PAPER

Traffic Engineering with Constrained Multipath Routing in MPLS Networks*

Youngseok LEE†, Yongho SEOK††, *and* Yanghee CHOI††,

SUMMARY A traffic engineering problem in a network consists of setting up paths between the edge nodes of the network to meet traffic demands while optimizing network performance. It is known that total traffic throughput in a network, or resource utilization, can be maximized if a traffic demand is split over multiple paths. However, the problem formulation and practical algorithms, which calculate the paths and the load-splitting ratios by taking bandwidth, the route constraints or policies into consideration, have not been much touched. In this paper, we formulate the constrained multipath-routing problems with the objective of minimizing the maximum of link utilization, while satisfying bandwidth, the maximum hop count, and the not-preferred node/link list in Linear Programming (LP). Optimal solutions of paths and load-splitting ratios found by an LP solver are shown to be superior to the conventional shortest path algorithm in terms of maximum link utilization, total traffic volume, and number of required paths. Then, we propose a heuristic algorithm with low computational complexity that finds near optimal paths and load-splitting ratios satisfying the given constraints. The proposed algorithm is applied to Multi-Protocol Label Switching (MPLS) that can permit explicit path setup, and it is tested in a fictitious backbone network. The experiment results show that the heuristic algorithm finds near optimal solutions.
key words: *Traffic engineering, multipath routing, load balancing, optimization, LP, MPLS*

1. Introduction

1.1 Traffic Engineering

Traffic engineering (TE) [1] is one of the most important instruments employed by Internet Service Providers (ISPs) to respond to the explosive Internet growth. Internet traffic engineering addresses the aspect of network engineering that includes the issue of performance optimization and fault-tolerant services. Specifically, it consists of the measurement, modeling, characterization, and control of Internet traffic. Its applications include the reliable movement of traffic through the network, the efficient utilization of resources, and the planning of capacities. Therefore, both service providers and end users benefit from the application of Internet traffic engineering.

Traditionally, effective traffic engineering is difficult in public IP networks, because of the limited functional capabilities of the conventional IP technologies. First, limitations of the intra-domain routing protocol control functions used within an Autonomous System (AS) are issues with the IP network system. The Interior Gateway Protocols (IGPs) such as Routing Information Protocol (RIP) [2], Open Shortest Path First (OSPF) [3], and Intermediate System to Intermediate System (IS-IS) [4] depend on the shortest path routing algorithm and the synchronized routing area link state database. Although the shortest path algorithm is simple and easy to be implemented, it may not efficiently utilize the network resources. For example, connections along a congested path will experience severe packet delays or losses, though there exist alternative unloaded paths between the source and the destination. Moreover, although IGP has a highly distributed and scalable architecture, it does not usually consider characteristics of the network state, such as, offered traffic load and available link capacities when making a routing decision. Therefore, the IGP may not efficiently utilize network resources without finding the optimal link metrics, whose calculation may be difficult in a large network topology and dynamic traffic demand [5].

Recent developments in MPLS [6] offer the possibility of reducing the limitations of the IP network system with respect to traffic engineering. In MPLS networks, the ingress router encapsulates IP packets with labels which are used to forward IP packets along the LSP. Before mapping packets to a Label Switched Path (LSP), LSPs between an ingress router and an egress router are set up by a signaling protocol such as Label Distribution Protocol (LDP) or the extended Resource ReSerVation Protocol for LSP tunnels (RSVP-TE). In addition, LSPs can be explicitly established along the specific paths. This means that it is possible to specify all the intermediate nodes between the ingress and the egress routers. Moreover, traditional optimal routing [7], which permits traffic bifurcation to multiple paths, compensates for the disadvantage of the single shortest path routing by load balancing. Particularly, a traffic-engineering problem in the Internet regarding performance optimization has a focus on setting up paths between the edge routers in a network to meet traffic demands while achieving low congestion and optimizing the utilization of network resources. In practice, the typical key objective of traffic engineering is to minimize the utilization of the most heavily used link in the network (the maximum of link utilization) or the average number of packets.

Manuscript received August 14, 2003.
Manuscript revised October 30, 2003.
Final manuscript received December , 2003.
†Dept. of Computer Science and Engineering, Chungnam National University
††School of Computer Science and Engineering, Seoul National University
*This work was supported in part by the Brain Korea 21 project of the Ministry of Education, in part by the National Research Laboratory project of the Ministry of Science and Technology, 2003, Korea.

1.2 Multipath Routing

It is known that the problem of minimizing the maximum of link utilization can be solved by the multi-commodity network flow formulation of optimal multipath routing, which leads to splitting traffic over multiple paths between source-destination pairs [8]–[10]. As multipath routing provides increased bandwidth, the network resources are more efficiently used than in the case of the single shortest-path routing algorithm. Multipath routing has been incorporated in recently developed or proposed routing protocols. The easiest extension to multipath routing is to use the equal-cost multiple shortest paths when calculating the shortest path, which is known as Equal-Cost Multi-Path (ECMP) routing [3]. This is explicitly supported by several routing protocols such as OSPF and IS-IS. Some router implementations allow equal-cost multipath with RIP and other routing protocols [11]. In MPLS networks, multipath routing can be easily supported to forward packets belonging to the same "Forwarding Equivalent Class (FEC)" by dispersing traffic over multiple LSPs, which are established over (disjoint) paths between the same pair of ingress and egress routers [12]. In order to prevent a packet reordering problem in multipath routing, the operation of dividing flows belonging to the same destination or FEC with the hashing table at the ingress router in the MPLS network can be easily implemented [13]. Although packet-level multipath forwarding divides traffic more accurately, flow-level multipath forwarding has been employed to prevent out-of-order packet delivery problems that will degrade end-to-end performance, especially for TCP (Transmission Control Protocol) connections. Therefore, optimal routing is useful for traffic engineering without the significantly increased overhead of introducing multipath packet forwarding function and routing table maintenance cost [14].

1.3 Motivation of This Work

The traditional research [7]–[9] on how to calculate the paths and the load-splitting ratios has proliferated in order to minimize end-to-end delay. However, it is only recent that the route or policy related constraints have been considered for multipath routing problems of traffic engineering [14]. Specifically, multipath routing algorithms considering constraints such as bandwidth, the maximum allowed number of hops and link/node affinity are the constraints necessary for efficient traffic engineering [1], [15].

The load-splitting ratio is fed to the routers for dividing the traffic demand of the same source-destination pair to the multiple paths or the same FEC (Fig. 1). When defining the problems and calculating the paths and the load-splitting ratios for traffic demand or bandwidth, we also consider constraints such as the maximum hop-count allowed for paths and the not-preferred node or link list. The maximum hop-count constraint is for reducing the delay and total traffic volume. On the other hand, the node or link affinity constraint, such as excluding not-preferred nodes/links when calculating paths between a node pair, is valuable in reflecting routing policies. The node/link affinity constraint is a subset of the resource class affinity in MPLS networks [1]. When nodes or links have resource class attributes or colors, the network administrator may enforce paths for a traffic demand to be established, not traversing specific colored links or nodes.

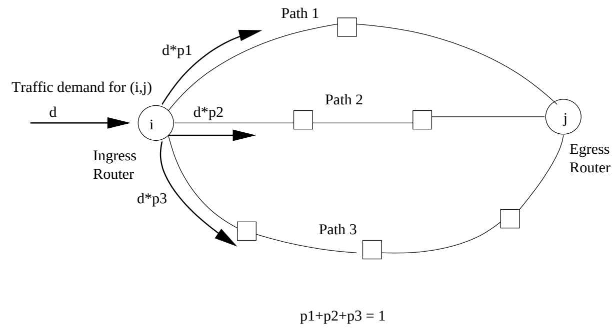


Fig. 1 Multiple paths with load-splitting ratios.

Hence, in this paper, we define the constrained multipath routing problems under the given traffic demand matrix in LP in order to get optimal solutions. Then, we propose practical heuristic algorithms in polynomial time complexity that find near optimal paths and their load-splitting ratios satisfying the given traffic demand and constraints.

The contributions of our work are summarized as follows.

- **Global optimization approach to mulitpath routing with the constraints:** Finding the optimal paths and load-splitting ratios of the constrained multipath problems by LP solvers such as CPLEX [17] carries out the computation in three steps. First, a traffic demand is split over the network links so that the maximum link utilization in the network is minimized while a traffic demand is satisfied and other constraints are observed. Although traffic bifurcation is allowed, the total network resources consumed for assigning traffic demands on links should be minimized in order to accept more connection requests in the future. Thus, the second step consists of finding a rearranged traffic split pattern under the constraint of the maximum of link utilization obtained in the first round, one that minimizes the total network resources used for assigning traffic demands on links (i.e., removing inefficiently assigned traffic demands over all the links). After the two steps, finding paths from the split pattern is easily performed by using the maximum flow path algorithm for each traffic demand [18]. The LSPs calculated in the offline manner under the given traffic demand are passed to the ingress routers for LSP setup, thereby the non-shortest paths will be efficiently utilized in multipath routing.
- **Heuristic of constrained multipath routing for large**

networks with dynamic traffic demands: Although the LP formulation gives the optimal solution which is calculated by a central TE server, its computation complexity is too large to be suitable for the fast computation and dynamic connection requests, which will not be scalable in the large network environment. Therefore, we propose a simple heuristic approach based on the well-known M shortest path algorithm [19], which is useful for the on-line routing application for LSP connection requests in large networks.

The remainder of this paper is organized as follows. The related works are introduced in section 2. In section 3 the general traffic bifurcation problem and the hop-count constrained traffic bifurcation problem with the node/link affinity condition are given. The heuristic algorithms are explained in section 4. The results of the performance evaluation by simulation are discussed in section 5, and section 6 concludes this paper.

2. Related Work

In connection-oriented networks, [20] has analyzed the performance of multipath routing algorithms and has shown that the connection establishment time for reservation is significantly lowered in the multipath case. They didn't, however, fully consider the path computation problem. [21] has proposed a dynamic multipath routing algorithm in connection-oriented networks, where the shortest path is used under light traffic condition and multiple paths are utilized as the shortest path becomes congested. In this work, only connection or call-level, not flow-level, routing and forwarding are considered. In [22], Quality-of-Service (QoS) routing via multiple paths under a time constraint is proposed when the bandwidth can be reserved, assuming all the reordered packets are recovered by optimal buffering at the receiver. This scheme has much overhead for dynamic buffer adjustment at the receiver. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed in [23], and it is shown that congestion can be greatly reduced. In [24], it is shown that the quality of services can be enhanced by dividing the transport-level flows into UDP and TCP flows. These works did not consider the path calculation problem.

For the MPLS network, a traffic engineering method using multiple multipoint-to-point LSPs is proposed in [12], where backup routes are used against failures. Hence, the alternative paths are used only when primary routes do not work. In [10], the traffic bifurcation problem is formulated in LP and heuristics for the non-bifurcating problem are proposed. Although [10] minimizes the maximum of link utilization, it does not consider total network resources and constraints. Wang and et al. have showed that the traffic bifurcation problem in LP can be transformed to the shortest path problem by adjusting link weights in [25]. In [26], 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 only single path routing for simplicity and does not include the constraint. [27] proposes a traffic partitioning mechanism for differentiation in MPLS networks, where an expedited forwarding (EF) traffic demand is divided into multiple LSPs to minimize the average end-to-end delay. However, it needs recalculation of load-splitting ratios whenever new requests arrive or the average rate of requests changes.

3. Problem Formulation of Constrained Multipath Routing

In this section, we propose three problem formulations: the first one in Sec. 3.2 is a slightly modified problem formulation which can prevent the inefficient or cyclic flow assignment; the second one in Sec. 3.3 is a new problem formulation by which each path satisfies the maximum hop count in multipath routing; and the last one in Sec. 3.4 is a policy routing-related constraint to problems in Sec. 3.2 and Sec. 3.3.

3.1 Assumption

In this paper, a traffic demand refers to the measured or estimated average traffic volume in bps between an ingress and an egress router, or customers' bandwidth requests for a connection in case of Virtual Private Network (VPN), which is expressed in the matrix form. It is assumed that even though a traffic demand varies largely at the nodes near the end users, it is quite stable for the backbone network with aggregated traffic [28]. Traffic engineering is performed on a set of newly arrived LSP requests for the existing network, or periodically (weekly or monthly) on all the established LSPs for the known traffic demand. The results of the computation are paths to set up as LSPs and load-splitting ratios among the paths. The traffic bifurcation function for the given load-splitting ratio is handled by multipath forwarding MPLS routers, and multiple LSPs between an ingress router and an egress router are explicitly established by a signaling protocol. The shortest path represents the minimum hop one.

3.2 General Traffic Bifurcation Problem

The general traffic bifurcation (TB) problem consists of finding multiple paths carrying a part of or all of a traffic demand between an ingress and an egress node which minimizes the maximum of link utilization, α. It is shown that it makes little difference whether the cost function of the maximum of link utilization or that of the average number of packets is used for routing optimization [7]. As the maximum link utilization qualitatively expresses that congestion sets in when link utilization increases higher, it is important to minimize the link utilization throughout the network so that no bottleneck link exists. In addition, when solving the TB problem, we minimize the total network resources allocated for the request by using the secondary optimization objective, which leads to reduce the end-to-end delay as well as the available bandwidth in the network.

Following variables and constants are defined for LP formulation.

- $G = (N, E)$: The network is modeled as a directed graph, $G = (N, E)$, where N is the set of ingress, transit and egress routers, and E represents the set of links. The capacity of a directional link $(i, j) \in E$ is c_{ij}.
- K : A set of all LSP requests. An LSP connection request from an ingress router to an egress router is denoted as k.
- d_k : The amount of a traffic demand for the LSP connection request, k.
- s_k : The ingress of an LSP connection request, k. The egress router for the LSP connection request is denoted by t_k.
- X_{ij}^k : This variable represents the fraction of a traffic demand assigned to link (i, j) for the LSP connection request, k.

The LP formulation of TB is stated as follows in [10].

$$Minimize\ \alpha$$
subject to

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(i,j)\in E} X_{ji}^k = 0, k \in K, i \neq s_k, t_k \quad (1)$$

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(i,j)\in E} X_{ji}^k = 1, k \in K, i = s_k \quad (2)$$

$$\sum_{j:(i,j)\in E} X_{ij}^k - \sum_{j:(i,j)\in E} X_{ji}^k = -1, k \in K, i = t_k \quad (3)$$

$$\sum_{k\in K} d_k X_{ij}^k \leq c_{ij}\alpha, \forall (i,j) \in E \quad (4)$$

$$0 \leq X_{ij}^k \leq 1, 0 \leq \alpha$$

The objective is to minimize α. Constraint (1), constraint (2) and constraint (3) represent the flow conservation constraints for intermediate, source and sink nodes, respectively. Constraint (4) is the link capacity constraint which means that the amount of all the flow assignments on a specific link should be less than the capacity scaled by α, where α states that the ratio of flow assignment to the scaled link capacity (=1). The LP formulation of the TB problem can be solved with the classic Simplex method using an LP solver such as CPLEX, and the solution gives the optimal flow values (X_{ij}^k).

The above TB problem finds the optimal α and identifies the bottleneck link [10]. However, traffic in the network may still be reduced, because many flow assignment candidates satisfying α exist. Moreover, unnecessarily long paths or cycles may occur, because the objective is to minimize not the total network resources but the maximum of link utilization. Therefore, in this paper, we tackle the above TB problem through a two-step approach with different optimization objectives as follows.

Step 1: Optimize TB with the objective of minimizing the maximum link load.

Step 2: Optimize TB with the objective of minimizing the total network resources used for carrying traffic demands, while the maximum link load found in Step 1 is given as a constant constraint.

The maximum of link utilization found in Step 1 is denoted as $\bar{\alpha}$. Then, in step 2, we solve the multi-commodity problem, TB, with the following objective (5) in order to minimize the total network resources used (i.e., summation of flows assigned to each link) and to get rid of cycles and unnecessary long paths.

$$Minimize \quad \sum_{(i,j)\in E} \sum_{k} d_k \cdot X_{ij}^k \quad (5)$$

Finally, we use the shortest augmenting algorithm for the maximum flow problem [18] to derive multiple LSPs and load-splitting ratios over paths from $\{X_{ij}^k : X_{ij}^k > 0\}$. The shortest augmenting algorithm is performed on a temporary graph consisting of links whose $\{(i,j) : X_{ij}^k > 0\}$ for each pair of ingress and egress routers. On the temporary graph, the traffic demand is assigned along the shortest path with the found load-splitting ratio. Multiple paths which may be overlapped between intermediate nodes can be found through this step.

3.3 Maximum Hop-count Constrained Traffic Bifurcation Problem

The maximum hop-count constrained traffic bifurcation problem (denoted by $HTB(H)$) consists of finding hop-count constrained multiple paths between a source-destination pair with the objective of minimizing α. H denotes the additional hop-count constraint compared to the shortest path. The maximum hop-count allowed for LSPs of each traffic demand is given as L_k†. Due to the maximum hop-count constraint, more candidate paths will be available for load balancing.

Although the maximum hop-count constrained non-bifurcation problem can easily be formulated as the LP problem by imposing $\sum_{(i,j)\in E} X_{ij}^k \leq L_k, (\{X_{ij}^k\} = 0, 1)$ [29], a different problem formulation is necessary for the bifurcation case to simultaneously consider multiple paths. Therefore, we formulate the $HTB(H)$ problem in LP by using the hop-level flow conservation rule: the sum of incoming flows to a node i reached by the source with l hops equals the sum of outgoing flows from the node i to adjacent nodes reached by the source with $(l + 1)$ hops.

The same network model and traffic demands are assumed as in the previous TB problem. Let X_{ij}^{kl} represent the fraction of a traffic demand, d_k, assigned on a link (i, j)

† $L_k = L_{SP(k)} + H$, $L_{SP(k)}$ is the hop-count of the shortest path for each LSP request, k.

where the node j is far from s_k in l hops. X_{ij}^{kl} means the amount of flow for the LSP connection request, k, allocated on the link (i, j) in l hops from source k as illustrated in Fig. 2.

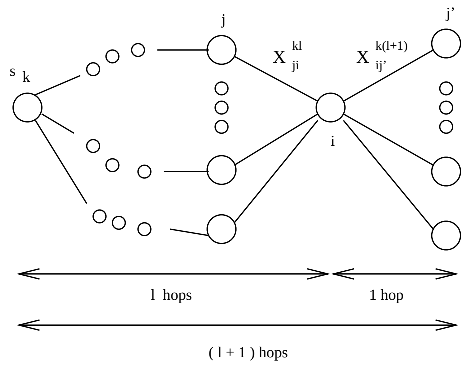


Fig. 2 The X_{ij}^{kl} variable for $HTB(H)$ problem formulation.

The LP formulation for the hop-count constrained traffic bifurcation problem is given as follows.

$$Minimize\ \alpha$$
subject to

$$\sum_{j:(i,j)\in E} X_{ij}^{kl} = \begin{cases} 1, & k \in K, i = s_k, l = 1 \\ 0, & k \in K, i \neq s_k, l = 1 \end{cases} \quad (6)$$

$$\sum_{j:(i,j)\in E} X_{ij}^{k(l+1)} - \sum_{j:(j,i)\in E} X_{ji}^{kl} = 0, \quad k \in K, i \neq s_k, t_k, 1 \leq l < L_k \quad (7)$$

$$X_{ij}^{kl} = 0, l > L_k \quad (8)$$

$$\sum_{l=1}^{L_k} \sum_{j:(j,i)\in E} X_{ji}^{kl} = 1, k \in K, i = t_k \quad (9)$$

$$\sum_{l=1}^{L_k} \sum_{k\in K} d_k X_{ij}^{kl} \leq c_{ij}\alpha, \forall (i,j) \in E \quad (10)$$

$$0 \leq X_{ij}^{kl} \leq 1, 0 \leq \alpha$$

The objective is to minimize α. Constraint (6) says that the total outgoing traffic over the first hop from the source is 1. In case of non-source nodes, the outgoing flow at the first hop should be zero, which prevents abnormal flow assignment of non-source nodes at the first hop during optimization. Constraint (7) and (8) are the hop-level flow constraint, which means that the sum of incoming flows toward a node i with l hops equals that of outgoing flows from that node with $(l + 1)$ hops. Constraint (9) is the flow constraint for the destination node. Constraint (10) is the link capacity constraint.

In $HTB(H)$, if H is zero, the traffic demand, d_k, will be assigned to the single shortest path or multiple equal-cost shortest paths. In the case of multiple shortest paths, traffic will be optimally bifurcated, whereas traditional ECMP routing will divide traffic evenly into multiple paths by 1/(the number of paths).

In the same way as TB, after the maximum of link utilization value ($\bar{\alpha}$) is found by the above LP formulation, we solve the multi-commodity LP problem with $\bar{\alpha}$ in order to minimize total network resources and to remove cycles. Then, multiple paths and load-splitting ratios are retrieved by the shortest augmenting algorithm.

3.4 Excluding Node or Link Constraint

In addition to the maximum number of the hop-count constraint, the excluding node or link constraint/links for a traffic demand, can be given according to the administrative policy.

The excluding node constraint (E_N^k) for the LSP request, k, is formulated as follows and may be added to the TB or $HTB(H)$ problems.

$$\sum_{j:(i,j)\in E} X_{ij}^{kl} = 0, \forall i \in E_N^k, \forall l \quad (11)$$

$$\sum_{j:(j,i)\in E} X_{ji}^{kl} = 0, \forall i \in E_N^k, \forall l \quad (12)$$

Similarly, the link exclusion constraint (E_L^k) for the LSP request, k, may be added.

$$X_{ij}^{kl} = 0, \forall (i,j) \in E_L^k, \forall l \quad (13)$$

4. The Heuristic Approach

Although the LP formulations give the optimal solution, they are not scalable when the network size grows or when the LSP connection requests are dynamic. Hence, we propose heuristic algorithms of $HTB(H)$ to find constrained multiple paths and their load-splitting ratios for each LSP connection request on the ingress and egress pair in one-by-one [30]. The proposed algorithms consist of four parts: 1) selecting one of LSP connection requests, and removing the not-preferred nodes/links; 2) modifying the original graph to the hop-count constrained one; 3) finding multiple paths; and 4) calculating the load-splitting ratios for multiple paths.

Step 1 : **Preprocessing**
When several LSP requests should be simultaneously served, they will be sorted according to the given priority or the administrative policy such as the maximum of the traffic demand. For each traffic demand, the specified nodes or links to be excluded will be deleted.

Step 2 : **Hop-count constrained graph conversion**
The given network, $G = (N, E)$, is converted to a hop-count ($L_k = L_{SP(k)} + H$) constrained graph, $G' = (N', E')$, where N' and E' are transformed as follows.

$$N_0 = \{s_k\}$$

$$N_k = \{j|(i,j) \in E, i \in N_{k-1}\}$$
$$N' = \cup_{0 \le k \le L_k} N_k$$
$$E_1 = \{(s_k, i)|(s_k, i) \in E\}$$
$$E_k = \{(i,j)|i \in N_{k-1}, j \in N_k\}$$
$$E' = \cup_{1 \le k \le L_k} E_k$$

The converted node set, N', consists of feasible nodes, N_k, with k hops adjacent to nodes with $(k-1)$ hops from the source node, s_k. Similarly, the converted link set, E', is composed of redundant links between N_k and N_{k-1} nodes.
An example of graph conversion is given in Fig. 3. Figure 3-(a) represents the original network topology. When an LSP connection request from node 1 to node 4 which requires bandwidth of 3 Mbps with the hop-count constraint of one additional hop arrives, the graph in Fig. 3-(b) is derived after adding redundant nodes and links. If two parallel paths are permitted for the request, it is easily seen that any path traversed from node 1 to node 4 in Fig. 3-(b) does not exceed three hop counts.

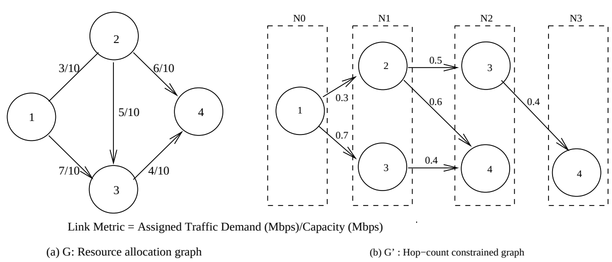


Fig. 3 Topology conversion example.

Step 3 : Finding M multiple paths

On the modified graph G', the link weight (w_{ij}) is defined by the current link utilization ratio (allocated traffic demand / link capacity). We propose two ways of choosing M multiple paths on the modified graph, $G' = (N', E')$. M multiple paths are found by the well-known M shortest path algorithm which always calculates M shortest paths that may include both equal cost shortest paths and longer paths. When finding multiple paths, paths are selected 1) to minimize the sum of the link utilization (M shortest paths), or 2) to minimize the maximum of link utilization (M widest paths).

- M shortest paths : $HST_{SP(M)}(H)$
  The M shortest paths are retrieved by selecting the adjacent node with the minimum sum of link metrics. The cost of a node j reached from source is denoted by $dist(j)$ which is 0 for source node.

$$dist(j) \leftarrow min_{i \in S}(dist(j), dist(i) + w_{ij});$$

  , where S is the set of nodes whose shortest path from source is already determined.

- M widest paths : $HST_{WP(M)}(H)$
  The M widest paths are found by choosing paths with the maximum reservable bandwidth among all the paths from source to sink, which will minimize the utilization of the bottleneck link. The maximum function of $max(dist(i), w_{ij})$ is applied only nodes whose $dist(i)$ is not infinity.

$$dist(j) \leftarrow min_{i \in S}(dist(j), max(dist(i), w_{ij}));$$

Step 4 : Calculating load-splitting ratios

After finding M multiple paths through the previous step, the amount of a traffic demand, d_k, is divided to M paths. If the maximum of link utilization on M paths (α_M) is less than α of the current bottleneck link, splitting the traffic demand, d_k, is performed in order to minimize the amount of used total resources (i.e., the number of links and routers). Then, a path with the smallest hop count is selected among the found M paths, and the traffic demand is assigned to the path until the maximum of link utilization of the path does not exceed α. This step is terminated when there remains no traffic demand or there exists no available path. Still, if the traffic demand is not fully assigned, it is allocated on M multiple paths in proportional to the sum of the link utilization of each path.

The detailed algorithm is explained in Fig. 4.

Heuristic : *Finding constrained multiple paths and their load-splitting ratios*
- Set α to be current maximum link utilization in the network;
- d_k is the requested traffic demand from s_k to t_k;
- Convert G to the L_k-hop constrained graph, G';
- Find a set of M shortest (or widest) paths from s_k to t_k, $P = p_i, i = 1, ..., M$;
- Set α_M to be the maximum link utilization of P;

if ($\alpha_M < \alpha$)
 while($d_k > 0$ and P is not empty)
  - Set p to be the minimum hop path in P;
  - Assign d_k to the path, p, until α_M is less than α;
  - Delete p from P;
  - Update d_k and α_M;

 endwhile
endif
while($d_k > 0$)
  - Assign remaining d_k to M paths in proportion to the available link capacity;

endwhile

Fig. 4 The proposed heuristics ($HST_{SP(M)}(H)$, or $HST_{WP(M)}(H)$).

Therefore, the proposed heuristic is either the shortest path based algorithm, $HST_{SP(M)}(H)$, or the widest path based one, $HST_{WP(M)}(H)$. Figure 5 explains the results of the multiple paths and their load-splitting ratios on the graph in Fig. 3-(a). It is seen that the paths found by the single shortest path algorithm wastes more network resources, while the proposed algorithm can find the optimal solution.

The proposed algorithm consists of three parts: preprocessing the given graph; finding multiple paths; and calculating load-splitting ratios. For each part, the time complexity is bound as follow. First, for the graph conversion, the

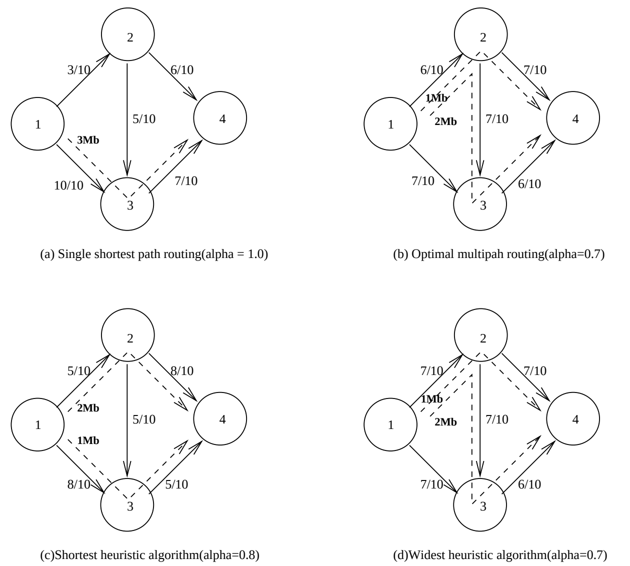


Fig. 5 The result of several multiple path calculation methods ($M = 2$).

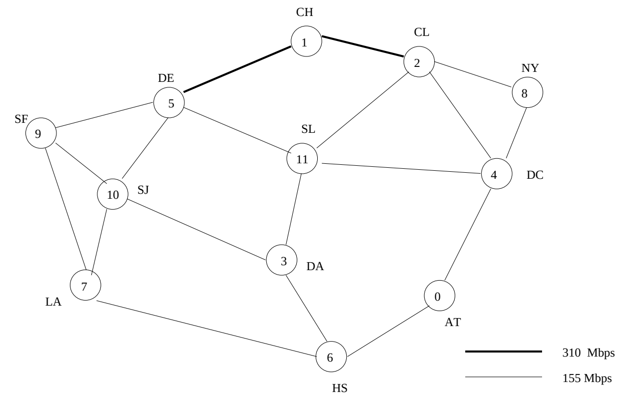


Fig. 6 Abstract US network topology for the experiments.

computation complexity will be $O(|m+n|)$, where m is the number of edges and n is the number of nodes in the graph. Second, for the M shortest simple paths problem, the best known bound for the ordered set is $O(|m + nlogn + M)|)$ in a directed graph [31], where M is the number of paths. Finally, the algorithm for splitting a traffic demand into M paths is bound by $O(|MlogM + 2M|)$, because sorting M paths according to hop counts is bound by $O(|MlogM|)$ and calculating the load-splitting ratio for each path is bound by $O(|2M|)$. Hence, the time complexity of the proposed algorithm is bound by $O(|m + nlogn + M + MlogM|)$.

5. Performance Evaluation

5.1 Simulation Environment

The network topology shown in Fig. 6 represents the abstract US backbone topology [32]. Also, a traffic demand matrix given in [32] is used. In this network condition, first, the results of TB and $HTB(H)$ problems solved with CPLEX are explained. Then, we compare the performance results of the proposed heuristic algorithm with those of the shortest path algorithm and $HTB(H)$. Since we want to compare with the solution of the proposed heuristic with the lower bound of the LP solution, the simulation was performed on the 12-node network topology where the LP-formulated problem can be solved within reasonable time by CPLEX. As the network size becomes large, it is more difficult to acquire the optimal solution by LP.

5.2 Comparison of Optimal Solutions

First, we investigated the performance results of the general traffic bifurcation method by solving the LP problems. The different traffic-engineering methods given below are compared to each other in terms of the maximum of link utilization (α), the total network resources ($R = \sum_{(i,j)\in E} d_k \cdot X_{ij}^{k(l)}$), and the number of LSPs (P).

- **Shortest path based non-bifurcation (SP)**: Each traffic demand is assigned along the shortest path from the ingress router to the egress router.
- **ECMP**: When multiple shortest paths exist, a traffic demand is evenly divided.
- **Traffic bifurcation (TB)**: Each traffic demand is assigned by the LP solution of TB.
- **H hop constrained traffic bifurcation ($HTB(H)$)**: All traffic demands are assigned by the LP solution of $HTB(H)$.
- **H hop constrained traffic bifurcation with node affinity ($HTB{-}NA(H)$)**: The node affinity policy for a bifurcated traffic demand is added to $HTB(H)$.

In Table 1 for the abstract US topology, $HTB(1)$, which includes paths one hop longer than the shortest path, finds the same α as TB, because many multiple paths are explored. $HTB(1)$ reduces α by 27.5% over SP. Yet, $HTB(1)$ needs 15 additional LSPs and a 12.9% increase in total network resources used. When the node exclusion constraint† is used, $HTB{-}NA(1)$ decreases α by 21.7% with seven more LSPs, compared to SP. Nonetheless, the total network resources of $HTB{-}NA(1)$ are the same as those of SP. ECMP reduces α by 10.1% compared to SP, but it requires 39.4% more LSPs. It is interesting to note that $HTB(0)$ requires only 4.6% more LSPs than SP, while decreasing α by 22.1%.

In order to examine algorithm sensitivity to variations of a traffic demand, we repeat the simulation with the varied traffic demand. The degree of maximum load variation ($v = \{10\%, 20\%, ..., 100\%\}$) is used to represent the upper bound on the percentage of varying traffic volume over the static traffic demand. Therefore, each traffic demand in the static traffic demand matrix is randomly altered within this range. For one value of v, ten sets of traffic demands are tested and averaged. The normalized $\hat{\alpha}_{TB}$ is defined to compare the

†In the simulation, for instance, LA or SJ nodes are specified not to permit transit traffic from SF.

Table 1 Maximum of link utilization (α), total network resources (R), and number of LSPs (P).

	SP	ECMP	TB	HTB(0)	HTB(1)	HTB-NA(1)
α	1.22	1.10	0.88	0.95	0.88	0.95
R	4211.1	4211.1	4233	4211.1	4755.7	4211.1
P	132	218	153	138	147	139

proposed algorithms (e.g., $HTB(H)$) with TB as follows.

$$\hat{\alpha}_{TB} = \frac{\alpha_{HTB(H)} - \alpha_{TB}}{\alpha_{TB}} \quad (14)$$

In Fig. 7 and 8, the results of the sensitivity experiment are given. In Fig. 7, where different levels of load variations are applied to the fixed paths and load-splitting ratios found at the traffic demand matrix with 0% load variation, it can be seen that the LP problems such as TB, $HTB(H)$, and $HTB-NA(H)$ produces results lower than that of SP and $ECMP$. Especially, $HTB(1)$ performs closely to TB in all levels of load variations, while satisfying the maximum hop-count constraint. Although the node exclusion constraint is added, $HTB-NA(1)$ outperforms $ECMP$. Sometimes, it is necessary that LSPs are reconfigured at minimal cost when the normalized α is greater than the threshold set by the administration policy. For example, the paths and load-splitting ratios found for the specific traffic demand matrix do not give the best solution as traffic changes (e.g., when load variation is 0.7 in Fig. 7). Figure 8 shows how much α (by the LSPs with the load-splitting ratios at 0%) deviates from that of the optimal traffic bifurcation, TB, which computes the paths and the load-splitting ratios whenever the load variation is changed. As the variation grows, the normalized α increases. However, $HTB(1)$ still minimizes α.

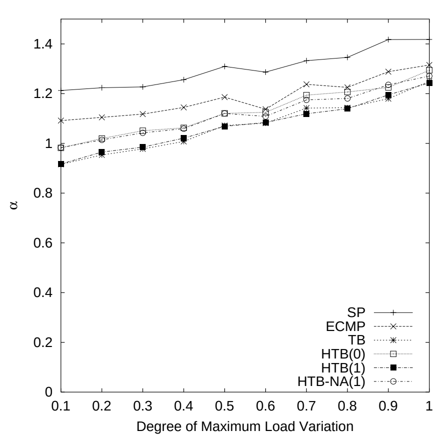


Fig. 7 Sensitivity to load variation when the fixed paths and load-splitting ratios for the initial traffic demand are used.

In this network topology, it is seen that one additional hop-count constraint is enough to find nearly the same α of TB. In addition, when hop-count constrained traffic bifurcation is combined with the appropriate node affinity policy

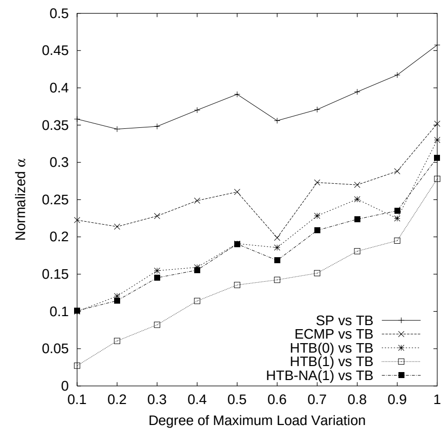


Fig. 8 Sensitivity to load variation when the paths and load-splitting ratios are recalculated whenever traffic demand changes.

($HTB-NA(1)$), the maximum of link utilization can be greatly reduced with a few additional LSPs and network resources. On the other hand, the hop-count constrained traffic bifurcation scheme greatly reduces the LP-solving time, because the size of the search space is limited by the hop-count constraint. With CPLEX for the LP solver†, for example, $HTB(1)$ takes only 8.8 seconds, whereas TB takes 1605.1 seconds. However, the LP-solving approach is not suitable for finding paths in large networks with dynamic LSP setup.

5.3 Results by the Proposed Heuristics

To examine the effectiveness of the proposed heuristic algorithm, we used the dynamic LSP connection requests based on the traffic demand matrix. In the same network topology, we generate ten random LSP connection requests between two nodes selected randomly. Therefore, 1,320 requests are tested in total. The duration of each LSP connection is exponentially distributed (ten seconds), and the inter-arrival time is randomly distributed between zero and one hundred seconds. The average rate of each LSP connection is set to 2 Mbps. The additional hop-count constraint (H) is given as zero or one. When finding multiple paths, we set the number of multiple paths (M) to be one of 1, 3 and 5. The proposed heuristic is compared with the simple shortest path algorithm or the optimal LP solution of $HTB(H)$ which was computed for every changed traffic demand matrix.

Figure 9 shows the normalized α by the optimal α of $HTB(H)$. In Fig. 9-(a), it is seen that although only one additional hop is constrained on the single path, the proposed

†For the LP-solving parameter, the optimal gap is 0.01.

heuristic performs better than the shortest path. When the heuristics search up to three multiple paths (Fig. 9-(b)), the maximum of link utilization is greatly reduced. Also, even when the additional hop-count constraint becomes zero (i.e., the equal cost multiple paths) (Fig. 9-(c)), it is seen that load balancing is well achieved among multiple paths, giving the similar α to the case of the one additional hop.

As shown in Table 2 including the average of α, the shortest path (widest path) based heuristic increases only 4.12 (3.01)% when compared to the optimal solution by LP formulation of $HTB(1)$. Yet, the maximum link utilization by the shortest path algorithm increases by 11.34%.

Table 2 Average of maximum of link utilization (α).

	SP	$HST_{SP(3)}(1)$	$HST_{WP(3)}(1)$	$HTB(1)$
α	1.08	1.01	1.00	0.97

However, the performance of the proposed algorithm may not be more enhanced although the number of hops and paths increases, because many multiple paths are overlapped. In this simulation, although heuristic can search up to five multiple paths (Fig. 10-(a)) or expands its search space by two additional hops (Fig. 10-(b)), the proposed algorithm shows the similar results as one with less hops and paths (e.g., one additional hop and three multiple paths).

In general, when α is greater than 1, the request will be blocked because of scarce network bandwidth assuming that over-booking is not permitted. We plotted the ratio of blocked requests to the total requests in every five seconds which cause α to be greater than 1 in Fig. 11. It is shown that the number of rejected requests is reduced by the proposed heuristic.

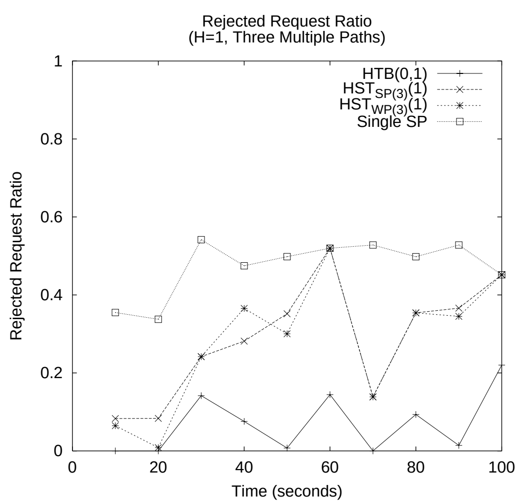


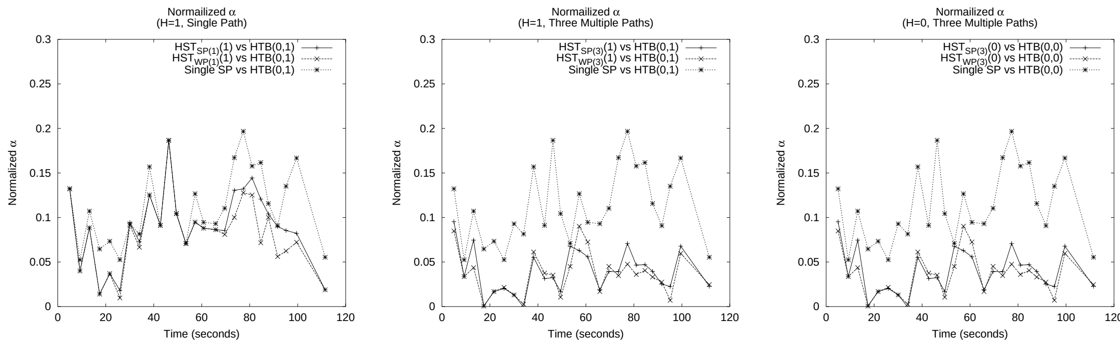
Fig. 11 Rejected traffic demand request ratio.

6. Conclusion

In this paper, we formulate the constrained multipath routing problems in LP and propose heuristic algorithms that find near optimal solution of minimizing the most congested link. First, the general traffic bifurcation problem without constraints formulated as an LP problem (TB) is to minimize α by splitting a traffic demand to multiple LSPs. Second, the maximum hop-count constrained traffic bifurcation problem ($HTB(H)$) finds the LSPs which minimize α while satisfying the given constraints and traffic demands. In addition, the policy-based constraint of excluding nodes/links can be supported in the proposed problem. We propose and verify efficient heuristic algorithms ($HST_{SP(M)}(H), HST_{WP(M)}(H)$) with the polynomial time complexity for the LP problems. It is shown from the simulation results that the proposed heuristic algorithm finds near optimal solution of the maximum of link utilization without the heavy complexity of computation. Therefore, the proposed algorithm will be practical and useful for reducing the probability of congestion by minimizing the utilization of the most heavily used link in the network.

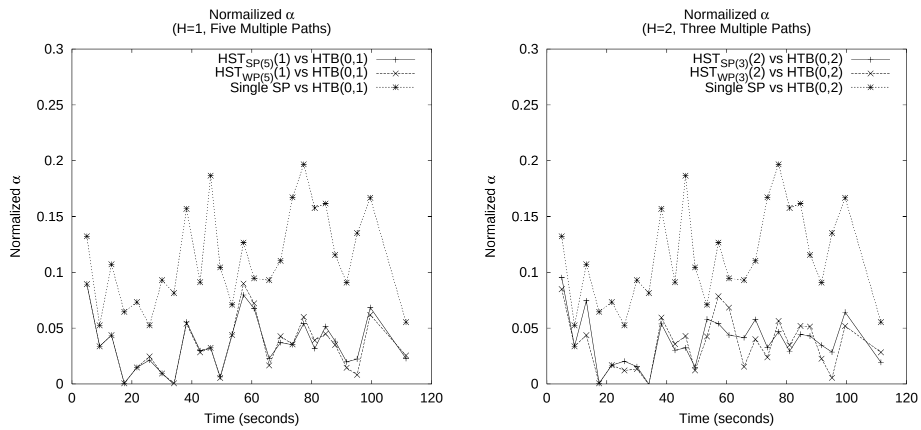
References

[1] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus, "Requirements for Traffic Engineering Over MPLS," IETF RFC2702, 1999.
[2] C. Hendrick, "Routing Information Protocol," IETF RFC1058, 1988.
[3] J. Moy, "OSPF Version 2," IETF RFC2328, 1998.
[4] R. Callon, "Use of OSI IS-IS for Routing in TCP/IP and Dual Environments," IETF RFC1195, 1990.
[5] B. Fortz and M. Thorup, "Optimizing OSPF Weights in a Changing World," IEEE Journal on Selected Areas of Communications, vol. 20, no. 4, pp. 756-767, May 2002.
[6] E. Rosen, A. Viswanathan, and R. Callon, "Multiprotocol Label Switching Architecture," IETF RFC3031, 2001.
[7] D. Bertsekas, and R. Gallager, Data Networks, Prentice Hall, 1992.
[8] R. G. Gallager, "A Minimum Delay Routing Algorithm Using Distributed Computation," IEEE/ACM Transactions on Networking, vol. 1, pp. 130 - 141, Feb. 1977.
[9] S. Vutukury and J. J. Garicia-Luna-Aceves, "MDVA: A Distance Vector Multipath Routing Protocol," in *Proc.* IEEE INFOCOM'2001, 2001.
[10] Y. Wang, and Z. Wang, "Explicit Routing Algorithms for Internet Traffic Engineering," in *Proc.* IEEE ICCCN'99, 1999.
[11] D. Thaler and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection," IETF RFC2991, 2000.
[12] H. Saito, Y. Miyao, and M. Yoshida, "Traffic Engineering using Multiple Multipoint-to-Point LSPs," in *Proc.* IEEE INFOCOM'2000, 2000.
[13] Z. Cao, Z. Wang and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," in *Proc.* IEEE INFOCOM'2000, 2000.
[14] S. Vutukury and J. J. Garicia-Luna-Aceves, "A Traffic Engineering Approach based on Minimum-Delay Routing," in *Proc.* IEEE ICCCN'2000, 2000.
[15] Y. Lee, Y. Seok, Y. Choi, and C. Kim, "A Constrained Multipath Traffic Engineering Scheme for MPLS Networks," in *Proc.* IEEE ICC'2002, 2002.



(a) Single path, $H = 1$ (b) Three paths, $H = 1$ (c) Three paths, $H = 0$

Fig. 9 Maximum of link utilization (α) of heuristics.



(a) Five multiple paths, $H = 1$ (b) Three multiple paths, $H = 2$

Fig. 10 α under the increased hop-count constraint and multiple paths.

[16] D. Awduche, and Y. Rekhter, "Multiprotocol Lambda Switching: Combining MPLS Traffic Engineering Control with Optical Crossconnects," IEEE Communications Magazine, vol. 39, no. 3, pp. 111 - 116, March 2001.
[17] CPLEX, http://www.cplex.com
[18] R. Ahuja, T. Magnanti, and J. Orlin, Network Flows, Prentice Hall, 1993.
[19] E. L. Lawler, Combinatorial Optimization: Networks and Matroids, Holt, Rinehart and Winston, 1976.
[20] I. Cidon, R. Rom, and Y. Shavitt, "Analysis of Multi-Path Routing," IEEE/ACM Transactions on Networking, vol. 7, no. 6, pp. 885 - 896, Dec. 1999.
[21] S. Bahk, and M. Zarki, "Dynamic Multi-path Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide Area Networks," ACM Computer Communications Review, vol. 22, no. 4, pp. 53 - 64, Oct. 1992.
[22] N. S. V. Rao, and S. G. Batsell, "QoS Routing Via Multiple Paths Using Bandwidth Reservation," in *Proc.* IEEE INFOCOM'98, 1998.
[23] A. Shaikh, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," in *Proc.* ACM SIGCOMM'99, 1999.
[24] P. Bhaniramka, W. Sun, and R. Jain, "Quality of Service using Traffic Engineering over MPLS: An Analysis," in *Proc.* IEEE LCN'2000, 2000.
[25] Z. Wang, Y. Wang, and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," in *Proc.* IEEE INFOCOM'2001, 2001.
[26] M. Kodialam, and T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," in *Proc.* IEEE INFOCOM'2000, 2000.
[27] E. Dinan, D. O. Awduche, and B. Jabbari, "Analytical Framework for Dynamic Traffic Partitioning in MPLS Networks," in *Proc.* IEEE ICC'2000, 2000.
[28] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True, "Deriving Traffic Demands for Operational IP Networks: Methodology and Experience," in *Proc.* ACM SIGCOMM, August/September 2000.
[29] M. K. Girish, Z. Bei, and J. Hu, "Formulation of the Traffic Engineering Problems in MPLS based IP Networks," in *Proc.* IEEE ISCC'2000, 2000.
[30] Y. Seok, Y. Lee, Y. Choi, and C. Kim, "A Dynamic Constrained Multipath Routing for MPLS Networks," in *Proc.* IEEE ICCCN'2001, 2001.
[31] D. Eppstein, "Finding the K Shortest Paths," SIAM Journal of Computing, vol. 28, no. 2, pp.652-673, 1998.
[32] Optimized Multipath, http://www.fictitious.org/omp

Youngseok Lee Younggseok Lee received BS, MS, and Ph D degrees in 1995, 1997 and 2002, respectively, all in Computer Engineering from Seoul National University. He was a

visiting scholar to Networks lab. in University of California, Davis from Oct. 2002 to July 2003. Since July 2003, he joined the Dept. of Computer Science and Engineering, Chungnam National University. His reseaerch interests include traffic engineering in next-generation Internet and wireless LAN.

Yongho Seok Yongho Seok received BS and MS degrees in 2000 and 2002 in Computer Engineering from Handong University and in Computer Science and Engineering from Seoul National University, respectively. He is a phd student of the Dept. of Computer Science and Engineering in Seoul National University since 2002. His research interests include traffic engineering in MPLS networks, quality of service in multi-rate wireless networks, and internet traffic measurement.

Yanghee Choi Yanghee Choi received B.S. in electronics engineering from Seoul National University, M.S. in electrical engineering from Korea Advanced Institute of Science, and Doctor of Engineering in Computer Science from Ecole Nationale Superieure des Telecommunications (ENST) in Paris, in 1975, 1977 and 1984 respectively. Before joining the School of Computer Engineering, Seoul National University in 1991, he has been with Electronics and Telecommunications Research Institute(ETRI) during 1977-1991, where he served as director of Data Communication Section, and Protocol Engineering Center. He was research student at Centre National d'Etude des Telecommunications (CNET) , Issy-les-Moulineaux, during 1981-1984. He was also Visiting Scientist to IBM T. J. Watson Research Center for the year 1988-1989. He is now leading the Multimedia Communications Laboratory in Seoul National University. He is also director of Computer Network Research Center in Research Institute of Advanced Computer Technology (RIACT). He was editor-in-chief of Korea Information Science Society journals. He was chairman of the Special Interest Group on Information Networking. He has been associate dean of research affairs at Seoul National University. He is now president of Open Systems and Internet Association of Korea. His research interest lies in the field of multimedia systems and high-speed networking.