Fault-tolerant Multipath Traffic Engineering for MPLS Networks

Yongho Seok*, Youngseok Lee†, Nakjung Choi* and Yanghee Choi*
*School of Computer Science and Engineering
Seoul National University, Seoul, Korea
Email: {yhseok, fomula, yhchoi}@mmlab.snu.ac.kr
†Department of Computer Science and Engineering
Chungnam National University, Seoul, Korea
Email: yslee@cs.cnu.ac.kr

***Abstract*— Multipath traffic engineering utilizes several multiple paths for transporting the traffic demand between source LSR and destination LSR. Using the multipath, we can effectively control the network resource utilization. If maximally disjoint multiple LSPs are found, the multipath traffic engineering can provide appropriate fault-tolerant routing. In this paper, we propose the fault-tolerant multipath traffic engineering mechanism based on linear programming (LP). When the statistical traffic demand is known, we apply the traffic engineering with the following objective; set all LSPs configuration in order to find maximally disjoint paths for each node pair, subject to minimization of the maximum of link utilization. We propose new load-balancing mechanism, when some link failures are detected, which routes the traffic flowing on the the failed LSPs into available LSPs. After presenting the proposed LP solution, we discuss the result obtained by exploiting it in the case-study network.**

***Index Terms*— Traffic Engineering, MPLS network, Multipath routing, Fault-tolerant traffic rou ting**

I. INTRODUCTION

Currently, traffic engineering problem is one of the most interesting issues in communication networks. The focus of traffic engineering is to set up paths between edge routers in a network to route the traffic demand of each node pair, while achieving low congestion and improving the utilization of network resources. In practice, the key objective of traffic engineering is to minimize the utilization of the most heavily used link in the network, or the maximum of link utilization. Since queueing delay increases rapidly as link utilization becomes high, it is important to minimize the link utilization throughout the network so that no bottleneck link exists. It has been known that this problem of minimizing the maximum of link utilization could be solved by the multi-commodity network flow formulation.

Traffic engineering problem is classified into several categories, for example, non-bifurcation case and bifurcation case. When the traffic bifurcation is allowed, the solution of the traffic engineering problem leads to split the traffic over multiple paths between node pairs for satisfying the given objective. The advantage of the multipath routing is to provide more bandwidth and more effective usage of network resources than the single path algorithm. Especially the most important merit of multipath routing is the easy support of fault-tolerant traffic routing, when paths of node pairs are configured maximally disjoint. When some link failures are occurred, some paths are disconnected so that the traffic flowing on the paths need to be rerouted into another paths. Although some link failures occur, maximally disjoint multipath has the high probability of having at least one connected path between the source and destination node. If there is no connected backup path for each node pair, we need to find new paths. However, since it requires the additional control message and setup time, it is inappropriate to support fast recovery for high speed backbone networks, such as optical networks.

Multipath routing has been incorporated in recently developed or proposed routing protocols. The easiest extension to multipath routing is to use equal-cost multiple shortest paths, which is known as Equal-Cost Multi-Path (ECMP) routing. This is explicitly supported by several routing protocols such as Open Shortest Path First (OSPF) [2] and Intermediate System to Intermediate System (IS-IS) [3]. Some router implementations allow equal-cost multipath with Routing Information Protocol (RIP) or with some other routing protocols. In Multi-Protocol Label Switching (MPLS) networks [1], where IP packets are switched through the pre-established Label Switched Paths (LSPs) by signaling protocols, multiple paths can be used to forward packets belonging to the same "forwarding equivalent class (FEC)" by explicit routing. However, multipath traffic engineering may require more total network bandwidth resources, i.e. sum of assigned bandwidth at each link of paths, than single shortest path. Therefore, the maximum hop-count constraint should be incorporated into proposed multipath traffic engineering scheme in order not to waste bandwidth. In addition, the number of paths should be restricted between a source-destination pair in the real network topology for the relaxed administrative complexity.

For the implementation of multipath traffic engineering, we must consider how to distribute the traffic demand of each node pair into multiple parallel paths. The split ratio of each traffic demand for the calculated paths is obtained from the solution of the proposed LP formulation in this paper. The split ratio is fed to the routers for dividing the traffic of the source-destination pair to multiple paths. Partitioning a traffic demand will be done by adjusting output range of the hashing function [4]. By using this policy in multipath routing, routers should

This work was supported in part by the Brain Korea 21 project of Ministry of Education, in part by the National Research Laboratory project of Ministry of Science and Technology, and in part by National Computerization Agency, 2003, Korea.

provide a flow-level forwarding mechanism not to cause the out-of-order packet delivery problem which will degrade end-to-end performance.

In this paper, we propose the fault-tolerant traffic engineering problem by using the maximally disjoint multipath, in MPLS networks. First, under hop-count and path-count constraints, we find the maximally disjoint multipath that minimizes the maximum of link utilization, while satisfying the given traffic demand of every node pair. A traffic demand is the average traffic volume that needs to be satisfied between two edge routers. Second, we consider the traffic rerouting solution that provides the fault-tolerant routing. We solve this problem through the LP formulation with the constraints which limit the total amount of rerouted traffic. The objective of proposed rerouting problem is to reduce the maximum of link utilization. In this proposed solution, we do not consider the path reconfiguration in which the established paths are torn down and new paths are configured. However, experimental results show that the proposed traffic rerouting solution is nearly same as the global optimal solution through the path reconfiguration.

The remainder of this paper is organized as follows. The related works are introduced in section II. The proposed algorithms based on the LP formulation are explained in section III. For the performance evaluation, simulation results using the case-study network are discussed in section IV, and section V concludes this paper.

II. Related Work

In the [5], the traffic bifurcation LP problem is formulated and heuristic for the non-bifurcation Integer Liner Programming (ILP) problem is proposed. Although the [5] minimizes the maximum of link utilization, it does not consider total network resources and some other constraints. The authors further showed that the traffic bifurcation LP problem can be transformed to the shortest path problem by adjusting link weights in the [8]. In the [7], the dynamic routing algorithm for MPLS networks is proposed, where the path for each request is selected to prevent the interference among paths for the future demands. It considers a traffic non-bifurcation routing and does not include the constraint such as hop-count. In the [9], the authors propose a constrained multipath traffic engineering for MPLS networks. It presents the ILP formulation and heuristic algorithm for multipath traffic engineering problem that includes the constraint such as hop-count and path-count. In the [11], the authors present several topics related to the survivable network. It includes the disjoint shortest path and the maximally disjoint shortest path algorithms in the unidirectional network. In the [12], the authors present the reconfiguration problem of virtual topology in the optical network. When the traffic demand pattern is changed, the improvement of the resource utilization is obtained by reconfiguring the lightpaths. The [4] proposes the hashing-Based Schemes for Internet Load Balancing. It distributes the traffic into multipath by the hashing-schemes using the combination of IP and port number. The several hashing-schemes are presented and evaluated. The [10] proposes an adaptive traffic assignment method to multiple paths with measurement information for load balancing. For differentiated services, finding the traffic split ratios to minimize the end-to-end delay and loss rates is proposed in the [6]. However, how to find appropriate multiple paths is not covered.

III. Fault-tolerant Multipath Traffic Engineering Problem

A. Problem Description

In this section, we present the proposed fault-tolerant multipath traffic engineering algorithm, in MPLS network. We divide this problem into two sub-problems, *Maximally Disjoint Multipath Configuration under Hop-count and Path-count Constraints* and *Traffic Rerouting for Recovery of Link failures*. The former is how to configure the multiple LSPs for maximally keeping the connectivity of each node pair even when some link failures occurr. The latter is, in case of some link failures, how to reroute the disturbed traffic flowing on the failed LSPs into the available paths without configuring the new LSPs. The following two sub-sections show the solutions for each sub-problems by using the following common notations.

- The network is modeled as a directed graph, $G = (V, E)$, where V is the set of nodes and E is the set of links. The capacity of a directed link (i, j) is c_{ij}.
- Each traffic demand d_k($k \in K$, where K is the set of all node pairs in G) is given for a node pair k between an ingress router s_k and an egress router t_k.
- The variable $X_{ij}^k(p)$ represents the fraction of the traffic demand d_k assigned to link (i, j) through the p-th path, $1 \le p \le P_k$, where P_k is the number of given paths between the node pair k. After this point, we simply refer the traffic to relative value from 0 to 1 normalized by the full traffic demand d_k, instead of absolute value.
- The variable α represents the maximum of link utilization.
- The integer variable $Y_{ij}^k(p)$ tells whether link (i, j) is used or not for the path p of the traffic demand k.

B. Maximally Disjoint Multipath Configuration under Hop-count and Path-count Constraints

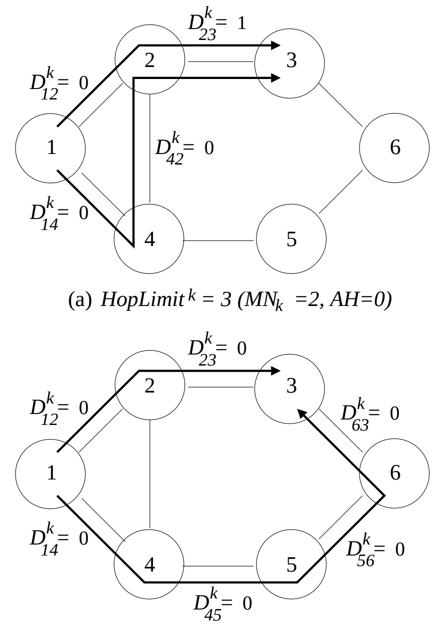
We define the hop-count and path-count constrained maximally disjoint multipath configuration problem by ILP formulation. For each node pair k, there are two constraints, the maximum number of path-count, P_k, and the maximum number of hop-count, $HopLimit_k$ is sum of AH and MH_k, where MH_k is the minimum number of hop counts from s_k to t_k for the node pair k and AH is additional hop-count that is added to MH_k. The ILP formulation is in the Fig. 1.

As shown in the Eq. (1), the primary objective is to minimize the link sharing degree D_{ij}^k of the paths for each node pair, in order to make paths maximally disjoint. The secondary objective is to minimize the maximum of link utilization α with the coefficient c as the weighted factor. The Eq. (2) means that the sum of total outgoing traffic over each path from the source is 1. The Eq. (3) is the flow conservation

$$
\begin{aligned}
&\mathit{Objective}\\
&\mathit{Minimize}\ \alpha + c \cdot \sum_{k\in K}\sum_{(i,j)\in E} D_{ij}^{k} && (1)\\
&\mathit{Subject\ to}\\
&\sum_{p=1}^{P_k}\sum_{j:(i,j)\in E} X_{ij}^{k}(p) = 1, k \in K, i = s_k && (2)\\
&\sum_{j:(i,j)\in E} X_{ij}^{k}(p) - \sum_{j:(j,i)\in E} X_{ji}^{k}(p) = 0\\
&, k \in K, i \neq s_k, t_k, 1 \leq p \leq P_k && (3)\\
&\sum_{j:(i,j)\in E} Y_{ij}^{k}(p) = 1, k \in K, i = s_k, 1 \leq p \leq P_k && (4)\\
&\sum_{j:(i,j)\in E} Y_{ij}^{k}(p) - \sum_{j:(j,i)\in E} Y_{ji}^{k}(p) = 0\\
&, k \in K, i \neq s_k, t_k, 1 \leq p \leq P_k && (5)\\
&\sum_{(i,j)\in E} Y_{ij}^{k}(p) \leq HopLimit^{k}, k \in K, 1 \leq p \leq P_k && (6)\\
&X_{ij}^{k}(p) \leq Y_{ij}^{k}(p), k \in K, (i,j) \in E, 1 \leq p \leq P_k && (7)\\
&\sum_{p=1}^{P_k} Y_{ij}^{k}(p) - 1 \leq D_{ij}^{k}, k \in K, (i,j) \in E && (8)\\
&\sum_{k\in K}\sum_{p=1}^{P_k} d_k X_{ij}^{k}(p) \leq c_{ij}\alpha, (i,j) \in E && (9)\\
&\mathit{Bounds}\\
&X_{ij}^{k}(p), D_{ij}^{k}, \alpha \geq 0, Y_{ij}^{k}(p) \in \{0,1\} && (10)
\end{aligned}
$$

Fig. 1. ILP Formulation of Maximally Disjoint Multipath Configuration under Hop-count and Path-count Constraints

rule of intermediate nodes, which means that for all nodes except source and destination the total amount of incoming traffic to a node is the same as that of outgoing traffic from the node. The Eq. (4) and the Eq. (5) present whether each link (i,j) is used for path p between the node pair k. The Eq. (6) means that the hop-count of path p between the node pair k does not exceed the $HopLimit^k$. The Eq. (7) means non-bifurcation routing of path p between the node pair k. In the Eq. (2) and the Eq. (3), we only state the network flow conservation rule, thus the traffic bifurcation is allowed. However, Eq. the (7) makes the network flow of each path be through the non-bifurcated path. The Eq. (8) means how many paths use this link for node pair k. The Fig. 2 shows the example for variable D_{ij}^k in the Eq (8). The Fig. 2 (a) and (b) present two maximally disjoint path between node 1 and node 3, when the AH is zero and one respectively. For the link $(2,3)$ in the Fig. 2 (a), the D_{23}^k is showed as one which means that a different path is sharing this common link. In this example, as the $HopLimit^k$ is increased, the maximal disjoint degree of paths is also increased. The Eq. (9) means that the maximum of link utilization among all paths for traffic demand k is α. This problem is $NP\!-\!hard$, but does not require the real-time processing. It can be solved by using the ILP solver such as CPLEX, at the initial setup of the network.



(a) *HopLimit* ^{k} *= 3 (MN* $_k$ *=2, AH=0)*

(b) *HopLimit* ^{k} *= 4 (MN* $_k$ *=2, AH=1)*

Fig. 2. Maximally disjoint LSPs as the $HopLimit^k$ is changed

C. Traffic Rerouting for Recovery of Link Failures

Now, we define the fault recovery algorithm for fault-tolerant quality of service (QoS) routing. When some links fail, some LSPs are disconnected, so the traffic on this failed LSPs is required to be rerouted into other available LSPs. We formulate this problem as the following LP with $G' = (V, E')$, where E' is the set of links excluding the failed links from original set E. In this formulation, we consider only traffic rerouting mechanism using the predefined multiple LSPs which is never modified. In addition, the maximum affordable reroutung traffic is given as the constraint, $RerouteLimit$ which is sum of ARR and $MinReroute$. $MinReroute$ means the total traffic flowing on the disconnected paths caused by some link failures. The details are given in the Eq. (21). ARR is the upper bound of the additional rerouted traffic through the node pair which no failure occurs between. The Fig. 3 shows the LP formulation of traffic for recovery of link failures.

As shown in the Eq. (11), the primary objective is to maximize the sum of routing traffic over every node pair by using the rerouting mechanism. The secondary objective is to minimize the maximum of link utilization α with the coefficient c as the weighted factor. The Eq. (12) and the Eq. (13) present the network flow conservation rule. It states not only that the sum of total outgoing traffic of each path from the source is less than or equal to 1 but also that for all nodes except source and destination, the amount of total incoming traffic to a node is the same as that of outgoing traffic from the node. The Eq. (14) means that the maximum of link utilization among all paths for a node pair k is α. The Eq. (15) means that the sum of routing traffic for every traffic demand pair is $load$. In the Eq. (16), $\overline{Y}_{ij}^{k}(p)$ presents the predefined constant which shows whether the link (i,j) is used the path p of a node pair k. It means that there is no path reconfiguration and

$Objective$

$$Minimize\ \alpha - c \cdot load \tag{11}$$

$Subject\ to$

$$\sum_{p=1}^{P_k} \sum_{j:(i,j)\in E'} X_{ij}^k(p) \le 1, k \in K, i \in s_k \tag{12}$$

$$\sum_{j:(i,j)\in E'} X_{ij}^k(p) - \sum_{j:(j,i)\in E'} X_{ji}^k(p) = 0$$

$$, k \in K, i \notin s_k, t_k, 1 \le p \le P_k \tag{13}$$

$$\sum_{k\in K} \sum_{p=1}^{P_k} d_k X_{ij}^k(p) \le c_{ij}\alpha, (i,j) \in E' \tag{14}$$

$$\sum_{k\in K} \sum_{p=1}^{P_k} \sum_{j:(i,j)\in E'} d_k X_{ij}^k(p) = load, i \in s_k \tag{15}$$

$$X_{ij}^k(p) \le \overline{Y}_{ij}^k(p), k \in K, (i,j) \in E', 1 \le p \le P_k \tag{16}$$

$$\sum_{j:(i,j)\in E'} X_{ij}^k(p) - \sum_{j:(i,j)\in E'} \overline{X}_{ij}^k(p) = PE^k(p) - NE^k(p)$$

$$, k \in K, 1 \le p \le P_k, i \in s_k \tag{17}$$

$$\sum_{k\in K} \sum_{p=1}^{P_k} NE^k(p) \le RerouteLimit \tag{18}$$

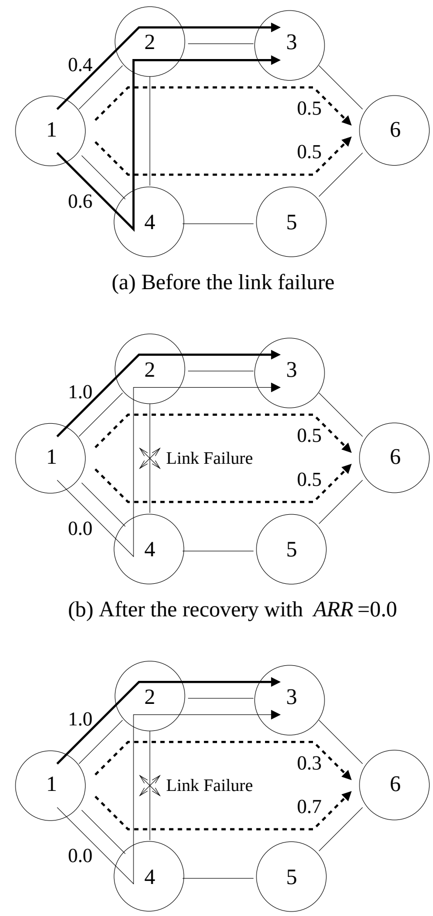
$Bounds$

$$X_{ij}^k(p), RerouteLimit, \alpha, PE_{ij}^k(p), NE_{ij}^k(p) \ge 0 \tag{19}$$

$Constants$

$$\overline{X}_{ij}^k(p), \overline{Y}_{ij}^k(p)\ are\ previous\ setup\ states \tag{20}$$

Fig. 3. LP Formulation of Traffic Rerouting for Recovery of Link Failures



(a) Before the link failure

(b) After the recovery with ARR=0.0

(c) After the recovery with ARR=0.2

Fig. 4. Traffic rerouting for link recovery as the ARR is changed

traffic is only allowed to be routed through predefined paths. In the Eq. (17), w_k is the weighted factor of node pair k for recovery and $\overline{X}_{ij}^k(p)$ is the traffic ratio through the link (i,j) of path p for node pair k before this link failure occurs. The higher weighted factor of some traffic demand has the lower probability to be rerouted for minimizing the α. The variable $PE_{ij}^k(p)$ and $NE_{ij}^k(p)$ mean how much traffic through the link (i,j) is increased or decreased, respectively. The Fig. 4 shows the traffic rerouting mechanism. The Fig. 4 (a) shows the situation before failure occurs and the Fig. 4 (b) shows the situation when link $(4,2)$ fails. In the Fig. 4 (b), for the traffic demand from node 2 to node 3, one path $1 \rightarrow 4 \rightarrow 2 \rightarrow 3$ is disconnected by failure and the traffic through this path is rerouted into another path $1 \rightarrow 2 \rightarrow 3$. In the Fig. 4 (b), the $PE_{12}^k(p)$ and $NE_{12}^k(p)$ of link $(1,2)$ are 0.6 and 0, respectively. Using one of these variables, the Eq. (18) defines the total traffic amount to be rerouted, $RerouteLimit$. When some link failures occur, the traffic through the failed links is rerouted into another connected paths within this limit in order to increase $load$. Rerouted traffic of node pair k usually can be presented as the increased or decreased traffic that flows on paths connecting node pair k. In this paper, the rerouted traffic is only confined to the decreased traffic flowing on the path. Since the traffic does not flow on the failed paths, the minimum rerouted traffic is the sum of traffic flowing on these paths. A constant in the Eq. (21), $MinReroute$, presents this lower-bound that must be rerouted. That is, the left hand side of the Eq. (18) must be larger than $MinReroute$ at least. In the Eq. (20), these constants are predefined value solved by the maximally disjoint multipath formulation in the Fig. 1. This formulation is LP problem and it can be quickly solved by LP solver, when ink failure occurs at the real-time.

$Constants$

$$Fail_p^k = \Sigma \overline{Y}_{uv}^k(p), (u,v) \in FaildLinks$$

$$\sum_{k\in K} \sum_{p=1}^{P_k} \sum_{j:(i,j)\in E} w_k \overline{X}_{ij}^k(p) \times Fail_p^k = MinReroute$$

$$, i \in s_k, Fail_p^k \in Bool \tag{21}$$

This traffic rerouting mechanism has two different policies according to the value of ARR, when $ARR = 0$, when $ARR > 0$. The Fig. 4 (b) and (c) show briefly these two traffic rerouting policies as follows when the link $(4,2)$ fails.

Case 1: **When the** $ARR = 0$

In the Fig. 4 (b), because ARR that presents the additional rerouted traffic amount is set to zero, the rerouted traffic is bounded by $MinReroute$. Since $MinReroute$ is the sum of traffic flowing on the failed paths, the decrease of traffic flowing on the connected paths is not allowed.

Case 2: **When the** $ARR > 0$

As you can see in the Fig. 4 (c), ARR is given as the value larger than zero. Now, it is possible for the traffic is rerouted through path $1 \rightarrow 2 \rightarrow 3 \rightarrow 6$ and $1 \rightarrow 4 \rightarrow 5 \rightarrow 6$, as much as the value of ARR at most. Although paths of node pair from node 1 to 6 never fails, in order to achieve the secondary objective maximizing $load$, the traffic is rerouted satisfying the Eq. (18).

IV. Case Study and Performance Analysis

A. Simulation Environment

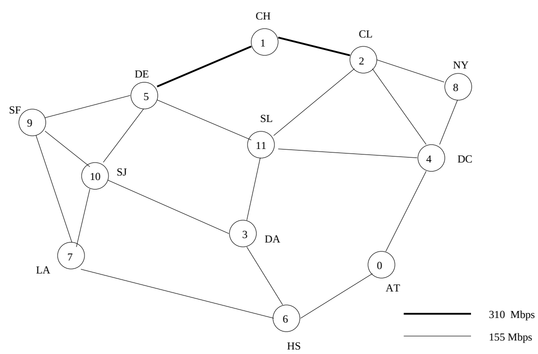


Fig. 5. Abstract US Network

The network topology, in this case study, shown in the Fig. 5 represents the abstract US backbone topology [13]. On this topology, we assume that the background traffic demands are given as in the [13]. Simulation of our proposed algorithms is done in two separate procedures, one is *the maximally disjoint multipath configuration procedure* and the other is *the traffic rerouting procedure* when the link failure occurs. Additional hop-count (AH) is given as 0, 1, 2, 3 and 4. The path-count constraint (P_k) is set to one of 1, 2, 3, 4 and 5. In the *Traffic rerouting procedure*, arbitrary failure of one or two links is assumed. In this topology, the number of distinct failure of one or two links is 38 and 703, respectively.

B. Performance Analysis

The Fig. 6 shows the sum of the link sharing degree ($\sum_{k\in K} \sum_{(i,j)\in E} D_{ij}^k$) when the additional hop-count (AH) and path-count constraint (P_k) is changed. As the number of path-count is increased, the sum of link sharing degree is increased, while as the additional hop-count is increased, the sum of link sharing degree is decreased. As the path-count increases, the sum of the link sharing is largely increased, and as the hop-count increases, the sum of the link sharing is decreased. Since the degree of the node is small, the increase of the number of paths increases the probability to share the same link. In addition, the increase of $HopLimit$ caused by more additional

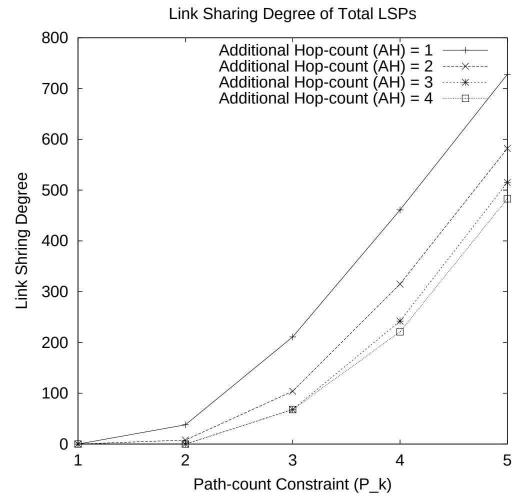
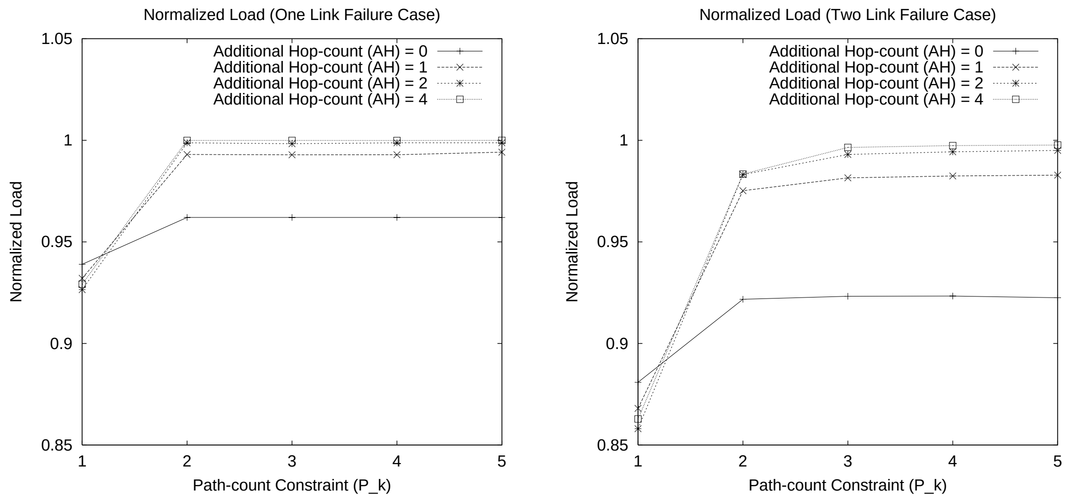


Fig. 6. Link sharing degree

hop-count(AH) lets the total candidate path set be abundant so that we can find more maximally disjoint paths.

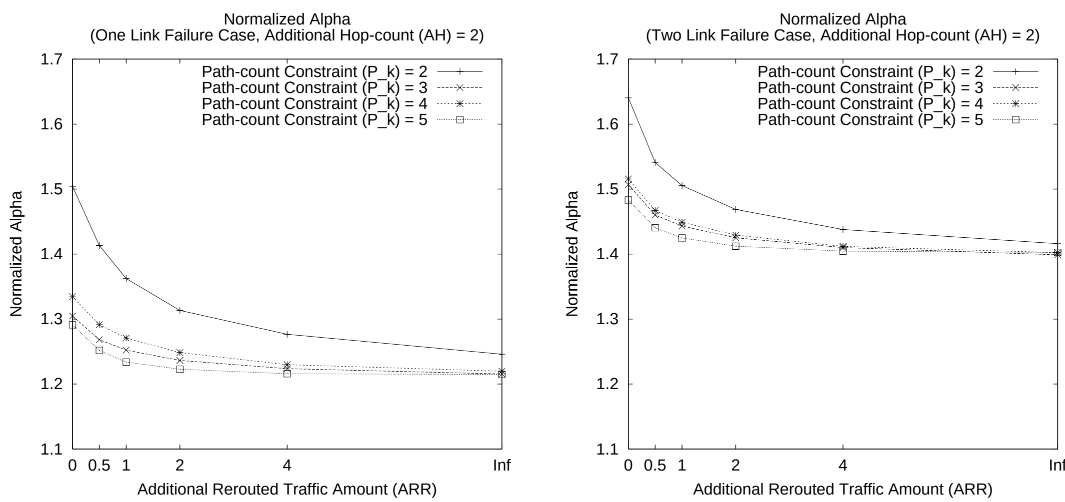
The Fig. 7 shows the normalized load when the failure of one or two links occurs. The normalized load is the quotient of the division of the sum of successfully routed traffic demands after the failure recovery by the sum of the routed traffic demands at no link failure. If the normalized load is 1, the full traffic demand is successfully routed. The weighted factors (w_k) through each node pair k are considered as the same. We averaged 38 simulation results in one link failure case, and 703 in two links failure case. As shown both in Fig. the 7 (a) and Fig. the 7 (b), when the path-count constraint (P_k) is one, the normalized load is very small and too much traffic is rejected by unconnected LSPs for some traffic demand. When two or more path-counts (P_k) are used, the normalized load is increased more than in the case of single path. However, there is only small amount of further increase as the path-count (P_k) increases from 2 to 5. When additional hop-count (AH) is increased from 0 to 1, we can see the drastic increase of the normalized load. It has a close relation with the link sharing degree of paths, in the Fig. 6. As the additional hop-count (AH) is changed from 0 to 1, the link sharing degree of paths also is decreased largely. However, with additional hop-count there is not any substantial further improvement.

In the Fig. 8, the normalized maximum of link utilization (Normalized α) is presented as the additional rerouted traffic amount and path-count constraint are changed. The normalized maximum of link utilization is the quotient of the division of the maximum of link utilization after the failure recovery by the maximum of link utilization at no link failure. Additional rerouting constraint (ARR) controls how much traffic is able to be rerouted to improve the network resource utilization. When this constraint is zero, only the rerouting of the traffic, which was routed through the failed links previously but can not flow due to the failure, is allowed. This constraint set to the infinite means that the rerouting of every traffic demand is allowed. The Fig. 8 shows that the small amount of the additional rerouting traffic is very effective to improve the



(a) One Link Failure Case (b) Two Link Failure Case

Fig. 7. Normalized load comparison on the additional hop-count and the path-count constraint: when one or two link failure is occurred



(a) One Link Failure Case (b) Two Link Failure Case

Fig. 8. Normalized α comparison on the the path-count and the additional rerouted traffic constraint: when one or two link failure is occurred

network resource utilization. When the additional rerouting constraint is increased, the normalized maximum of link utilization is exponentially decreased. When the path-count constraint is increased, the normalized maximum of link utilization is decreased, thus the probability of the network congestion is decreased. Especially it is largely decreased as the path-count is changed from 2 to 3.

We also compare this normalized maximum link utilization with one, which is obtained by allowing reconfiguration of paths themselves. The Table I shows that they are nearly the same.

TABLE I

COMPARISON OF NORMALIZED α

	Rerouting	Reconfiguration
One Link Failure	1.07	1.06
Two Link Failure	1.24	1.23

V. CONCLUSION

In this paper, we propose the fault-tolerant multipath traffic engineering scheme for MPLS networks. The proposed scheme consists of *the maximally disjoint multipath configuration* and *the traffic rerouting mechanism for fault recovery*. The initial configuration of maximally disjoint multipath is formulated as ILP and the traffic rerouting mechanism form fault recovery is formulated as LP. Thus, after the exact optimal initial setup, whenever link failure occurs, the traffic can be rerouted in polynomial time. Through the case study, we show that the traffic engineering using the maximally disjoint multipaths recovers effectively when some link failures occur. Also, if the $ReroutingLimit$ constraint value is appropriately selected, the network resource utilization is more improved. It was shown that, even without the path reconfiguration for traffic demand, the maximum of link utilization can be decreased enough. Therefore, the proposed traffic engineering scheme is practical and useful for providing the fault-tolerant traffic engineering in MPLS networks.

References

[1] E. Rosen, A. Viswanathan, and R. Callon, ”Multiprotocol Label Switching Architecture,” Internet RFC3031, 2001

[2] J. Moy, ”OSPF Version 2,” Internet RFC2328, 1998

[3] R. Callon, ”Use of OSI IS-IS for Routing in TCP/IP and Dual Environments,” Internet RFC1195, 1990

[4] Z. Cao, Z. Wang and E. Zegura, ”Performance of Hashing-Based Schemes for Internet Load Balancing,” INFOCOM’2000

[5] Y. Wang, and Z. Wang, ”Explicit Routing Algorithms for Internet Traffic Engineering,” ICCCN’99

[6] E. Dinan, D. O. Awduche, and B. Jabbari, ”Analytical Framework for Dynamic Traffic Partitioning in MPLS Networks,” ICC’2000

[7] M. Kodialam, and T. V. Lakshman, ”Minimum Interference Routing with Applications to MPLS Traffic Engineering,” INFOCOM’2000

[8] Z. Wang, Y. Wang, and L. Zhang, ”Internet Traffic Engineering without Full Mesh Overlaying,” INFOCOM’2001

[9] Y. Seok, Y. Lee, Y. Choi, and C. Kim, ”Dynamic Constrained Multipath Routing for MPLS Networks,” ICCCN’2001

[10] A. Elwalid, C. Jin, S. Low, and I. Widjaja, ”MATE: MPLS Adaptive Traffic Engineering,” INFOCOM’2001

[11] Ramesh Bhandari, Survivable Networks: Algorithms for Diverse Routing, Kluwer Int’l Series in Engineering and CS, 1999

[12] Biswanath Mukherjee, Optical Communication Networks, McGraw-Hill, 1997

[13] Optimized Multipath, http://www.fictitious.org/omp