A Disjoint Path Selection Scheme With Shared Risk Link Groups in GMPLS Networks

Eiji Oki, *Member, IEEE*, Nobuaki Matsuura, Kohei Shiomoto, *Member, IEEE*, and Naoaki Yamanaka, *Fellow, IEEE*

***Abstract*—This letter proposes a disjoint path selection scheme for generalized multi-protocol label switching (GMPLS) networks with shared risk link group (SRLG) constraints. It is called the weighted-SRLG (WSRLG) scheme. It treats the number of SRLG members related to a link as part of the link cost when the k-shortest path algorithm is executed. In WSRLG, a link that has many SRLG members is rarely selected as the shortest path. Simulation results show that WSRLG finds more disjoint paths than the conventional k-shortest path algorithm.**

***Index Terms*—Disjoint path algorithm, GMPLS, SRLG.**

I. Introduction

WITH optical fiber bandwidth and node capacity increasing explosively, a break in a fiber span or node failure can cause a huge damage to customers [1]. Therefore, network providers should design survivable networks so that the communication loss can be minimized. A disjoint path routing enhances the survivability of a network. Several disjoint paths, which are routed without sharing the same links or nodes, must be set between source and destination nodes.

Generalized multi-protocol label switching (GMPLS) is being developed in the Internet Engineering Task Force (IETF) [2]. It is an extended version of multi-protocol label switching (MPLS). While MPLS was originally developed to control the IP-packet layer, GMPLS controls several layers, such as IP-packet, time-division-multiplexing (TDM), wavelength, and optical-fiber layers, in a distributed manner. IP link-state-based routing protocols, such as open shortest path first (OSPF) [3] and Intermediate System to- Intermediate System (IS-IS) [4] are being extended to support GMPLS by advertising TE (Traffic Engineering)-link states defined for each layer. In the OSPF/IS-IS extensions, information about the Shared Risk Link Group (SRLG) is also advertised. For example, when multiple wavelengths are carried on the same fiber, SRLG information is taken into account for the disjoint path routing.

Several algorithms have been presented to efficiently find maximum disjoint paths between source and destination nodes [5]–[9]. Bhandari [10] described several algorithms to take account of the equivalent of SRLGs. However, no efficient algorithm has addressed finding maximum disjoint paths taking into consideration SRLGs in GMPLS networks.

Manuscript received April 4, 2002. The associate editor coordinating the review of this letter and approving it for publication was Prof. D. Petr.

E. Oki, K. Shiomoto, and N. Yamanaka are with NTT Network Innovation Laboratories, Tokyo 180-8585, Japan (e-mail: oki.eij@lab.ntt.co.jpi; shiomoto.kohei@lab.ntt.co.jp; yamanaka.naoaki@lab.ntt.co.jp).

N. Matsuura is with NTT Network Service Systems Laboratories,Tokyo 180-8585, Japan (e-mail: matsuura.nobuaki@lab.ntt.co.jp).

Publisher Item Identifier 10.1109/LCOMM.2002.803477.

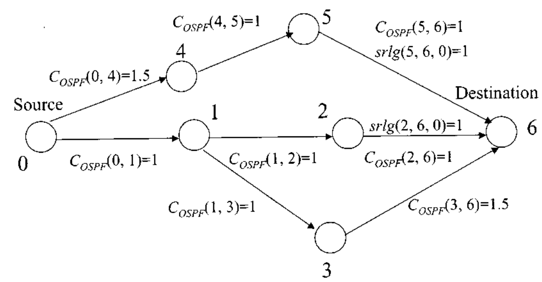

Fig. 1. Network example.

A k-shortest path algorithm is widely used to find disjoint paths because of its simplicity [10]. First, the shortest path algorithm provided by Dijkstra [11] is used to find the first path from a given network topology. Node disjoint paths are considered as disjoint paths in this paper[1] Next, the links and nodes that are used by the first route are pruned from the given network topology. Note that pruning a node means that links that is connected to it are also pruned. Then, the second shortest path is searched for in the pruned network topology. In the same way, the kth-shortest disjoint path is searched for. Although the k-shortest path algorithm does not find *maximum* disjoint paths, Dunn *et al.* [6] showed that its results are very nearly equal to the Max Flow solution.

Let us explain how the k-shortest path algorithm is performed for a network with SRLG in a conventional manner. The first path is found based on link costs advertised by OSPF. OSPF is considered as a link-state-based routing protocol here. Then, the k-shortest path algorithm prunes the links and nodes on the first route and the links that belong to the same groups as the links and nodes on the first route do. To find more disjoint paths, the same procedure is followed. However, when a link that shares many risks with other links is used as a disjoint route, a problem occurs in that we are not able to obtain a sufficient number of disjoint paths.

Fig. 1 shows a network example. Consider the solution of disjoint paths between source node 0 and destination node 6. Link cost $C_{\rm OSPF}(i,j)$ for link $L(i,j)$[2] is advertised by OSPF. $\mathrm{srlg}(i,j,g)$ indicates the SRLG information, where $\mathrm{srlg}(i,j,g) = 1$ or 0 means that $L(i,j)$ does or does not belong to SRLG g, respectively. $L(5,6)$ and $L(2,6)$ belong to SRLG 0. Fig. 2 shows an SRLG example for $L(5,6)$ and $L(2,6)$. Both links use the same fiber by using wavelengths $\lambda 1$ and $\lambda 2$, respectively, through an optical crossconnect. When

[1] Although we treat *node* disjoint paths in this paper, we can apply our concept to *link* disjoint paths in a similar way.

[2] In GMPLS, link $L(i,j)$ is referred to as a *TE-link*. We call it a *link* for simplicity here.

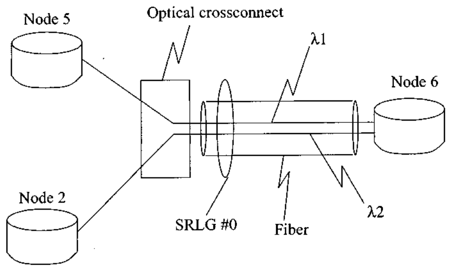

Fig. 2. SRLG example.

the k-shortest algorithm is used by taking C_{OSPF} as the link cost, route 0-1-2-6 is selected as the shortest path. To find the second disjoint path, links and nodes on the first selected path, which are $L(0,1)$, node 1, $L(1,2)$, node 2, and $L(2,6)$ are pruned. In addition, since $L(5,6)$ belongs to SRLG 0, to which $L(2,6)$ belongs, $L(5,6)$ is also pruned. As a result, we cannot find more than one disjoint path between nodes 0 and 6.

This paper proposes a disjoint path-selection scheme for GMPLS networks with SRLG constraints. It is called the weighted-SRLG (WSRLG) scheme. It treats the number of all SRLG members related to a link as part of the link cost when the k-shortest path algorithm is performed. In WSRLG, a link that has many SRLG members is rarely selected as the shortest path. Simulation results show that WSRLG finds more disjoint paths than the conventional k-shortest path algorithm.

II. Weighted-SRLG Path Selection Algorithm

The terminology used in this paper is as follows:

- $C_{\mathrm{OSPF}}(i,j)$ link cost for $L(i,j)$ that is advertised by OSPF;
- $\mathrm{srlg}(i,j,g)$ SRLG information; $\mathrm{srlg}(i,j,g) = 1$ means that $L(i,j)$ belongs to SRLG g and $\mathrm{srlg}(i,j,g) = 0$ means that it does not;
- $C_{\mathrm{comp}}(i,j)$ link cost for $L(i,j)$ that is used to find k-shortest paths;
- α weight factor for SRLG;
- $D_{\mathrm{req}}(s,d)$ required number of disjoint paths between nodes *s* and *d*;
- $C_{\mathrm{path}}(s,d)$ sum of costs for all disjoint paths between nodes *s* and *d*;
- $K(s,d)$ number of obtained disjoint paths between nodes *s* and *d*;
- G number of SRLGs in a network;
- $M(g)$ number of members for SRLG g. $M(g) = \sum_i \sum_j \mathrm{srlg}(i,j,g)$.

$C_{\mathrm{comp}}(i,j)$ is defined by

$$C_{\mathrm{comp}}(i,j) = \frac{1-\alpha}{C_{\mathrm{OSPF}}^{\max}} C_{\mathrm{OSPF}}(i,j) + \frac{\alpha}{\mathrm{SRLG}^{\max}} \max\{\mathrm{SRLG}(i,j), 1\} \quad (1)$$

where

$$\mathrm{SRLG}(i,j) = \sum_{g}^{G} M(g) srlg(i,j,g) \quad (2)$$

$$C_{\mathrm{OSPF}}^{\max} = \max_{i,j} C_{\mathrm{OSPF}}(i,j) \quad (3)$$

and

$$\mathrm{SRLG}^{\max} = \max_{i,j} \mathrm{SRLG}(i,j). \quad (4)$$

$C_{\mathrm{path}}(s,d)$ is expressed by

$$C_{\mathrm{path}}(s,d) = \sum_{k=1}^{K} \sum_{(i,j)\in \mathrm{path} k} C_{\mathrm{OSPF}}(i,j). \quad (5)$$

In WSRLG, α is set to the smallest value possible by using a well-known binary search method so that $C_{\mathrm{path}}(s,d)$ can be minimized under the condition that $K(s,d)$ is equal to or larger than $D_{\mathrm{req}}(s,d)$.

We set $\alpha_{\min} = 0.0$ and $\alpha_{\max} = 1.0$ as initial values. ϵ is used as a parameter to judge whether the binary search method converges. WSRLG is described as

- **Step 1:** $\alpha = (\alpha_{\min} + \alpha_{\max})/2$.
- **Step 2:** $K(s,d)$ is calculated by the following k-shortest path algorithm by taking SRLG into account.
- **Step 3:** If $K(s,d) < D_{\mathrm{req}}(s.d)$, then $\alpha_{\min} = \alpha$ is set. Otherwise, $\alpha_{\max} = \alpha$ is set.
- **Step 4:** If $\alpha_{\max} - \alpha_{\min} > \epsilon$, go to Step 1. Otherwise, go to Step 5.
- **Step 5:** A set of paths, in which $K(s,d) - D_{\mathrm{req}}(s,d)$ is the smallest value, is considered as a solution. If more than one path set is found, the one with the smallest value of $C_{\mathrm{path}}(s,d)$ is taken as the solution.

The k-shortest path algorithm with SRLG is described below. First, we set $k = 1$ as the initial value.

- **Step 1:** The kth shortest path between source and destination nodes is searched for based on link cost $C_{\mathrm{comp}}(i,j)$. If the path is found, go to Step 2. Otherwise, $K(s,d) = k$ is set and the k-shortest algorithm is ended.
- **Step 2:** Prune links $L(i,j)$ and nodes that are on the kth shortest path. For all gs, if $\mathrm{srlg}(i,j,g) = 1$, all links $L(i\prime,j\prime)$, where $\mathrm{srlg}(i\prime,j\prime,g) = 1$, are also pruned.
- **Step 3:** We set $k = k+1$ and go to Step 1.

Consider that WSRLG is applied to the network model in Fig. 1. α is set to 1.0, as a special case. WSRLG finds two disjoint paths between nodes 0 and 6, while the conventional k-shortest scheme finds only one path. $C_{\mathrm{comp}}(2,6)$ and $C_{\mathrm{comp}}(5,6)$ are twice as large as other $C_{\mathrm{comp}}(i,j)$. Route 0-1-3-6 is selected as the shortest path. Then, route 0-4-5-6 is found as the second shortest path. Thus, WSRLG is able to find more disjoint paths.

III. Performance Evaluation

To evaluate the WSRLG performance, we use network topologies generated in a random manner under the condition that average node degree D is satisfied for a given number of nodes N and at least one path exists between every source-destination node pair. D is the average number of other nodes to which individual nodes are connected by links. D is defined as $D = \left(\sum_i^N \sum_j^N a_{ij}\right)/N$, where $a(i,j)$ is the (i,j) element of the network adjacency matrix. If there is a link from node i to node j, $a(i,j)$ is set to 1. Otherwise, $a(i,j)$ is set to 0. Other evaluation conditions are assumed to be as follows.

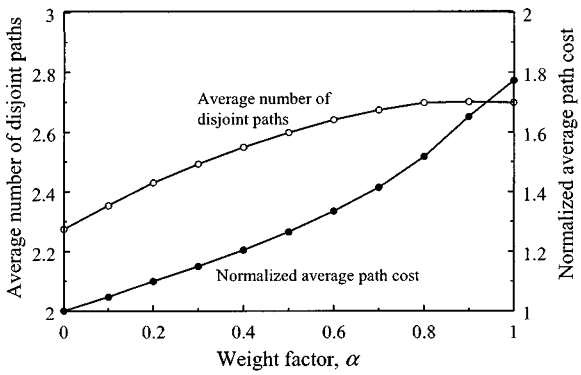

Fig. 3. α vs. number of disjoint path and path cost $D_{\mathrm{path}}(s,d)$ ($N = 20$, $D = 6$, $G = 6$, $m = 14$).

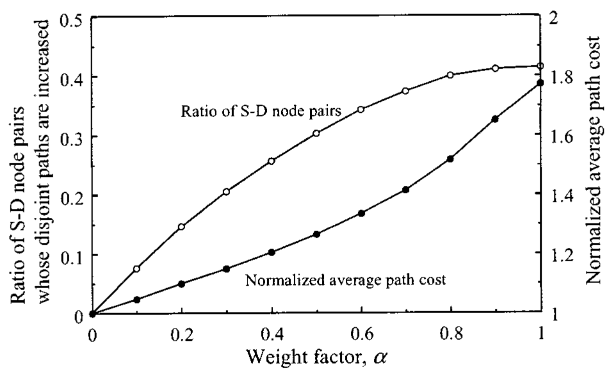

Fig. 4. α vs. ratio of S-D node pairs whose number of disjoint paths is increased and path cost $D_{\mathrm{path}}(s,d)$ ($N = 20, D = 6, G = 16, m = 14$).

$M(g)$ is set to m, which is independent of g to simplify the discussion. $\mathrm{srlg}\,(i,j,g)$ is set randomly under the condition that $\sum_i \sum_j \mathrm{srlg}\,(i,j,g) = m$ is satisfied. C_{OSPF} is set randomly between 0 and 1. The number of sample network topologies for each evaluation is more than 100.

Fig. 3 shows the average number of disjoint paths between source and destination nodes and the normalized average path cost of $C_{\mathrm{path}}\,(i,j)$, where the path cost for $\alpha = 0.0$ is assumed to be 1.0. In our evaluation, we set α manually to show its dependency. As α increases, the average number of disjoint paths becomes large. When $\alpha = 1$, the average number of disjoint paths is 19% more than for $\alpha = 0$. Note that $\alpha = 0$ means that the conventional disjoint path selection algorithm is used. $C_{\mathrm{path}}\,(i,j)$ also increases with α. When α is large, the second term in (1) is taken into account more than the first term in finding the shortest paths. By using WSRLG, α is automatically determined so that the required number of disjoint paths can be found while minimizing the increase in path cost.

Fig. 4 shows how many source-destination node pairs has more disjoint paths for $\alpha > 0$ compared to $\alpha = 0$, divided by the total number of source-destination node pairs, expressed as a ratio. WSRLG with $\alpha = 1$ dramatically increased the ratios to 41%.

IV. Conclusions

This paper presented the weighted-SRLG scheme, called WSRLG, to find disjoint paths for GMPLS networks with SRLG constraints. By applying WSRLG to randomly generated network topologies, we observed that WSRLG finds more disjoint paths than the conventional k-shortest path algorithm. The ratios of source–destination node pairs whose number of disjoint paths is increased were 41% higher than for the conventional algorithm. In addition, since WSRLG adjusts the weight of the SRLG factor by using a binary search algorithm, while satisfying the required number of disjoint paths between source and destination nodes, it can find cost-effective disjoint paths.

References

[1] E. Oki and N. Yamanaka, "A recursive matrix-calculation method for disjoint path search with hop link number constraints," *IEICE Trans. Commun.*, vol. E78-B, no. 5, pp. 769–774, 1995.

[2] A. Banerjee, J. Drake, J. P. Lang, B. Turner, K. Kompella, and Y. Rekhter, "Generalized multiprotocol label swtching: An overview of routing and management enhacements," *IEEE Commun. Mag.*, pp. 144–150, Jan. 2001.

[3] J. Moy, "OSPF Version 2,", IETF RFC 2328.

[4] D. Oran, "OSI IS-IS Intra-Domain Routing Protocol,", RFC 1142.

[5] J. W. Suurballe, "Disjoint paths in a network," *Networks*, vol. 4, pp. 125–145, 1974.

[6] D. A. Dunn, W. D. Grover, and M. H. MacGregor, "Comparison of k-shortest paths and maximum flow routing for network facility restoration," *IEEE J. Select. Areas Commun.*, vol. 12, pp. 88–99, Jan. 1994.

[7] M. Pollack and W. Wiebenson, "Solution of the shortest-route problem – A REVIEW," *Oper. Res.*, vol. 8, pp. 224–230, 1960.

[8] A. Itai, Y. Perl, and Y. Shiloach, "The complexity of finding maximum disjoint paths with length constraints," *Networks*, vol. 12, pp. 277–286, 1982.

[9] C. Li, S. T. McCormick, and D. Simchi-Levi, "Finding disjoint paths with different path-costs: Complexity and algorithm," *Networks*, vol. 22, pp. 653–667, 1992.

[10] R. Bhandari, *Survivable Networks Algorithms for Diverse Routing*. Amsterdam, The Netherlands: Kluwer Academic , 1999.

[11] E. W. Dijkstra, "A note on two problems in connection with graphs," *Num. Math.*, vol. 1, pp. 269–271, 1959.