:`

- A network graph, $NG(N, L)$, which is composed of nodes, $n \in N$, and links, $l \in L$.
- A node, $n(w, d, v)$, where w is a weight, d is an accumulated delay, and v is a visiting flag.
- A link, $l(w, r, d)$, where w is a weight, r represents reachability (*yes* or *no*), and d is a delay.
- n_{active} represents the active node allocating the proper weights to all of its neighbor nodes and links connected to it.
- $n_{passive}$ represents the passive node order which is assigned by an active node, that is to say, $n_{passive}$ is a neighbor node of n_{active}.

`Algorithm:`

Refer to Fig. 2.

`Output:`

An Intermediate weighted network graph($IWNG(N, L, B_{AVA}, W)$), where N is a set of node, E is a set of link, B_{AVA} is the avaiable bandwidth of the link, and W is the assigned weight on a link.

```
1.  For each n ∈ N do
2.      n(w,v,d) ← (∞,no,0);
3.  For each l ∈ L do
4.      if (l(r)==no and (l(r)==yes and l(B_AVA) < B_REQ))) trims l from NG ;
5.      if (l(r)==yes and l(B_AVA) < B_REQ)) l(w) ← (∞);
6.  Define the ingress node (LSP(I)) as the initial active node (n_active);
7.  Define the egress node (LSP(D)) as the destination node (n_dest);
8.  n_active(w,d) ← (0,0);
9.  Procedure Allignment(n_active, n_dest)
10.     if(n_active == n_dest) return;
11.     n_active(v) ← yes;
12.     n_passive ← ladj;
13.     for each l connected to n_active do
14.         if( l(w) > n_active(w)+1)
15.             l(w) ← (n_active(w)+1);
16.         if( n_passive(v) == no)
17.             l(d) ← l(d)+n_active(d);
18.         if(n_passive(v) == yes) continue;
19.         if(n_passive(w) > l(w)) {
20.             n_passive(w,d) ← (l(w),l(d));
21.             Allignment(n_passive, n_dest);
22.         }
23.     end of for
24. end of procedure
```

Fig. 2. An algorithm for determiantion of restoration scope

Starting at the ingress node, our algorithm visits all the neighbors, then visits all the neighbors of these neighbors, and so on, until there are no neighbors left to visit. Our algorithm is very simple but there are some rules for assigning the appropriate weight of each node and link with the following steps:

[Step 1] We initialize the nodes of the network using infinity (∞), zero (0) and *no* of visit flag to weight, delay and visiting flag, respectively.

[Step 2] We trim the unfavorable links from NG. If the reachability of link is no or the reachability is yes but its available bandwidth is less than the requested bandwidth, we trim the link from NG. If the link is favorable, we assign zero to the weight of the link.

[Step 3] We define the ingress node ($LSP(I)$ representing the ingress node terminating the LSP) as a first active node (n_{active}) and define the egress node ($LSP(D)$ representing the destination node terminatiing the LSP) as a destination node (n_{dest}).

[Step 4] Initially, we assign zero to the weight and delay of $n_{active}(n_{active}(w,d) \leftarrow (0,0);)$. From now on, we traverse NG until there are no nodes to visit, calling the procedure of Alignment(n_{active}, n_{dest}).

[Step 5] We assign yes to the visiting flag of n_{active} ($n_{active}(v) \leftarrow yes;$) in order to avoid the duplicated traverse of NG.

[Step 6] We determine the weight and the accumulated delay of each links ($l(w,d)$) connected to the $n_{active}$with the following rule:

$$l(w) = \begin{cases} n_{active}(w)+1, & if(l,w) > n_{active}+1 \ or \ l(w) == 0 \\ l(w), & if(l(w) \leq n_{active}+1 \ and \ l(w) \neq \infty \end{cases} \quad (1)$$

$$l(d) = \begin{cases} n_{active}(d)+l(d), & if(n_{passive}(v) == no) \\ l(d), & if(n_{passive} == yes) \end{cases} \quad (2)$$

[Step 7] If the visiting flag of $n_{passive}$is yes, then we traverse another link that is not traversed. If $n_{passive}$is no, we assign the weight and accumulated delay of $n_{passive}$ with the following rule:

$$n_{passive}(w) = \begin{cases} l(w),\ if(n_{passive}(w) == \infty\ or\ n_{passive}(w) > l(w)\ or\ n_{passive}(v) == no); \\ n_{passive}(w),\ \ if(n_{passive}(w) \leq n_{passive}(v) == no) \end{cases} \quad (3)$$

$$n_{passive}(d) = \begin{cases} l(d),\ if((n_{passive}(d) == 0\ or\ n_{passive}(v) == no\ and\ n_{passive}(d) > l(d); \\ n_{passive}(d),\ \ if(n_{passive}(d) \leq l(d)); \end{cases} \quad (4)$$

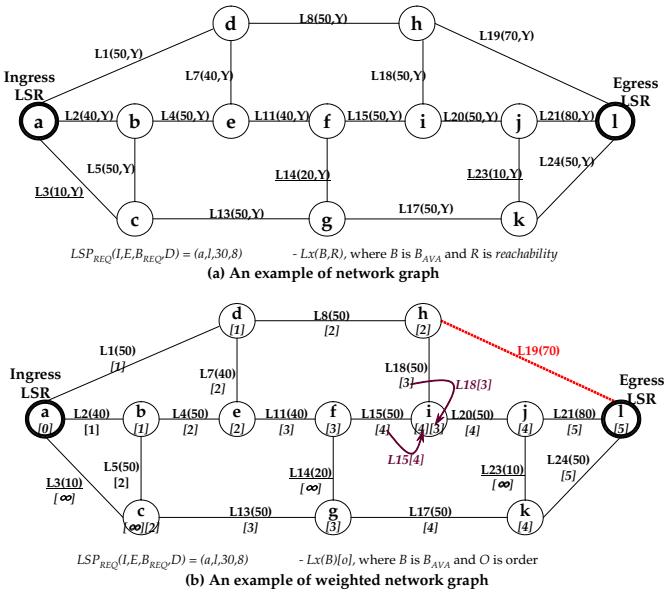


Fig. 3. An example of network graph (NG) and weighted network graph (WNG)

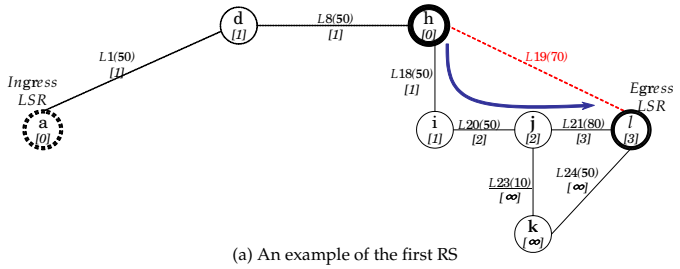
Let's assume that the working path, $LSP_{WORKING}$, traverses *(a-L1-d-L8-h-L19-l)* and there is a fault at *L19*. Fig. 3 (a) shows an NG, where link *L9* is faulty, $LSP(I, E, B_{REQ}, D)$ is *(a,l,30,8)*, B_{AVAs} of *L3*, *L14* and *L23* are less than B_{REQ}. We trim the unfavorable links of *L3*, *L14*, and *L23* and traverse NG from the ingress node of *l* until all visiting flags of nodes can be *yes* in accordance with the rules described in Step 6 and Step 7. As a result of NG traversing, we can generate WNG as shown in Fig. 3 (b).

After generating the WNG, we determine the reasonable RS. At first, we determine I', which can be the node connected to the abnormal link along the reverse traffic flow. In the case of fault at *L19* as shown in Fig. 3 (a), the first candidate I' shall be h because nodes of *h* and *l* are connected to abnormal link,

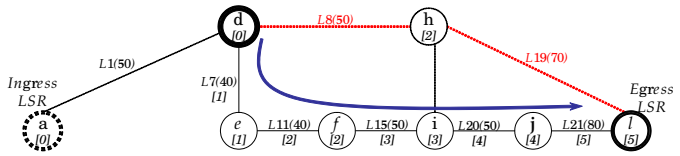
L19. But h is in the reverse traffic flow of the working LSP. So, we select h as I'. Once the I' is selected, we define the restoration scope as the set of nodes and links whose weights are greater than $I'(w)$ and the I' itself as following rule:

$$RS = (n(w), l(w)) \cup (I'), \quad where\ w \supset I'(w) \tag{5}$$

For example, the first RS to restore the link fault (*L19*) can be *(h, L18, i, L20, j, L21, l)* according to the rule for the determination of RS as shown in Fig. 3 (a). Thus, we find an alternative path avoiding the abnormal link of *L19* between h and l. The alternative path can be *(h-L18-i-L20-j-L21-l)*. The algorithm for selecting an alternative path will be given in the next section.



(a) An example of the first RS



(b) An example of the second RS

Fig. 4. An example of the intermediate weighted network graph (IWNG)

If we failed to find an alternative path avoiding the abnormal link of *L19* within the first RS for some reason or another, we need to rearrange the RS, which is the concept of the extension of the restoration scope. Rules for extension of the restoration scope are as follows:

- We suppose that there is an abnormality at I'.
- We propagate the abnormality of I' to its connected links.
- We select one link among the links connected to I', which is a part of the working LSP.

Having selected the link for the extension of RS, we determine the restoration scope with the same rule of equation (5)applied to the determination of the first RS. As a result of determination of the second RS, the RS can be *(d,L7,e,L11,f,L15,i,L20,j,L21,l)* as shown in Fig. 3 (b). Our algorithm can extend RS until I' is equal to the original ingress node. As we extend the restoration scope, we enhance the resource utilization and reduce the restoration speed. From this perspective, the RS is gradually widened.

2.2 An Alternative Path Computation

Once we determine RS, we should find an alternative path avoiding the abnormal link. In this section, we describe the algorithm to find an optimal alternative path within RS. This algorithm is very simple because it utilizes the weight in WNG. We traverse WNG from the egress node until we reach the intermediate ingress node in accordance with the following rules.

1. Select the link having the least weight among the links connected to a node.

2. If there are two or more links whose weights are equal, we select the link having the largest residual bandwidth ($B_{RESIDUAL} = B_{AVA} - B_{REQ}$).
3. If there are two or more links whose weights and the residual bandwidth are equal, we select the link having the least delay.
4. If there are two or more links whose weights, residual bandwidth, and delay are equal, we select an arbitrary link.

3 Performance Issues

In order to simulate the proposed restoration scheme and compare the restoration performance of our restoration scheme with two existing backup protection schemes, global backup [4,11] and reverse backup [5,11], we used the simple network topology as shown in Fig. 5.

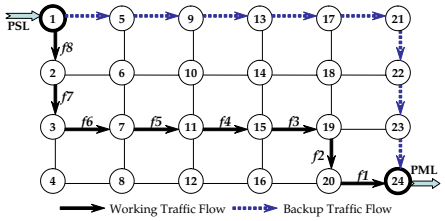


Fig. 5. A network topology for simulation

In this topology, there are 24 nodes and 38 links. With $LSP_{REQ}(1, 24, 24ms, 10Mbps)$, we create a working LSP traversing *(a-c-l-m-p)* and create a global backup path traversing *(1-2-3-7-11-15-19-20-24)* and a reverse backup path traversing *(1-5-9-13-17-21-22-23-24)*. Each node is connected with a duplex link with a 50Mbps bandwidth, 5ms delay and a CBQ queue. We use one pair of real-time traffic that is inserted into node *a* corresponding to the PSL and escapes through node *p* corresponding PML. We use a real-time traffic model for setting up LSPs from a working LSP, a global backup LSP and a reverse backup LSP with specific bandwidth requirements as the QoS traffic. For this, the traffic generator [10] generates 256-byte packets at 10 Mbps and at a constant bit

rate. We define the 8 fault locations *f1, f2, f3, f4, f5, f6, f7* and *f8* along the working LSP to measure the restoration performance according to the different fault location. The measured performances are shown in Fig. 6.

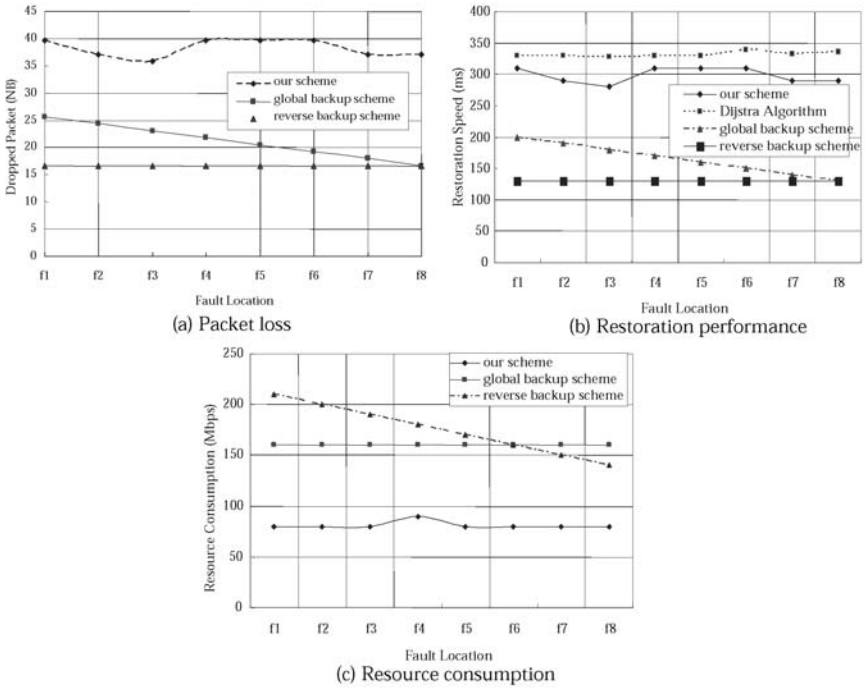


Fig. 6. Restoration performance comparison

There are various performance evaluation criterions in relation to MPLS path restoration, such as recovery time, full restoration time, setup vulnerability, backup capacity, additive latency, reordering, state overhead, packet loss, and coverage [8]. Because our algorithm focuses on the maximization of the network resource utilization and the moderation of the restoration speed, we evaluate packet loss, resource utilization and restoration performance in the proposed restoration scheme. Fig. 6 (a) shows the packet loss depending on the four different fault locations along the working LSP. From the perspective of packet loss, our scheme shows the worst performance comparing with the two protection schemes of the global backup scheme and the reverse backup scheme, which is a natural result because our scheme dynamically restores the fault. Fig. 6 (b) shows the resource utilization performance at the eight different fault locations. In order to measure the resource utilization, we define three metrics, label, bandwidth and buffer used, for the working LSP and the two backup paths. Our

scheme shows the best performance compared to other schemes in terms of resource utilization because our scheme finds the alternative path avoiding the fault location, taking into account the global network status. Fig. 6 (c) shows the restoration performance. If we compare our scheme with other protection schemes, it is a natural result that our scheme shows the worst performance compared with the protection schemes because our scheme dynamically finds the alternative path for restoration. However, if we compare our restoration algorithm with the Dijstra algorithm in terms of restoration speed, our algorithm shows a better performance than the *Dijstra* algorithm because our scheme minimize the restoration scope according the fault location using the proposed algorithm of RS determination. Another important point is to identify the relation between restoration speed and resource reusability because the reusability of the existing working LSP for restoration is entirely related to the restoration performance of the dynamic restoration scheme, including our scheme. In addition, we introduced the concept of RS extension in order to enhance the restoration speed and resource utilization.

4 Concluding Remarks

This paper has proposed a rerouting algorithm to enhance restoration speed and resource utilization compared to existing rerouting schemes. The algorithm for the dynamic RS adjustment determines the most reasonable candidate nodes or links to be applied to the restoration and assigns the appropriate weights to the candidate nodes and links. Our algorithm showed a higher performance than the rerouting approach based on the Dijstra algorithm in terms of restoration speed and resource utilization. On the other hand, we can enhance the restoration speed with the concept of the dynamic adjustment of the restoration scope that minimizes the complexity of network topology which will be used for restoration. Also, by the extension of the RS, we reduced the restoration failure probability. Our restoration scheme showed that the fault location directly affected the restoration speed and the resource reusability of the working LSP. On the basis of the performance evaluation results, we concluded that our scheme can be applicable for the protection of bronze-class working LSPs and all kinds of backup LSPs, such as the global backup LSP and the reverse working LSP.

References

1. E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF RFC3031, 2001.
2. D. Awduche, J. J. Malcolm, J. Agogbua, M. O'Dell and J. McNabus, "Requirements for Traffic Engineering over MPLS," IETF RFC2702, 1999.
3. D. Awduche et al., "Requirements for Traffic Engineering Over MPLS," IETF REC 2702, 1999.
4. C. Huang, V. Shrma, S. Makam, Ken Owens, "A Path Protection/Restoration Mechanism for MPLS Networks," Internet Draft, draft-chang-mpls-path-protection-02.txt, 2000.

5. D. Haskin, R. Krishnan, "A Method for Setting an Alternative Label Switched Paths to Handle Fast Reroute," Internet Draft, draft- haskin-mpls-fast-reroute-05.txt, 2000.
6. V. Sharma, B.M. Crane, K. Owens, C. Huang, F.Hellstrand, B. Cain, S. Civanlar and A. Chiu, "Framework for MPLS-based Recovery," Internet Draft, draft-ietf-mpls-recovery-frmwrk-03.txt, 2001.
7. C. Huang, V. Sharma, K. Owens, and S. Makam, "Building Reliable MPLS Networks Using a Path Protection Mechanism," IEEE Communications Magazine, pp156-162, March 2002.
8. G. Ahn and W. Chun, "MPLS Restoration Using Least-Cost Based Dynamic Backup Path," ICN2001, Springer LNCS 2094, pp319-328, 2001.
9. E. Harrison, A. Farrel, and B. Miller, "Protection and Restoration in MPLS Networks," Data Connection (http://www.dataconnection.com), October 2001.
10. Helsinki University of Technology, "QoS Routing Simulator (QRS) Version2.0," Available at http::://www.tct.hut.fi/ pgzhang/QRS/index.html.
11. G. Ahn and J. Jang, "An Efficient Rerouting Scheme for MPLS-based Recovery and Its Performance Evaluations," Telecommuincation Systems, Vol. 3, pp.481-495, 2002.