MPLS 트래픽 엔지니어링을 위한 다중경로 Constraint-based 라우팅 알고리즘

정회원 이 재 용*, 김 병 철*

Multi-path Constraint-based Routing Algorithms for MPLS Traffic Engineering

Jae-Young Lee*, Byung-Chul Kim* ***Regular Members***

요 약

이 논문에서는 MPLS를 사용한 인터넷 트래픽 엔지니어링에 사용될 수 있는 다중경로 한계조건 기반 라우팅 알고리즘을 제안한다. 기존에 사용되던 한계조건 기반 최단경로 라우팅 알고리즘은 트래픽 엔지니어링의 중요한 요소중에 하나인 대역폭 한계조건이 큰 값을 가지는 경우에는 조건을 만족하는 경로를 찾지못할 확률이 높아진다. 제안된 다중경로 알고리즘은 대역폭 한계조건을 만족하는 하나의 경로를 찾을 수 없는 경우, 대역폭 조건을 여러개의 작은 값으로 나누어 각각의 작은 대역폭 한계조건에 대해 다중경로를 찾는 알고리즘이다. 시뮬레이션을 통하여 제안한 알고리즘을 사용할 경우 주어진 네트웍 상황과 대역폭 한계 조건에서 더 높은 경로설정 성공확률을 나타내며, 네트웍 자원 이용률도 개선되는 것을 보였다.

Key Words : Traffic engineering, multi-path routing, MPLS, QoS, load balancing

ABSTRACT

This paper proposed two multi-path constraint-based routing algorithms for Internet traffic engineering using MPLS. In normal constraint-based shortest path first (CSPF) routing algorithm, there is a high probability that it cannot find the required path through networks for a large bandwidth constraint that is one of the most important constraints for traffic engineering. The proposed algorithms can divide the bandwidth constraint into two or more sub-constraints and find a constrained path for each sub-constraint, if there is no single path satisfying the whole constraint. Extensive simulations show that they enhance the success probability of path setup and the utilization of network resources.

Ⅰ. INTRODUCTION

Internet traffic engineering (TE) is a procedure of traffic mapping to network topology according to network status and traffic requirement. A major goal of Internet TE is to facilitate efficient and reliable network operations while optimizing network resource utilization and enhancing the QoS of traffic streams [1][2]. Traffic engineering has become an indispensable function in many large Internet service providers (ISP) for maximal operational efficiency of the network, because extending or upgrading networks requires high cost and much time.

---

* 충남대학교 정보통신공학과 데이터통신연구실 (jylee@datacom.cnu.ac.kr, byckim@ngn.cnu.ac.kr)

논문번호 : 030247-0609, 접수일자 : 2003년 6월 9일

※This work was supported by grant No.(R01-2003-000-10628-0) from Korea Science & Engineering Foundation.

Multi-protocol label switching (MPLS) [3] has many attractive features for Internet TE and will be widely used for TE [3]. First of all, MPLS can create easily explicit-route label switched path (ER-LSP) as needed through manual administrative action or traffic requirement by using CR-LDP [4] or RSVP-TE [5] signaling protocol. And, it can easily map traffic trunks that consist of traffic flows with similar characteristics or traffic requirements onto that ER-LSPs.

Usually, plain IGP (Internet-domain gateway protocol) routing protocols are not suitable for Internet traffic engineering. The current IGP routing protocols such as OSPF [6] and IS-IS [7] calculate only shortest-paths to destination in a distributed fashion. They do not take into account traffic requirements and network condition such as remaining bandwidths of links. Constrained Shortest Path First (CSPF) routing algorithms can calculate shortest paths that do not violate a set of constraints. For instance, when a bandwidth constraint is given for a path between source and destination, the CSPF first removes links that has remaining bandwidth less than bandwidth constraint and then calculates the shortest path. The CSPF is a promising routing algorithm for MPLS traffic engineering since it tries to find a suitable path under a given network condition and traffic requirement. There are many possible constraints CSPF can take such as bandwidth, buffer size, delay, delay jitter, loss probability and etc. In this paper, we mainly consider the bandwidth constraint.

However, for a large bandwidth constraints, there is high probability that the CSPF cannot find a bandwidth-constrained path for a given network condition. Two multi-path bandwidth-constrained routing algorithms proposed in this paper divide the bandwidth constraint into two or more sub-constraints appropriately, and find a constrained path for each sub-constraint, if there is no single path satisfying the whole bandwidth constraint. In these algorithms, we try to find minimal number of paths satisfying the constraint to restrict the computational complexity. And then, for the multiple paths found, we suggest a load balancing method between multiple paths to approximately equalize the QoS of each partitioned traffic flow. Extensive simulations show that the proposed routing algorithms enhance the success probability of path setup and network resource utilization compared to single-path constraint-based routing algorithms.

This paper is organized as follows. Section II summarizes the related works for this paper. In Section III, we present the proposed multi-path bandwidth constraint-based routing algorithms and load balancing method in detail. The performance of the proposed algorithms is shown by simulation in Section IV. We conclude this paper in Section V.

Ⅱ. RELATED WORKS

In this section, we present some research works related to the topic of this paper. Internet traffic engineering has become an essential function for Internet Service Providers (ISPs) to optimize the utilization of existing network resources and to maintain a desired QoS of user traffic with less network resources. A framework for Internet traffic engineering is presented in [1]. Awduche et. al. described the principle of traffic engineering in the Internet [2]. This document includes a set of generic recommendations and options for Internet traffic engineering. This can be used as a guide to implementors of online and offline Internet traffic engineering mechanisms. In [8], Awduche et. al. explained the attractive features of MPLS for traffic engineering and requirements for TE over MPLS. One of the most important features of MPLS for TE is its capability to set up ER-LSPs suitable for traffic or network condition.

In order to find an appropriate path that satisfies the QoS of traffic and enhances the utilization of the network, we must adopt some kind of QoS routing algorithms in networks. In [9], Chen et. al. gave an overview of the QoS routing algorithms. They classified various routing constraints and presented three routing strategies; source routing, distributed routing and hierarchical routing. They also discussed basic routing algorithms for each routing problem and compared them. There have been many QoS routing algorithms proposed for various routing constraints [10-15]. Wang and Crowcroft [10] found a bandwidth-delay-constrained path by using Dijkstra' s shortest-path algorithm. First, all links with available bandwidth less than the requirement are eliminated so that any paths in the resulting network will satisfy the bandwidth constraint. Then, the shortest path in terms of delay is found. There is a feasible path if and only if it satisfies the delay constraint.

Apostolopoulos et. al. [11] presented a QoS routing algorithm and the necessary modifications to OSPF to support this algorithm. They presented three QoS routing algorithms based on Bellman-Ford and Dijkstra' s shortest path routing algorithms. After all links that do not satisfy the bandwidth constraint are removed, the algorithms find a widest shortest path through the networks. Salama et. al. proposed a distributed heuristic algorithm for the NP-complete delay-constrained least cost routing problem called DCUR (delay-constrained unicast routing) [12]. Each node maintains two routing tables, cost vector and delay vector obtained by distance-vector protocol using link cost and delay as link metric. It selects a least-cost path as long as delay constraint is satisfied, otherwise selects a least-delay path. It also presents a method to solve routing loops.

QoS routing information such as available bandwidth and delay is propagated through networks periodically or at the time of the change of corresponding information. Thus, network nodes have somewhat inaccurate QoS routing information at some time instant. Guerin and Orda [16] studied the bandwidth-delay-constrained routing problem with inaccurate network states. Apostolopoulos et. al. [17] presented a method to improve QoS routing performance under inaccurate link state information. Mechanisms for link state update have large influence to the accuracy of information and performance of QoS routing. Ariza et. al. proposed an adaptive mechanism for link state update to stabilize the update rate independent of the traffic load and increase the QoS routing performance [18].

There have been several studies for traffic load balancing among multiple LSPs between ingress LSR and egress LSR [19-21]. MPLS-OMP (optimized multipath) [19] tries to balance the loads among multiple LSPs according to the loading for each path. The distribution of load among a set of alternate paths is determined by the amount of number space from a hash computation allocated to each path. Widjaja and Elwalid [20] proposed a load balancing protocol called "MATE" (MPLS Adaptive Traffic Engineering). The main goal of the MATE is to avoid network congestion by balancing the loads among multiple LSPs between source and destination LSRs. In the MATE, the ingress LSR transmits probe packets periodically to the egress LSR that returns them back to the ingress LSR. Based on the information in the returning probe packets, the ingress LSR can compute the LSP characteristics and distribute the load among them. Dinan et. al. proposed a stochastic framework for the traffic partitioning problem among LSRs [21]. Within this framework, multiple LSPs are modeled by parallel queueing networks and load balancing is performed among LSPs to minimize overall delay of traffic by analytical methods.

However, there are few studies for mechanisms of multiple LSP setup between ingress and egress LSR satisfying a constraint. In this paper, we propose two multi-path (LSP) bandwidth constraint-based routing algorithms and a traffic partitioning method.

Ⅲ. MULTI-PATH BANDWIDTH CONSTRAINT- BASED ROUTING ALGORITHMS

In this section, we propose two multi-path bandwidth constraint-based routing algorithms to find multiple LSPs between an ingress and egress LSR satisfying a given bandwidth constraint. First, we present general optimization criteria for this problem. Since satisfying this criteria may cause heavy computational complexity, we try to find minimal numbers of multiple paths for a given bandwidth constraint, if a single feasible path is not available.

3.1. General optimization criteria for multi-path constraint-based routing

For the same bandwidth constraint, an optimal path should have minimal end-to-end cost. Let m be the number of required paths, B_i be the bandwidth of path i, C_{ij} be the per bit cost of transmitting packets over link j of path i, and n_i be the number of links in path i. Then, we can say that an optimal multi-path should minimize the cost function C,

$$C=\sum_{i=1}^{m} B_i \sum_{j=1}^{n_i} C_{ij} \tag{1}$$

subject to the following constraints,

$$\sum_{i=1}^{m} B_i = BW_C \tag{2}$$

$$\sum_{i\in S_k} B_i \leq b_k,$$

$$S_k = \{i \mid path\ i\ includes\ link\ k\} \tag{3}$$

where b_k is the remaining (or available) bandwidth of link k, and BW_c is the bandwidth constraint.

One of the feasible solutions for the problem (1), (2) and (3) is found as follows. First, we calculate the least-cost path between the source and destination node, and allocate the path bandwidth, i,e., minimum bandwidth of all links within the path, to that path. Next, we calculate the next shortest path through the networks after removing links having no available bandwidth and allocate the path bandwidth to that path. We continue this process until we can allocate the whole bandwidth constraint to the successive shortest paths. Then, the bandwidth allocation will satisfy the optimization condition (1), under the constraints (2) and (3).

But, this algorithm may result in too many LSPs between ingress and egress LSR to satisfy the whole constraints. This increases the computation and signaling time. It is also a very difficult task to partition and assign the traffic among a large number of parallel LSPs. Thus, it is one of the important objectives to maintain the number of LSPs satisfying the whole constraint as small as possible.

Next, we introduce two multi-path constraint-based routing algorithms: equal-bandwidth (EB) multi-path routing and maximum path bandwidth first (MPBF) multi-path routing.

3.2. Equal bandwidth (EB) multi-path constraint-based routing algorithm

This algorithm divides the bandwidth requirement into multiple sub-constraints with equal bandwidth, if there is no single path between source and destination. For each sub-constraint, the algorithm tries to find a constrained path by using the CSPF algorithm. For a whole bandwidth constraint BW_c, it first tries to find a single path whose path bandwidth is greater than or equal to BW_c. We can accomplish this by using a normal shortest path first (SPF) routing algorithm such as Dijkstra's [21] or Bellman-Ford algorithm [22], after removing all links having the available bandwidth less than BW_c. If there is no such a single path, it will try to calculate two paths satisfying the sub-constraint $BW_c/2$. If it fails, it will compute three paths satisfying BW_c /3, and so on. We continue this process until we find multiple paths whose total path bandwidth is equal to or greater than the bandwidth requirement BW_c. In section IV, we show by simulation that three or four partitions of the constraint saturate the success probability of constraint-path setup. Fig. 1 shows the flow chart of the EB multi-path constraint-based routing algorithm.

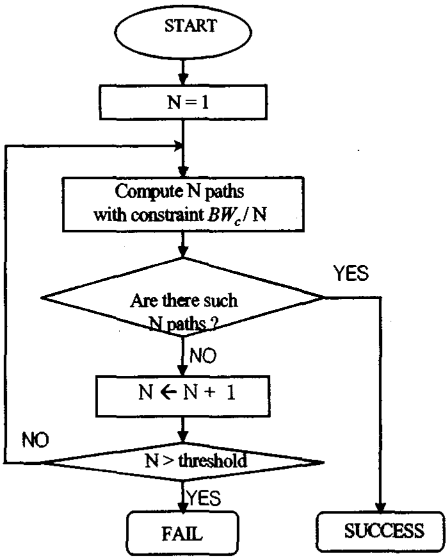


Fig. 1 Equal bandwidth (EB) multi-path constraint-based routing algorithm

This algorithm increases the success probability to find paths through networks for the same bandwidth constraint compared to a single path constraint-based routing as shown in section IV. But, there is still room to reduce the number of paths to satisfy a bandwidth requirement. Let us look at the example situation in Fig. 2. There are 4 paths between source and destination. Path bandwidths of them are 5 Mbps, 3 Mbps, 2 Mbps and 2 Mbps, respectively. If we want to find paths with bandwidth constraint 7 Mbps by using the EB multi-path routing, we have to compute 4 paths with sub-constraint 7/4 Mbps. However, if we can divide the constraint unequally, for example, 4Mbps and 3 Mbps, only the first two paths can satisfy the bandwidth requirement 7 Mbps.

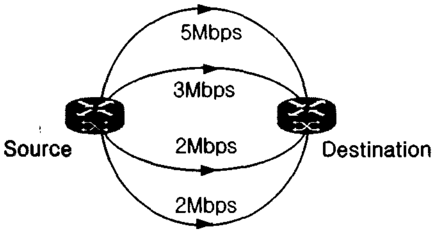


Fig. 2 Example network for a multi-path finding problem

3.3. Maximum Path Bandwidth First (MPBF) multi-path routing algorithm

In order to satisfy the bandwidth requirement, this algorithm tries to find minimum number of multiple paths. First, the algorithm tries to compute a single path for the constraint. If there is no such path, it computes the maximum bandwidth path between source and destination. Then, it allocates its path bandwidth Δ_1 (minimum of available bandwidths of links included along the path) to that path. After that, the algorithm computes another constrained path using the remaining bandwidth constraint (BW_c -Δ_1).

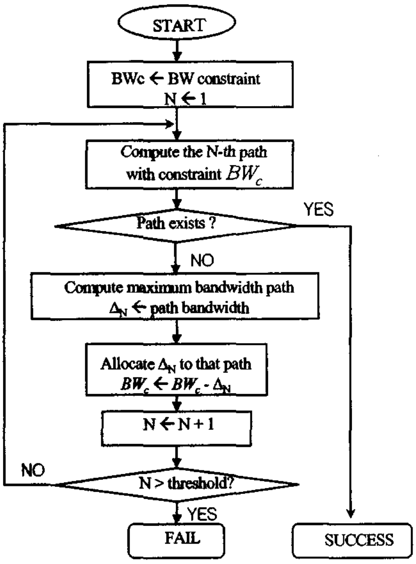


Fig. 3 Maximum Path bandwidth First (MPBF) multi-path constraint-based routing algorithm

If it is found, this completes the algorithm. If it is not, the algorithm calculates another maximum bandwidth path and allocates its path bandwidth Δ_2 to that path. The algorithm continues this process until it allocates all of the bandwidth requirement BW_c to the paths found during algorithm computation. In section IV, simulation shows that the algorithm requires less number of paths to allocate the bandwidth constraint BW_c compared to the EB multi-path routing. The flow chart of the MPBF multi-path routing algorithm is shown in Fig. 3.

There can be several algorithms to calculate a maximum bandwidth path between source and destination used in the MPBF routing algorithm. In this paper, we present a widest path first (WPF) routing algorithm by modifying Dijkstra' s algorithm. Originally, Dijkstra' s algorithm [21] solves the single-source shortest-paths problems on a weighted directed graph $G=(V,E)$, where V is a node set and E is a link set. Here, the WPF algorithm modifies Dijkstra' s algorithm to find a single-source maximum bandwidth path problem. Let $b(u,v)$ be the available (or remaining) bandwidth for the link $(u,v) \in E$. $\delta(s,v)$ denotes the final maximum bandwidth from node s to node v after completion of the algorithm, and let MB[v] be the variable denoting the current maximum bandwidth of node v from the source s during algorithm calculation. This algorithm maintains a set S of nodes whose final maximum bandwidth from the source s has already been determined. That is, for all nodes $v \in S$, we have MB[v]= $\delta(s,v)$. The algorithm repeats the selection of the node $u \in (V-S)$ which have the maximum bandwidth from source, inserts u into S, and updates the maximum bandwidth MB[v] of node v adjacient to node u. It also maintains a priority queue Q that contains all the nodes in $V-S$, ordered by their MB[v] values. Fig. 4 represents a pseudocode for the WPF Dijkstra' s algorithm.

Lines 1 to 4 in the pseudocode initialize MB [v] = 0 for all $v \in V$ except the source node s. $\pi[v]$ denotes the predecessor node of v in the maximum bandwidth tree from the source s. For the first time, line 8 selects the source node s because MB[s] $=\infty$. When Q becomes empty (ø), the algorithm completes the calculation of maximum bandwidth tree from source s to the other nodes in the network. Maximum bandwidth of node v is stored in MB[v], and we can trace the maximum bandwidth tree using $\pi[v]$.

The MPBF multi-path routing algorithm does not satisfy the optimization criteria (1), (2) and (3). However, it does find the minimum number of multi-path for any bandwidth constraint, which is one of the most important objectives in a multi-path constraint routing.

```
WPF_Dijkstra($G, b, s$)

1.  for each node $v \in V[G]$
2.    do $MB[v] \leftarrow 0$
3.       $\pi[v] \leftarrow$ Null
4.  $MB[S] \leftarrow \infty$
5.  $S \leftarrow \phi$
6.  $Q \leftarrow V[G]$
7.  while  $Q \neq \phi$
8.    do  $u \leftarrow$ Extract node with max.$MB[v]$, $v \in Q$
9.        $S \leftarrow S \cup \{u\}$
10.       for each node $v \in Adj[u]$
11.         if  $MB[v] < \min(MB[u], b(u,v))$
12.           then  $MB[v] \leftarrow \min(MB(u), b(u,v))$
13.                 $\pi[v] \leftarrow u$
```

Fig. 4 Pseudocode for widest shortest path first (WPF) Dijkstra' s algorithm

3.4. Load balancing between multiple paths

If a single path complies with a whole bandwidth constraint, there is no need to balance the traffic load. However, if we find multiple paths by using the above algorithms, we should partition the traffic load optimally for mapping to multiple paths.

Usually, we can determine the level of congestion of a link i by link utilization ρ_i defined by

$$\rho_i = \frac{\overline{x_i}}{C_i} = \frac{C_i - b_i}{C_i} \quad (4)$$

where C_i is the link capacity of link i and $\overline{x_i}$ is the current bandwidth usage of link i. When ρ_i approaches 1, the link becomes congested and packet loss and delay increase drastically. So, the basic idea of load balancing between multiple paths found by the above algorithms is to maintain the remaining path bandwidths of multiple paths to be equal each other after allocation of traffic load to each path.

Let N be the total number of paths found, Δ_i be the path bandwidth of the i-th path p_i, and a_i be the traffic load allocated to the i-th path. If we assume all the link capacities are equal, the above mechanism of load balancing can be expressed as follows.

$$\Delta_i = \min_{k \in p_i} \{b_k\} \geqq a_i \quad (5)$$

$$\Delta_1 - a_1 = \Delta_2 - a_2 = \ldots = \Delta_N - a_N \quad (6)$$

$$\sum_{i=1}^{N} a_i = BW_C \quad (7)$$

We can solve a_i (i = 1,$\cdots$,N) because there are N equations in (6) and (7). For example, if N = 2, equations (5), (6) and (7) become

$$\Delta_1 \geqq a_1 \text{ , } \Delta_2 \geqq a_2 \quad (8)$$

$$\Delta_1 - a_1 = \Delta_2 - a_2 \quad (9)$$

$$a_1 + a_2 = BW_C \quad (10)$$

,which produce solutions

$$a_1 = \frac{1}{2}(\Delta_1 - \Delta_2 + BW_c) \quad (11)$$

$$a_2 = \frac{1}{2}(\Delta_2 - \Delta_1 + BW_c) \quad (12)$$

From the fact that we found two paths in this example, we can say

$$0 < \Delta_1 < BW_c,\ 0 < \Delta_2 < BW_c \quad (13)$$

If Δ_1 or Δ_2 is greater than or equal to BW_c, there must be a single path satisfying constraint BW_c. Thus, a_1 and a_2 in equations (11) and (12) have positive values.

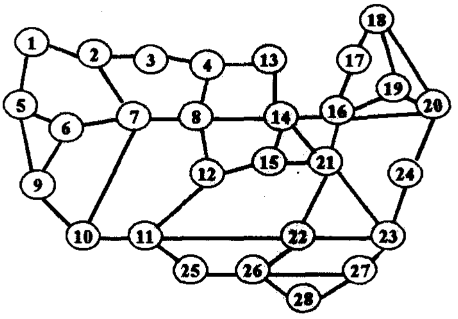
In real network situation, we can restrict the maximum of link utilization $\rho_{\max}$ less than 1 in order to provide a reasonable quality of service to user traffic. After we adjust the available path bandwidth Δ_i so that the link utilization of each link in path p_i does not exceed $\rho_{\max}$ after allocation of bandwidth Δ_i, we can apply the equations (5), (6) and (7) to calculate the bandwidth allocation a_i.

Ⅳ. SIMULATION

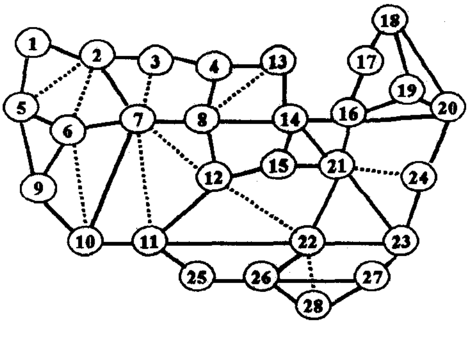
In this section, we present and compare the routing performance of the EB and MPBF multi-path routing algorithms by simulation. For our simulation, we select the network topology as shown in Fig. 5(a), which is the long haul network of U.S.A consisting of 28 nodes and 45 links. As needed, we add extra links to the network to increase the connectivity as shown in Fig. 5(b) and (c).

Fig. 6 shows the success probabilities of the EB and MPBF multi-path routings according to the bandwidth constraint. We assume that the capacity of links is equal to 30 Mbps. The available (remaining) bandwidths of links are generated randomly from 1 Mbps to 30 Mbps.

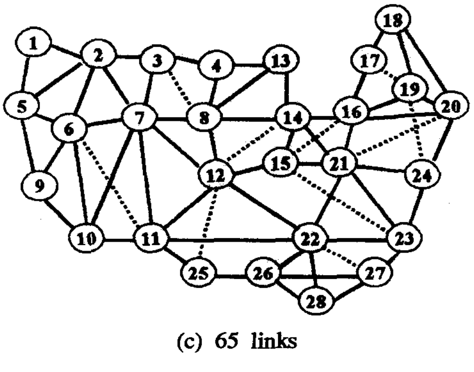
The results are obtained by repeated simulation for each path finding. We choose node 6 as a source and node 27 as a destination. The figure shows that the success probability of path setup decreases as the bandwidth constraints increase, as we expect. Multi-path routing algorithms have larger success probability than the single path routing. As the number of paths increases, the success probability also increases. The success probability of the MPBF multipath routing is much higher than that of the EB multi-path routing. In other word, the MPBF routing can satisfy the bandwidth constraint with smaller number of paths between source and destination than the EB routing.



(a) 45 links



(b) 55 links



(c) 65 links

Fig. 5 Network topology for simulation of multi-path routing algorithms.

Fig. 7 represents the success probability of path setup versus number of multiple paths. As the number of paths increases, so does the success probability. But, the rate of increase of the success probability is decreased as the number of multipath. It is almost saturated when the number of paths is 4. In Fig. 8, we show the success probability to set up the bandwidth constrained paths as the number of links in the network increase. We add appropriate links to the network for the simulation. The number of links taken is 45, 55 and 65. As the number of links increase, the success probability increases accordingly, since the possibility of path finding is enhanced when the connectivity of the network becomes abundant.

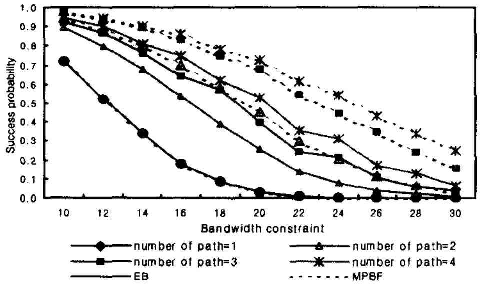


Fig. 6 Success probability of multi-path setup versus bandwidth constraints of the EB and MPBF routings for 45 links, source node 6, and destination node 27.

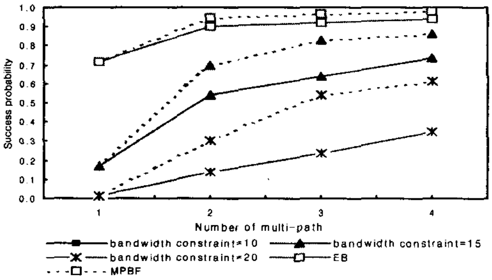


Fig. 7 Success probability of multi-path setup versus number of multi-path of EB and MPBF routings for 45 links, source node 6, and destination node 27 and for various bandwidth constraints.

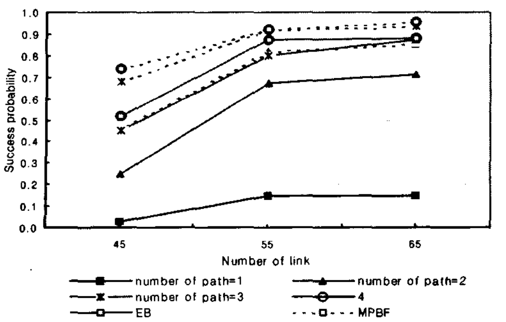
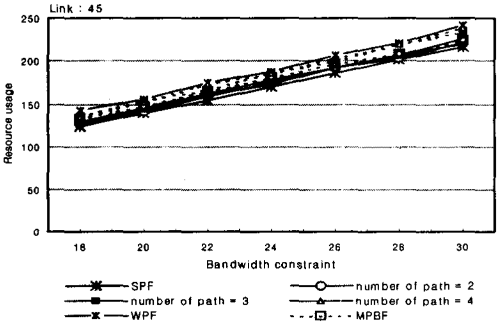


Fig. 8 Success probability of multi-path setup versus number of links of EB and MPBF routing for source node 6, destination node 27 and bandwidth constraint 20 Mbps

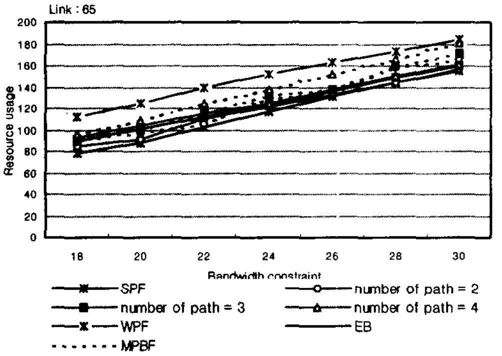
One of the most important performance parameters in the QoS routing algorithm is the usage of network resource such as bandwidth. Fig. 9 shows the simulation results of the bandwidth usage for each routing algorithm. We define the bandwidth usage U as,

$$U = \sum_{all\ p_i} \Delta_i \cdot h_i \qquad (14)$$

where Δ_i is the path bandwidth of path p_i and h_i is the number of hops for path p_i between source and destination. Fig. 9 shows the value of U for the widest path first (WPF) routing, the shortest path first (SPF) routing, the EB multi-path routing and the MPBF multi-path routing. For the EB and MPBF routings, we find the value of U for the number of paths up to 4 paths. As we expect, the WPF routing has the largest bandwidth usage for the same bandwidth requirement, since it tries to find the widest bandwidth path regardless of hop counts. The SPF routing has the least bandwidth usage. As the number of links increase, the bandwidth usage decreases because it is more likely that the algorithms can find shorter path between source and destination as shown in Fig. 9 (a) and (b). The values of U for the EB and the MPBF lie between those of WPF and SPF algorithms. The bandwidth usage of the MPBF is a little larger than that of the EB routing. From this result, we can say that the MPBF routing is a hybrid algorithm of the SPF and the WPF. Its performance is much closer to that of the SPF. In Fig. 10, we show the bandwidth usage for the EB and the MPBF multi-path routings regardless of the number of paths found to satisfy the bandwidth constraints for the number of links 45 and 65. The available bandwidth of each link is generated randomly from 1 Mbps to 45 Mbps. We take node 6 as a source and node 27 as a destination. We only include the case where multi-path finding is successful for both EB and MPBF with the maximum numbers of paths 4. The figures show that the MPBF uses more bandwidth than the EB routing. The difference of U between them is about 4% for 45 links and 10% for 65 links.



(a) 45 links



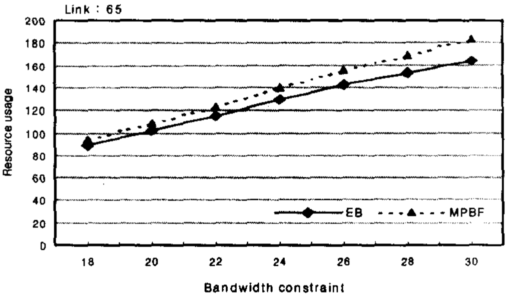
(b) 65 links

Fig. 9 Bandwidth usage of multipath routing algorithm according to the number of paths for source node 6 and destination node 27

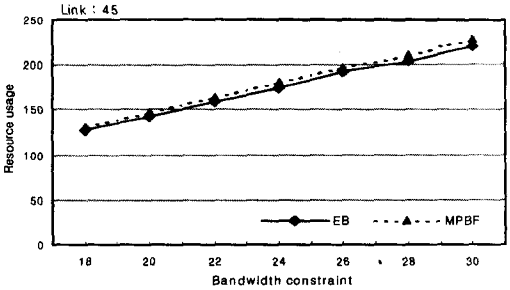
In Fig. 11, we find the bandwidth rejection ratio R defined by

$$R=\frac{Total\ bandwidth\ rejected}{Total\ bandwidth\ requested} \quad (15)$$

for the proposed algorithms. For the network topology shown in Fig. 5, the capacity of links is assumed to be 10 Mbps. Bandwidth requests are arrived according to a Poisson process with the arrival rate λ. Requested bandwidth of each arrival is generated by using uniform distribution from 1 Mbps to 5 Mbps. The holding time of each bandwidth request is assumed to be exponentially distributed with mean $1/\mu$ =1sec. The total offered load to the network, Λ, becomes $\lambda \cdot \overline{B}/\mu$, where $\overline{B}$ is the mean requested bandwidth.



(a) 45 links



(b) 65 links

Fig. 10 Bandwidth usage of the EB and the MPBF multi-path routing algorithms regardless of the number of multipath for source node 6 and destination node 27

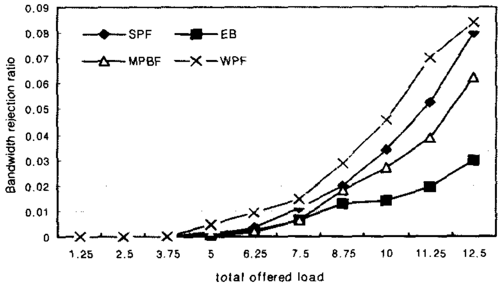


Fig. 11 Bandwidth rejection ratio of multi-path routing algorithms for link capacity of 10 Mbps, mean bandwidth request of 3 Mbps, Poisson arrival and exponentially distributed session holding time of mean 1 sec

The widest path first (WPF) routing with a single path represents the largest bandwidth rejection ratio among four algorithms considered. This is because the WPF routing finds a path that has the largest path bandwidth and does not find the shortest one. The single path CSPF routing has smaller value of R than that of the WPF. Two multi-path routing EB and MPBF have superior performance to the two single path routing algorithms, the WPF and the single path CSPF. Between the two algorithms, the EB multipath routing has the smaller bandwidth rejection ratio than that of the MPBF routing. This means that the bandwidth utilization of the EB routing is more efficient than the MPBF.

V. CONCLUSION

In this paper, we proposed two multi-path bandwidth constraint-based routing algorithms. When there is no single path through the network satisfying a whole bandwidth constraint, suggested algorithms divide the bandwidth constraint into multiple sub-constraints and find a constrained path for each sub-constraint. In the equal-bandwidth (EB) multi-path routing, the total bandwidth constraint is divided equally. In the maximum path bandwidth first (MPBF) multi-path routing, we apply the CSPF routing and the WPF routing alternately and recursively, and allocate appropriate bandwidth to each path found. We also propose a simple load balancing method among multiple paths.

Simulation results show that these algorithms improve the success probability of constrained-based path setup routing. The MPBF routing needs less number of paths for the same bandwidth constraint than that of the EB routing. But, the MPBF routing utilizes a bit more bandwidth resource than the EB routing does, for the same condition. The two multi-path routing algorithms have better performance for the more complex networks. One can select one of the two algorithms according to network condition and management policy. Although the proposed algorithms is somewhat more complex to implement than single-path routing algorithms, they are still a kind of polynomial order algorithms with complexity $O(E\log V)$ (where E is number of links and V is the number of nodes in the network), as for the traditional shortest-path routing algorithms. Since the proposed multi-path routing algorithms increase the probability of successful path finding for a given bandwidth requirement and enhance the utilization of network resource, they are promising routing algorithms for MPLS traffic engineering.

참고문헌

[1] D. Awduche, A. Chiu, A. Elwalid and X. Xiao, "A famework for internet traffic engineering", *Internet Draft* <draft-ietf-tewg-framework-02.txt>, July 2000.

[2] D. O. Awduche, A. Chiu, A. Elwalid, I. Widjaja and X. Xiao, "Overview and principles of internet traffic engineering" *Internet Draft* <draft-ietf-tewg-principles-01.txt>, January 2001.

[3] E. Rosen, A. Viswanathan and R. Callon,"Multi-protocol label switching architecture", IETF RFC 3031, January 2001.

[4] L. Andersson, R. Callon, P. Doolan and P. Vaananen, "Constraint-based LSP setup using LDP", *Internet Draft* <draft-ietf-mpls-cr-ldp-05.txt>, July 2000.

[5] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan and G. Swallow, "RSVP-TE: Extension to RSVP for LSP tunnels", *Internet Draft* < draft-ietf-mpls- rsvp-lsp- tunnel-08.txt>, August 2000.

[6] J. Moy, " OSPF version 2", *IETF RFC 1247*, July 1991.

[7] D. Oran, "OSI IS-IS intra-domain routing protocol", *IETF RFC 1142*, February 1990.

[8] D. Awduche, J. Malcolm, J. Agogbua and J. McManus, "Requirements for traffic engineering over MPLS", *IETF RFC 2702*, September 1999.

[9] S. Chen and K. Nahrstedt, "An overview of quality of service routing for next-generation high-speed networks : problems and solutions", *IEEE Network Mag.* pp. 64-79, November/December 1998.

[10] Z. Wang and J. Crowcroft, "QoS routing for supporting resource reservation", *IEEE JSAC*, Vol. 14, No. 7, pp. 1228-1234, September. 1996.

[11] G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin, A. Orda and T. Przygienda, "QoS routing mechanisms and OSPF extensions", *IETF RFC 2676*, August 1999.

[12] H. F. Salama, D. S. Reeves, and Y. Viniotis, "A distributed algorithm for delay-constrained unicast routing," *IEEE/ACM Transactions on Networking*, Vol. 8, No. 2, pp. 239-250, June. 2000.

[13] Q. Ma and P. Steenkiste, "Quality-of-service routing with performance guarantees", *Proc. 4th Int'l. IFIP Wksp.* QoS, May 1997.

[14] S. Chen and K. Nahrstedt, "On finding multi-constrained paths", *IEEE ICC '98*, vol. 8, No. 3, pp. 305-320, June 1998.

[15] K. G. Shin and C. C. Chou, "A distributed route-selection scheme for establishing real-Time channel", *6th IFIP Int'l. Conf. High Perf. Networking*, pp. 319-329, September 1995.

[16] R. Guerin and Ariel Orda, "QoS-based routing in networks with inaccurate information: theory and algorithms", *IEEE/ACM Transactions on Networking*, Vol. 7, No. 3, pp. 350-364, June 1999.

[17] G. Apostolopoulos and D. Williams, "Improving QoS routing performance under inaccurate link state information", *Proc. of the 16th internationsl Teletraffic Congress (ITC-16)*, Endinburgh, UK, June 1999.

[18] A. Ariza, E. Casilari and F. Sandoval, "QoS routing with adaptive updating of link states", *Electronics Letters* 26th, Vol. 37, No. 9, pp. 604-606, April 2001.

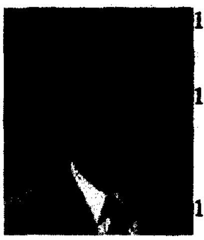
[19] V. Curtis, "MPLS optimized multipath (MPLS-OMP)", *Internet Draft* <draft-ietf-mpls-omp-00>, August 1999.

[20] I. Widjaja and A. Elwalid, "MATE : MPLS adaptive traffic engineering", *Internet Draft* <draft-widjaja-mpls-mate-01.txt>, October 1999.

[21] E. W. Dijkstra, "A note on two problems in connection with graphs", *Number. Math.*, Vol. 1, pp. 269-271, October. 1959.

[22] R. E. Bellman, 'Dynamic programming', Princeton, NJ: Princeton University Press, 1957.

이 재 용(Jae-Yong Lee) 정회원



1988년 : 서울대학교 전자공학과 (공학사)

1990년 : 한국과학기술원 전기 및 전자공학과 (공학석사)

1995년 : 한국과학기술원 전기 및 전자공학과 (공학박사)

1990년~1995년 : 디지콤 정보통신연구소 선임연구원

1995년~현재 : 충남대학교 정보통신공학부 부교수

<관심분야> 초고속통신, 인터넷, 네트워크 성능분석

김 병 철(Byung-Chul Kim) 정회원

1988년 : 서울대학교 전자공학과 (공학사)

1990년 : 한국과학기술원 전기 및 전자공학과 (공학석사)

1996년 : 한국과학기술원 전기 및 전자공학과 (공학박사)

1993년~1999년 : 삼성전자 CDMA 개발팀

1999년~현재 : 충남대학교 정보통신공학부 조교수

<관심 분야> 이동인터넷, 이동통신 네트워크, 데이터통신