A Segment-based Protection Scheme for MPLS Network Survivability

Daniel W. Hong[*], Choong Seon Hong[**], Woo-Sung Kim[*]
[*]Network Technology Lab.
52 Junmin-Dong Yuseong-Gu, Daejeon 305-811 Korea
wkhong@kt.co.kr, kwsun@kt.co.kr
[**]Department of Computer Engineering, Kyung Hee University
cshong@khu.ac.kr

***Abstract*—This paper proposes a new approach, segment-based path protection, which provides much more enhanced network resource utilization than the local protection scheme and achieves much more fast restoration than the global protection scheme. This segment-based protection scheme consists of two subsequent steps: determination of the optimal restoration scope taking the working path state, bandwidth, and delay constraints into account and calculation of optimal segmented backup path that is link and node disjoint path with the found working path. In addition, we evaluate the performance of proposed protection scheme under the randomly generated Waxman's network topology.**

Keywords-MPLS, Survivability, Segemented Protection, Local Protection, Global Protection

I. INTRODUCTION

With the advent of fiber and its increasing deployment in networks, the risk of losing huge volumes of data due to a span cut or node failure has escalated. Because of fierce competition among service providers and customers intolerance of disruption of service, survivability of a network has assumed great importance. Survivability refers to the ability of a network to provide continuity of service with no disruption, no matter how much the network may be damaged due to events such as fiber cable cuts or node failures due to equipment breakdown at a central office or other events such as fires, flooding, etc.). As a result, network designers are beginning to incorporate provisioning of services over disjoint paths, so that if one path fails due to a link or node failure, the second path can carry the traffic to its destination.

This paper describes the construction of algorithms for finding the optimal disjoint paths between a given pair of nodes in an MPLS network. The key features of the algorithms will be optimality, since the aim is to reduce network costs. The algorithms will be applicable not only to the ideal graph-theoretic networks (described by nodes and links), but also to the more practical cases of networks described by nodes, links, and spans. Because of economic and practical considerations, spans, which are the actual physical connections in a network, can be shared by more than one link. Span-sharing complicates the network, and leads to network topologies not found or discussed in books. This paper will discuss such network topologies and special algorithms for finding disjoint paths in such networks. It will also discuss the construction of a survivable network design based on physical-disjointness, and basic network structures that permit such disjoint paths.

On the other hand, fast rerouting or protection switching uses pre-established LSPs. When a fault is detected, the protected traffic is switched over to the alternative LSP(s). Establishing pre-established alternative paths, results in a faster switchover compared to establishing new alternate paths on demand [4-7] rerouting can be accomplished by protection mechanisms that are activated locally or that are global in scope. Local repair uses an alternative Label Switched Path (LSP) that serves as a bypass from the point of protection to the next LSR node or to the destination. The techniques proposed for local repairs in MPLS networks are splicing and stacking [8].Global repair is activated on an end-to-end basis.

That is, an alternative LSP is pre-established from ingress to egress nodes of the path to be protected. Our proposal combines both these techniques. The main factors that affect the performance of fast rerouting mechanisms are: packet loss, traffic recovery delay (full restoration time) and packet disorder. There are some existing fast restoration schemes [1-8]. Both Haskim's [4] and Makam's [6] schemes are poor in resource utilization and cannot support protection in case of link/node failures on both a working path and its recovery path. This is because they use the protection switching model. In order to enhance resource utilization and restoration speed, Yoon [5] proposed a segmented protection scheme. However, Yoon did not propose the way to determine the segment. So, segment is determined by network operator's heuristic. Therefore, the resource utilization and restoration speed are variant according to the different operator's heuristic.

To solve this problem, we propose a segment-based path restoration scheme that can dynamically determine the restoration scope, called segment, based on the found working path and bandwidth and delay constraints. So, we can maximize the restoration speed and provide the QoS guaranteed backup paths.

II. PROPOSED SEGMENT-BASED PROTECTION SCHEME

In this section, we describe the segmented based backup path computation scheme, which main idea is to define the working path as a sequence of segments and protect each segment separately. The entire working path can be broken down into variable sized segments as shown in Figure 1.

1-4244-0143-7/06/$20.00 ©2006 IEEE

In segment-based protection, the idea is to provide protection for each segment as a whole, instead of providing protection for the whole path or each link separately. Each segment consists of a Segment Switching LSR (SSL) and a set of protected routers. SSL is responsible for switching over the traffic to the backup path in case of any failure among the protected routers or links connecting the routers in the segment.

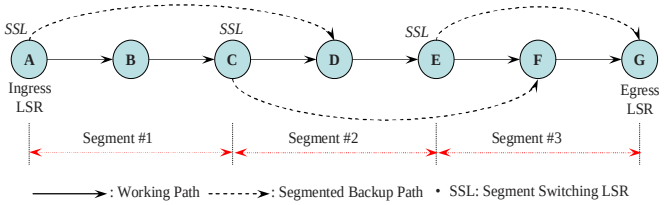

Figure 1. Segment-based protection scheme

For example, in the case of Segment #1 shown in Figure 1, LSR *A* is the Segment Switching LSR (SSL) and it protects the LSRs *B* and *C* and the links connecting LSR *A*, LSR *B* and LSR *B* and LSR *C*. We can denote this segment by $S_i(A,B,C)$, where $_i$ represents the *i*th segment along the working path . Similarly in Segment #2 ($S_2(C,D,E)$) , LSR *C* protects LSR *D* and *E* and the interconnecting links. Note that the SSL cannot protect itself but is protected by the upstream SSL. In case of LSR *E* fails which is the SSL of Segment #3, its failure will be protected by the SSL *C*.

A. Order Assignment Process

We reuse the already proposed algorithm of *Alignment()* [10]. With *alignment()* algorithm [10], we can archive the *weighted network graph (WNG)*. We can define the LSP constraints such as $LSPreq(B_{wp},D_{wp},IL_{wp},EL_{wp})$, where B_{wp} is requested bandwidth, D_{wp} is requested end-to-end delay, IL_{wp} is the ingress LSR and EL_{wp} is egress LSR.

B. Working Path Computation

With the weighted network graph, we find the optimal working path traversing WNG from destination LSR to source LSR meeting the requested QoS constraints: bandwidth and end-to-end delay. In order to find the working path, we traverse WNG from destination until source reaches with the working path computation algorithm as shown in [10]. As a result of this algorithm, an optimal working path is found as shown in Figure 2.

As we assume that available bandwidth and delay on every links are 5 Mbps and 1ms on the sample network topology. On traversing WNG and selecting links, we add accumulated delay to *WP(d)* and adjust available bandwidth of the selected link according the request bandwidth ($B_{ava} = B_{ava}-B_{req}$). Therefore, the selected available bandwidth on selected link is 3 Mbps because the initial available bandwidth was 5 Mbps and request bandwidth was 2 Mbps as shown in Figure 3. And the accumulated delay on *LSP(d)* is 8ms. As a result of working path computation, we find the optimal working path between ingress LSR and egress LSR meeting the requested QoS constraints: bandwidth and end-to-end delay and compute the accumulated delay on the selected working path.

B_{ava} : Available bandwidth on an link
B_{req} : Requested bandwidth for establishing LSP
B_{res} : Residual bandwidth on an link ($B_{res}= B_{ava}-B_{req}$)
D : Delay on an link
$WP(B_a,D_a,H)$: Working Path, where B_a is accumulated bandwidth, D_a is accumulated delay, and H is total hop count

```
do {
    select the link having the least order among links connected to LSR
        if there are two or more links having same order
            select the link having largest available bandwidth among the links
                if there are two or more links having same residual bandwidth
                    select the link having least delay among the links
                        if there are two or more links having same delay
                            select an arbitrary link
    adjust the available bandwidth on a link as B_ava = B_ava – B_req
    add LSR and selected link to WP
    accumulate the delay of WP as WP(D_a)=WP(D_a)+E(D)
    accumulate the bandwidth of WP as WP(B_a)=WP(B_a)+B_req
    increment hop count by one as WP(H) = WP(H)+1
    determineSegment()
} until (ingress LSR reaches)
```

Figure 2. A working path computation algorithm

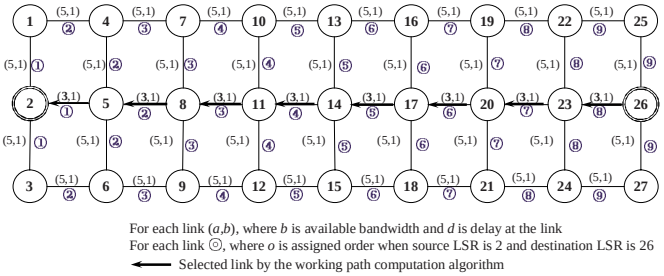

Figure 3. An example of working path selection under the weighted network graph

C. SSL Selection for Optimal Segment Determination

In this section, we describe the way to determine the reasonable restoration scope called segment taking into account the overall network topology and state, requested constraints of bandwidth and end-to-end delay for creating working path and the found working path information such as total consumed bandwidth, accumulated delay and the total hop count.

The number *k* can be variable according to the total hop count, accumulated delay, total consumed bandwidth for a working path. Matter of thing to determine *k* is to minimize the network resource utilization, to guarantee the requested end-to-end delay and to maximize restoration possibility. In addition, we must consider the required total restoration time.

There are two major aspects affecting the restoration time: failure notification to nearest Segment Switching LSR (SSL) and switch over time from working path to pre-provisioned

backup path at SSL. Let's assume that the switching over time (T_{so}) is fixed because it depends on the protection switch performance varying from vendor to vendor (T_{so} = constant). Therefore, the critical factor to determine k is the delay to notify failure occurrence to nearest SSL along the reverse direction of the working path. We define the failure notification time as T_{fn}.

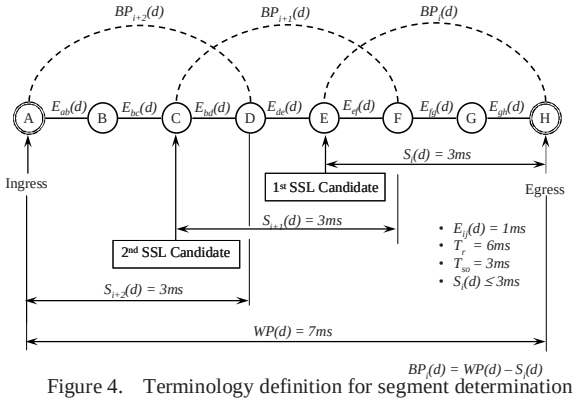

Figure 4. Terminology definition for segment determination

As shown in Figure 4, we define some terminologies for determination of segment. For example, we assume that a working path between LSR A and LSR H traverses <A-B-C-D-E-F-G-H> and delay of all links is *1ms*. $E_{ij}(d)$ is the delay on the link between LSR i and LSR j. $S_i(d)$ is the accumulated delay within the i^{th} segment boundary, which should be less than or equal to the failure notification time (T_{fn}). That is to say, $S_i(d) \le T_{fn}$. If $S_i(d)$ is larger than T_{fn}, we cannot guarantee the recovery within the predetermined recovery time.

$WP(d)$ represents the accumulated delay on working path. At this example of Figure 4, we define the switch-over time (T_{so}) at each SSL and ingress LSR as 6ms. In addition, we define the total recovery time (T_r) as 6ms. Therefore, we must define the T_{fn} as 3ms, which is corresponding to $S_i(d)$. When we compute the accumulated delay of working path ($WP(d)$), we evaluate whether it exceeds the T_{fn} or not. The $S_i(d)$ should be less than or equal to T_{fn}. If it firstly exceeds T_{fn}, we select the LSR connected to the link as a candidate SSL. For example, as we assume that T_{fn} is *3ms* in Figure 4, the accumulated delay on working path in the direction from egress node to ingress node is firstly exceed when it encounter E_{de}. Therefore, the first candidate SSL can be LSR E that is connected to the link firstly exceeding the T_{fn}.

D. Segmented Backup Path Computation

Once finding the working path and selecting the possible SSLs along the working path, we compute the backup paths for each segment taking into account the end-to-end delay and bandwidth constraints.

In order to find the optimal backup path for each segment, we must consider the end-to-end delay constraint for each backup path. Let's assume that requirement for creation of LSP is defined by $LSP_{req}(B_{wp},D_{wp},IL_{wp},EL_{wp})$, where B_{wp} is requested bandwidth, D_{wp} is requested end-to-end delay, IL_{wp} is ingress LSR and EL_{wp} is egress LSR. The requirement for i^{th} backup path creation can be defined by $LSP_{req}(B_{bp},D_{bp},IL_{bp},EL_{bp})^i$, where B_{bp}^i is same with B_{wp} but D_{bp}^i, IN_{bp}^i, and EG_{bp}^i are different from those of working path. After identifying the necessary requirements for creating segmented backup path, we find the optimal backup path taking into account bandwidth and delay constraints. The backup path computation algorithm is shown in Figure 5, which consists of two major procedures: trimming and order assignment.

At first we trim the link and nodes on which availabilities are not normal (e.g., fault, performance degradation, congestion, and administrator's affinity) and trim the link on which available bandwidth is less than the requested bandwidth for simplifying the network topology. In addition, we further simplify network topology by trimming the node and link which orders are less than that of SSL.

1. Trims the links pertained in already found working path and backup paths.
   - 1.1 Trims the unfavorable links or nodes
     - 1.1.1 status on link or node is abnormal
     - 1.1.2 residual bandwidth on a link is less than the requested bandwidth
   - 1.2 Trims links and nodes, whose orders are less than the order of link connected to SSL.

2. Assigns appropriate orders to each node and link from ingress LSR and the first SSL.

3. Assigns appropriate order to each node and link from intermediate SSL

Figure 5. Algorithm for computing segmented backup path

Next, we assign appropriate orders to each link and node with trimmed network topology and find the optimal segmented backup path conforming to bandwidth and delay constraints.

III. PERFORMANCE DISCUSSION

In order to evaluate feasibility of proposed segmented protection scheme, we consider a version of the prominent random network topology generator introduced by Waxman [8] as shown in Figure 6. The connected random graphs are widely used for testing different algorithms on networks because random graph is good model for the task as using different real network structures for algorithms testing is usually impossible.

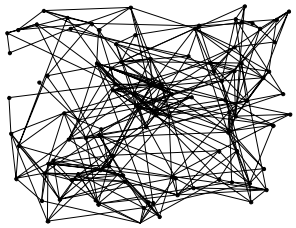

Figure 6. Randomly generated Waxman network topology

This randomly generated network topology consists of 102 nodes and 367 links. In addition, the bandwidth of each link is assigned as 50Mbps, while the delay of each link is set to 1ms. In this Waxman's topology generator [13], the nodes are distributed according to a Poisson process in the

plane. This means that the number of nodes is Poisson distributed with the intensity proportional to the area of the domain. Given the number of nodes the nodes are uniformly distributed in the plane. We generate this topology with *a=1*, *b=1*, and intensity of the Poisson process equal to *0.1*.

Network resource utilization is measured by the total reserved bandwidth for backup path provision in the case of global protection scheme (GPS), local protection scheme (LPS), and segmented protection scheme (SPS). Especially, in the case of SPS, we measured the total reserved bandwidth for configuring segmented backup path by gradually increase the total recovery time (T_r=10ms, 12ms, 14ms, and 16ms).

In terms of resource utilization, the global backup scheme showed much more enhanced utilization ratio than local protection scheme and segment protection scheme. In the case of local protection and segment protection schemes, they need to configure the backup paths in overlapped, which cause the waste of bandwidth. On the hand, segment protection scheme showed more resource utilization than local protection scheme. In this simulation, we assume that the switch-over time (T_{so}) at each LSR is constant (3 ms) because the switch-over time at LSR can be vary from vendor to vendor. Therefore, in this simulation model, the total recovery time (T_r) depends entirely on the fault notification time (T_{fn}) from the fault location to the SSL. Therefore, as we gradually moderate the total fault recovery time (T_{fn} = 10ms, 12ms, 14ms, and 16ms), the resource utilization is also gradually enhanced in proportion to the number of ms for T_r.

In addition, if we minimize the T_r (ms), then the segment scope can be narrowed, but if we increase the T_r (ms), then the segment scope can be also widen because the total recovery time (T_r) in our simulation model depends on the fault notification time (T_{fn}). Therefore if we maximize T_r (ms), then we can archive high resource utilization as much as that of global protection scheme. If we minimize T_r (ms), then we can archive low resource utilization as much as that of local protection scheme.

In order to measure, we create 100 working paths and their segmented protection paths with bandwidth requirement of 10kbps, end-to-end delay constraint of 10ms and total recovery time constraint of 5ms. Our scheme absolutely guarantees the end-to-end delay constraint on the protected segment backup paths. But Yoon's [5] and Gupta's [11] scheme failed to guarantee end-to-end delay constraint on most of protected segment backup path except approximately 30 LSPs.

IV. CONCLUSION

In this paper, we propose a segment-based protection scheme that dynamically determines the segment switch LSR (SSL) taking into account the guarantee of end-to-end delay constraint. We propose an algorithm that determines the reasonable SSL taking the role of PSL in MPLS network considering the total recovery time (T_r) and we also describe the algorithm to determine the optimal backup path that is working path disjoint and guarantees end-to-end delay constraint. With performance evaluation, our scheme showed higher acceptance ratio and resource utilization than any other existing segment protection schemes such as Yoon's [5] and Gupta's [11] schemes.

However, in terms of resource utilization, our scheme showed enhanced performance than local backup scheme but showed somewhat degraded performance than global backup scheme. Because, global protection scheme needs only one working path and one backup path but our scheme must pre-configured one working path and number of segmented backup paths. But, local protection needs more backup paths than that of our model because segment scope of local backup scheme will be narrower than our model. In terms of protection performance, our scheme absolutely recovered all LSPs regardless of fault location within the imposed recovery constraints. Gupta's algorithm also showed same recovery speed with our model but Yoon's scheme showed unstable showed unstable recovery performance.

Our scheme absolutely guarantees the end-to-end delay constraint on the protected segment backup paths. But Yoon's [5] and Gupta's [11] scheme failed to guarantee end-to-end delay constraint on most of protected segment backup path.

REFERENCES

[1] Der-Hwa Gan, P. Pan, A. Ayyangar, and K. Kompella, "A method for MPLS LSP fast-reroute using RSVP detours," Internet Draft, Apr. 2001.

[2] V. Sharma and F. Hellstrand, "Framework for multi-protocol label switching (mpls)-based recovery," request for comments 3469, 2003.

[3] J.-M. Chung, "Analysis of mpls traffic engineering," in Circuits and Systems, in Proceedings of the 43rd IEEE Midwest Symposium on, vol. 2, pp. 550–553 vol.2, 2000.

[4] D. Haskin and R. Krishnan, "A method for setting an alternative label swtiched paths to handle fast reroute," IETF Internet Draft, draft-haskin-mpls-fastreroute-05.txt., 2000,

[5] S. Yoon, H. Lee, D. Choi, Y. Kim, G. Lee, and M. Lee, "An efficient recovery mechanism for mpls-based protection lsp," In Proceedings of Joint 4th IEEE International Conference on ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001., pp. 75–79, 2001.

[6] K. Owens, V. Sharma, S. Makam, and C. Huang, "A Path Protection/Restoration Mechanism for MPLS Networks," IETF, Internet draft <draft-chang-mpls-protection-03.txt>, July 2001.

[7] T.V. Lakshman, M. Kodialam, "Dynamic Routing of Bandwidth Guaranteed Tunnels with Restoration" in Proceedings of IEEE INFOCOM 2000, April 2000.

[8] B.M. Waxman, "Routing of Multipoint connections," IEEE Journal of Selected Areas on Communications, Vol.6 (9), pp.1617-1622, 1988.

[9] Ingmar Kaj, Raimundas Gaigalas, Stochastic simulation using MATLAB, http://www.math.uu.se/~ikaj/courses/matlab.

[10] D.W. Hong, C.S. Hong, "A Rerouting Scheme with Dynamic Control of Restoration Scope for Survivable MPLS Network," ICOIN2005, Lecture Notes in Computer Science, LNCS3391, pp.233-243, 2005.

[11] A. Gupta, et al, QoS Aware Path Protection Schemes for MPLS Networks, In Proceedings of International Conference of Computer Communications, August 2002.