Explicit Multicast Routing Algorithms for Constrained Traffic Engineering

Yongho Seok, Youngseok Lee, Yanghee Choi
Seoul National University
Seoul, Korea
{yhseok, yslee, yhchoi}@mmlab.snu.ac.kr

Changhoon Kim
Electronic Telecommunication Research Institute
Taejon, Korea
kimch@etri.re.kr

Abstract— **This paper presents a new traffic engineering technique for dynamic constrained multicast routing, where the routing request of traffic arrives one-by-one. The objective we adopted in this paper is to minimize the maximum of link utilization. Although this traffic engineering is useful to relax the most heavily congested link in Internet backbone, the total network resources, i.e. sum of link bandwidth consumed, could be wasted when the acquired path is larger(in terms of number of hops) than the conventional shortest path. Accordingly we find a multicast tree for routing request that satisfies the hop-count constraint. We formulate this problem as mixed-integer programming problem and propose a new heuristic algorithm to find a multicast tree for multicast routing request. The presented heuristic algorithm uses the link-state information, i.e. link utilization, for multicast tree selection and is amenable to distributed implementation. The extensive simulation results show that the proposed traffic engineering technique and heurisitc algorithm efficiently minimize the maximum of link utilization better than the shortest path.**

I. Introduction

The dynamic traffic engineering problem in Internet is how to set up paths between edge routers in a network to meet the traffic demand of a request while achieving low congestion and optimizing the utilization of network resources. In practice, the key objective of traffic engineering is usually to minimize the utilization of the most heavily used link in the network, or the maximum of link utilization. Since the 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 link utilization could be solved by the multi-commodity network flow formulation[11].

However, the present traffic engineering technique assumes that the traffic routing is done in unicast. We extend the scope of traffic engineering to multicast environment. Therefore, the problem is to set up bandwidth guaranteed multicast tree in a network for minimizing the maximum link utilization. Although this traffic engineering scheme is useful to minimize the maximum link utilization, it may require more total network bandwidth resources, i.e. sum of assigned bandwidth at each link of the paths, than the single shortest path. Therefore, the maximum hop-count constraint should be incorporated into multicast routing scheme in order not to waste bandwidth. We formulate this problem to mixed-integer programming(MIP) problem but it is $NP-hard$ problem. So, this paper proposes a practical heuristic algorithm of polynomial running time that finds hop-count constrained multicast tree to minimize the maximum of link utilization for each traffic request, while satisfying the requested traffic demand.

We assume that the traffic request is composed of the source, the destinations set and the traffic demand of multicast session. A traffic demand represents the average traffic volume between edge routers, in bps. For Virtual Private Network (VPN) application, the traffic demand may be the requested amount of bandwidth reservation. Even though the traffic demand varies largely at nodes near end users, it becomes quite stable for the backbone network with aggregated traffic.

This problem is motivated by the need of service providers to quickly setup constrained paths for multicast routing in their networks. An important context in which these problems arise is that of dynamic label switched path(LSP) setup in Multi-Protocol Label Switched(MPLS) networks. In Multi-Protocol Label Switching (MPLS) networks[2] where IP packets are switched through the pre-established Label Switched Path (LSP) by signaling protocols, a multicat tree can be used to forward packets belonging to the same "forwarding equivalent class (FEC)" by explicit routing. We assume that quasi-static information such as priorly known network topology. The only dynamic information available to the routing algorithm is the link utilization which is provided by extension of several routing protocol such as Open Shortest Path First(OSPF)[3] and Intermediate System to Intermediate System(IS-IS)[4].

There are some requirements that a multicast tree setup technique must satisfy in order to be useful in practice. Since the possibility of having new routing request in the future cannot be excluded, the routing algorithm must be an on-line algorithm capable of handling requests in an "optimal" manner when the requests are not all presented at once. Since all traffic requests are not known in advance, the current maximum link utilization is not optimal. To find the optimal maximum link utilization value, it is necessary that all the established paths for the previous requests need to be re-optimized whenever a new request arrives or the traffic characteristics change. However re-

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 Electronic Telecommunication Research Institute, 2001, Korea.

routing of existing requests causes a lot of path disruptions and this should not be allowed.

The remainder of this paper is organized as follows. The related works are introduced in section II. The proposed algorithm is explained in section III. The results of the performance evaluation by simulation are discussed in section IV, and section V concludes this paper.

II. Related Work

The problem of computing the minimum cost tree for a given multicast group is known as a Steiner tree problem. This Steineer tree problem is $NP-complete$. Some recently approximation algorithm of this routing problem in directed networks is proposed in [5].

A minimal hop like algorithm for routing unicast flows which attempts to balance the load of network traffic is proposed in [6]. This widest-shortest algorithm finds a feasible minimal hop path between two node such that the chosen path maximize the residual capacity of the bottleneck link along the path. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed in [8], and it is shown that congestion can be greatly reduced. In [9], it is shown that the quality of services can be enhanced by dividing the transport-level flows into UDP and TCP flows. However these works did not consider path calculation problem.

For the MPLS network, a traffic engineering method using multiple multipoint-to-point LSPs is proposed in [10], where backup routes are used against failures. Hence, the alternate paths are used only when primary routes do not work. In [11], the traffic bifurcation linear programming (LP) problem is formulated and heuristics for the non-bifurcating problem are proposed. Although [11] minimizes the maximum of link utilization, it does not consider the total network resources and constraints. The authors further showed that the traffic bifurcation LP problem can be transformed to the shortest path problem by adjusting link weights in [16].

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 a unicast routing and does not include the constraint such as hop-count. [12] proposes a constrained multipath traffic engineering for MPLS networks. Although this assumes unicast and multipath, the traffic engineering with the constraint such as maximum hop count and path count is formulated in mixed-integer programming problem and a heuristic algorithm is proposed. In [15], a minimal interference based algorithm is presented for dynamic routing of multicast request.

[17] proposes an adaptive traffic assignment method to multiple paths with measurement information for load balancing, though this work uses multipath. For differentiated services, finding the traffic split ratios to minimize the end-to-end delay and loss rates is proposed in [13]. However, how to find the appropriate multiple paths is not covered.

III. Hop-count Constrained Multicast Traffic Engineering

A. Problem Definition

In this section, we define the hop-count constrained multicast traffic engineering problem in mixed integer programming formulation. For conciseness and ease of terminology, we focus on finding label switched paths(LSPs) for multicast in MPLS networks in the rest of this paper. 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 $(k \in K)$ is given for a node pair between an ingress router (s_k) and an egress router set(T_k) consisting of multicast group. Ingress router (s_k) is the source of the multicast connection, and egress routers $(t_k \in T_k)$ are destinations. For each traffic demand, there is a maximum number of hop count constraint, H_k.[1]

We classify the hop-count constrained multicast traffic engineering problem into generally two categories. One is the traffic bifurcation case and another is the traffic non-bifurcation case. In this traffic bifurcation case, the demand of a multicast routing request is routed by using multiple multicast trees. But in the traffic non-bifurcation case, each traffic request is routed through the only single multicast tree.

First, we formulate the hop-count constrained multicast traffic engineering problem of the traffic non-bifurcation case. The variable $X_{ij}^k(h)$ represents the traffic demand k that flows through link (i, j), where j is h hops far from s_k. The integer variable Y_{ij}^k tells whether link (i, j) is used or not for the multicast tree rooted at the ingress router s_k and reaching all egress routers t_k. Let d_k be a scaling factor to normalize the total traffic demand from the source to 1. The mixed integer programming problem is formulated as follows.

$$Minimize\ \alpha + c \cdot \sum_{(j,i)\in E} \sum_{k\in K} d_k Y_{ij}^k$$

Subject to

$$\sum_{j:(i,j)\in E} X_{ij}^k(h) = \begin{cases} 1, k \in K, i \in s_k, h = 1 \\ 0, k \in K, i \notin s_k, h = 1 \end{cases} \quad (1)$$

$$\sum_{j:(i,j)\in E} X_{ij}^k(h+1) - \sum_{j:(j,i)\in E} X_{ji}^k(h) = 0 \quad (2)$$

$$, k \in K, i \notin s_k, T_k, 1 \le h < H_k$$

$$\sum_{j:(j,i)\in E} Y_{ji}^k = 1, k \in K, i \in T_k \quad (3)$$

[1] $H_k = H + H_{MH_k}$, H_{MH_k} is the minimum number of hop counts from s_k to T_k for traffic demand k. All destinations $t_k \in T_k$ are reachable from s_k within H_{MH_k} hop. H is additional hop-count that is added to H_{MH_k}.

$$0 \leq Y_{ij}^k - \sum_{h=1}^{H_k} X_{ij}^k(h) < 1 \quad (4)$$

$$\sum_{k \in K} d_k Y_{ij}^k \leq c_{ij}\alpha, \forall (i,j) \in E \quad (5)$$

$$where, 0 \leq X_{ij}^k(h) \leq 1, Y_{ij}^k \in \{0,1\}, 0 \leq \alpha, h \in Z$$

The objective is to minimize the maximum of link utilization, α. If there are solutions with same maximum of link utilization, the optimal is to find one with minimum resource utilization among them. Constraint (1) says that the sum of total outgoing traffic over the first hop from the source is 1, and all nodes over the first hop from the source never receive the traffic except the source node. Constraint (2) is the hop-level flow constraint which means 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. Constraint (3) means that all destinations must be connected from the source by using multicast tree. Constraint (4) means that only the link (i,j) being used by multicast tree is computed for the maximum link utilization. Constraint (5) means that the maximum link utilization for traffic demand k is α. This problem is $NP-hard$ because it includes the constrained integer variables.

Next, we formulate the hop-count constrained multicast traffic engineering problem of the traffic bifurcation case. For this purpose, we define the additional constraint(P_k) on the number of the multiple multicast trees to be used the multicast routing. The variable $X_{ij}^k(h,p)$ represents the tree p, $1 \leq p \leq P_k$, of the traffic demand k that flows through link (i,j), where j is h hops far from s_k. The variable $L_{ij}^k(p)$ represents the fraction of the traffic demand k assigned to link (i,j) and the tree p. The integer variable $Y_{ij}^k(p)$ tells whether link (i,j) is used or not for the multicast tree p rooted at the ingress router s_k and reaching all egress routers t_k. Let d_k be a scaling factor to normalize the total traffic demand from the source to become 1. The mixed integer programming problem is formulated as follows.

$$Minimize\ \alpha + c \cdot \sum_{(j,i) \in E} \sum_{k \in K} \sum_{p=1}^{P_k} d_k L_{ij}^k(p)$$

$$Subject\ to$$

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

$$\sum_{j:(i,j) \in E} X_{ij}^k(h+1,p) - \sum_{j:(j,i) \in E} X_{ji}^k(h,p) = 0 \quad (7)$$

$$, k \in K, i \neq s_k, t_k, 1 \leq p \leq P_k, 1 \leq h < H_k$$

$$\sum_{j:(j,i) \in E} Y_{ji}^k(p) = 1, k \in K, i = T_k, 1 \leq p \leq P_k \quad (8)$$

$$0 \leq Y_{ij}^k(p) - \sum_{h=1}^{H_k} X_{ij}^k(h,p) < 1 \quad (9)$$

$$\sum_{k \in K} \sum_{p=1}^{P_k} d_k L_{ij}^k(p) \leq c_{ij}\alpha, \forall (i,j) \in E \quad (10)$$

$$L_{ij}^k(p) \leq Y_{ij}^k(p) \quad (11)$$

$$L_{ij}^k(p) \leq \sum_{j:(j,i) \in E} L_{ji}^k(p), i \neq s_k \quad (12)$$

$$\sum_{p=1}^{P_k} \sum_{j:(j,i) \in E} L_{ji}^k(p) = 1.0, i = t_k \quad (13)$$

$$where, 0 \leq X_{ij}^k(h,p) \leq 1, 0 \leq L_{ij}^k(p) \leq 1.0$$

$$, Y_{ij}^k(p) \in \{0,1\}, 0 \leq \alpha, h \in Z, p \in Z$$

The objective also is to minimize the maximum of link utilization, α. Constraint (6)$-$(10) mean the same as the traffic non-bifurcation case. In constraint (10), there is only a minor change that the variable $L_{ij}^k(p)$ is used to present the traffic split ratio. Constraint (11) means that the load of link (i,j) excluded by multicast tree p is zero. Constraint (12) means that for all nodes except source, the amount of total incoming traffic to a node can not be less than that of outgoing traffic from the node to the other node. Constraint (13) means that the incoming traffic of the destination router through the multiple routes must be the amount of the traffic demand. This problem also is $NP-hard$ because it includes the constrained integer variables.

Since the traffic bifurcation case is allowable to split traffic in an arbitrary ratio, it can be used to obtain the optimal solution of the our multicast routing problem. But it is not always permissible for the routing algorithm to split traffic in arbitrary manner since the traffic being routed may be inherently unsplittable. In the rest of this paper, we only concentrate on the traffic non-bifurcation case.

B. Proposed Heuristic

We propose a heuristic algorithm to find hop-count constrained multicast tree for each traffic demand request between an ingress router and multiple egress routers. The proposed algorithm consists of two parts: 1) modifying the original graph to the hop-count constrained one [12], 2) finding a multicast tree to minimize the maximum link utilization.

Step 1 : **Conversion to hop-count constrained graph** The given network, $G = (N,E)$, is converted to H_k hop-count constrained graph, $G' = (N',E')$, where N' and E' are transformed as follows,

$$N' = \cup_{0 \leq m \leq H_k}\ N'_m,$$

$$N'_0 = \{s_k\},$$

$$N'_m = \{j_m | (i,j) \in A, i_{m-1} \in N'_{m-1}\}.$$

$$E' = \cup_{1 \leq m \leq H_k}\ E'_m,$$

$$E'_1 = \{(s_k, i) | (s_k, i) \in E\},$$

$$E'_m = \{(i_m, j_m) | i_m \in N'_{m-1}, j_m \in N'_m, (i, j) \in E\}.$$

An example of graph conversion is given in Figure 1. Figure 1 (a) represents the original network topology. When a traffic demand request from node 1 to node 4 which requires bandwidth of 3 Mbps with the hop-count constraint of one additional hop and the path-count constraint of two arrives, the graph in Figure 1 (b) is derived after adding redundant nodes and links. It is easily seen that any path traversed from node 1 to node 4 in Figure 1 (b) does not exceed three hop counts.

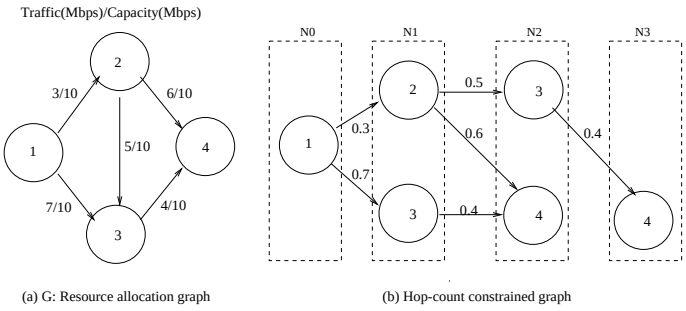


Fig. 1. Topology conversion example

Step 2 : **Finding multicast tree**
On the modified graph G', the link metric(c_{ij}) is given with the current utilization ratio (allocated bandwidth / link capacity). We propose a way of choosing multicast tree on the modified graph, $G' = (N', E')$. For each destination $t_k \in T_k$, we calculate the widest path from s_k to t_k by using Dijkstra's algorithm.

- widest path

The widest path is selected in order to minimize the usage of the bottleneck link, the link with the maximum utility. In this case, $dist(i)$, the cost of a node i, denotes the maximum link utility from source to the node.

$$dist(i) = min_{j \in S}(dist(i), max(dist(j), c_{ji})).$$

, where S is the set of nodes whose shortest path from source is already determined.
Next, we determines a destination r_k obeying the following constraint.

- destination r_k

$$(dist(t_k) | t_k \in T_k) \leq dist(r_k)$$

Then, we setup a path from s_k to r_k and reserve the bandwidth at each link along this path. In multicast algorithm, it is important to minimize the cost of the tree, especially network resources(like bandwidth). So we apply a heuristic for the directed Steiner tree problem. After finding this path, we set the cost of all the edges along this path to zero. Setting these edge costs to zero encourages future runs of Dijkstra's algorithm to use them. Until when all destinations are reachable from the source s_k, we repeat this process with the destination set T_k excluding r_k.

The detailed algorithm is explained in Figure 2. Figure 3 explains the simple result of the multicast tree selection. It is seen that the proposed heuristic algorithm constructs the multicast tree better than that using Dijkstra's shortest path algorithm. The proposed algorithm minimizes the maximum link utilization, while the resource utilization of each algorithm is the same.

Heuristic : *Find hop-count constrained multicast tree*
- Set d_k to be the traffic demand of traffic request k
- Set s_k to be a source of traffic request k
- Set T_k to be a set of destinations($t_k \in T_k$)
- Modify G to G' satisfying H_k hops;

while(T_k is not empty)
for each($t_k \in T_k$)
- Run widest path algorithm from s_k to t_k;
- Set $\alpha(t_k)$ to be the maximum link utilization when this path is setup;

endfor
- Set r_k to be the destination of maximum $\alpha(t_k)$;
- Reserve d_k to each link along the path from s_k to r_k;
- Set the link utility of all edges along this path to zero;
- Set T_k to $T_k - r_k$;

endwhile

Fig. 2. The proposed heuristic.

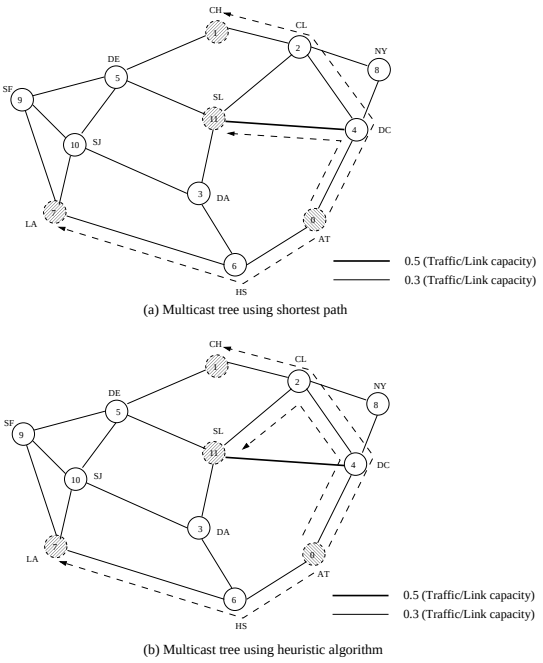


Fig. 3. The result of several multiple path calculation methods

C. Complexity Analysis

Proposed algorithm consists of two nested loops, inner loop for computation of widest paths, outer loop for each destination in the set of T_k. For each part, time complexity

is bound as follows. First, for the widest path problem, the best known bound is O($nlogn$) in a directed graph, where n is number of vertices in the graph. Algorithm the complexity of inner loop is bound by O(n^2logn), because the maximum number of destinations($|T_k|$) is n. The other remained code inside outer loop is bounded by O(n). Second, the outer loop is executed maximum n times. Hence, the worst-case time complexity of the proposed algorithm is bounded by O($n^3logn + n^2$).

IV. Performance Evaluation

A. Simulation Environment

The network topology shown in Figure 3 represents the abstract US backbone topology[18]. In this network condition, we generate one hundred random requests of traffic demands from one source. For each traffic request, the set of destinations is randomly selected by two cases. In the first case, the multicast session consists of small number of receivers (average 3.3 receivers per source). In the second case, it consists of many receivers (average 7.7 receivers per source). So, we treat both sparse and dense mode. Therefore, 2400 requests are tested in total. The duration of each traffic demand is exponentially distributed (ten seconds), and the inter-arrival time is randomly distributed between zero and two hundred seconds. The average traffic demand of each routing request is set to 5 Mbps. Among the simulation, the optimal mixed-integer programing solution is solved with CPLEX tool.

B. Simulation Result

The proposed heuristic algorithm is compared with the simple shortest path algorithm(SP) and the optimal MIP solutions, both in the traffic bifurcation case(B-OPT) and in the traffic non-bifurcation case(NB-OPT). In the traffic bifurcation case, the maximum path-count constraint is given as three. The maximum hop-count constraint(H_k) is given as zero or more additional to that of the minimum hop-count(H_{MH_k}) between an ingress and an egress router set.

Figure 4 shows the average rejected request ratio in sparse mode. In this simulation, we change the provisioned network capacity from 10% to 140% relatively. When the provisioned network capacity is fixed, the average rejected request ratio gap between heuristic and the shortest path algorithm shows a critical performance difference. Also, the required network capacity to satisfy the average reject request ratio below some fixed percentage is largely different between two algorithms. However, the heuristic algorithm shows the average rejected request ratio nearly same as the optimal solution.

Figure 5 shows the average rejected request ratio in dense mode. The result of dense mode shows the nearly same result as the sparse mode. We efficiently manage the network capacity through the proposed traffic engineering mechanism so that the more request of multicast routing can be accepted.

Table I and Table II present the average and the maxi-

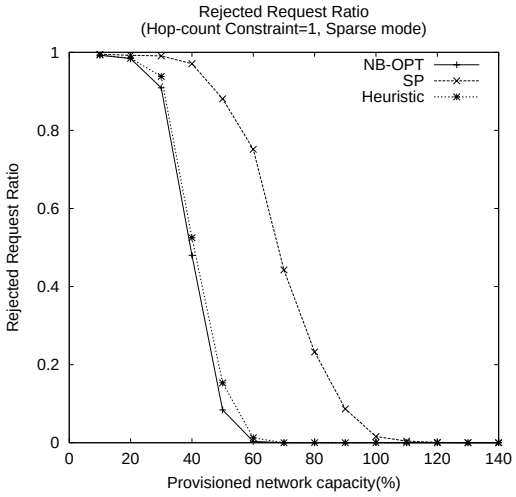


Fig. 4. The average rejected request ratio in sparse mode

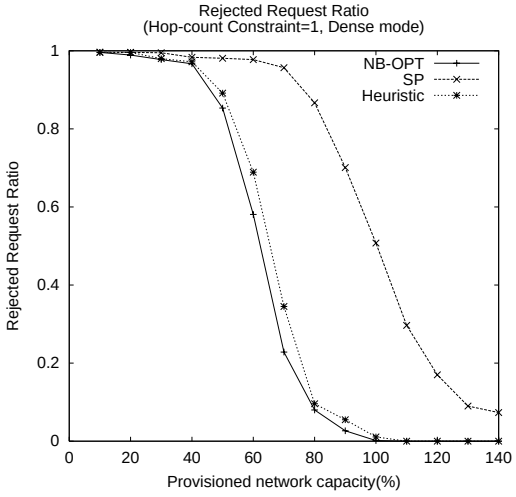
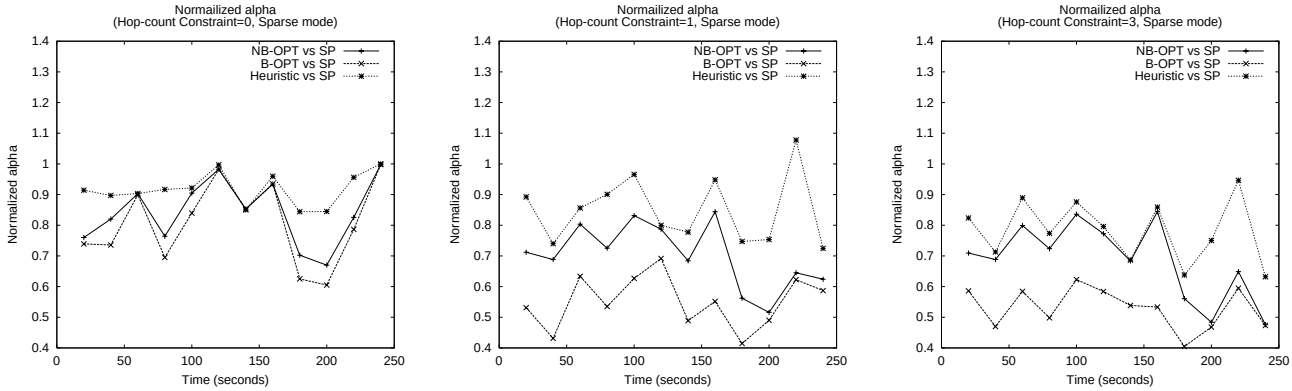


Fig. 5. The average rejected request ratio in dense mode

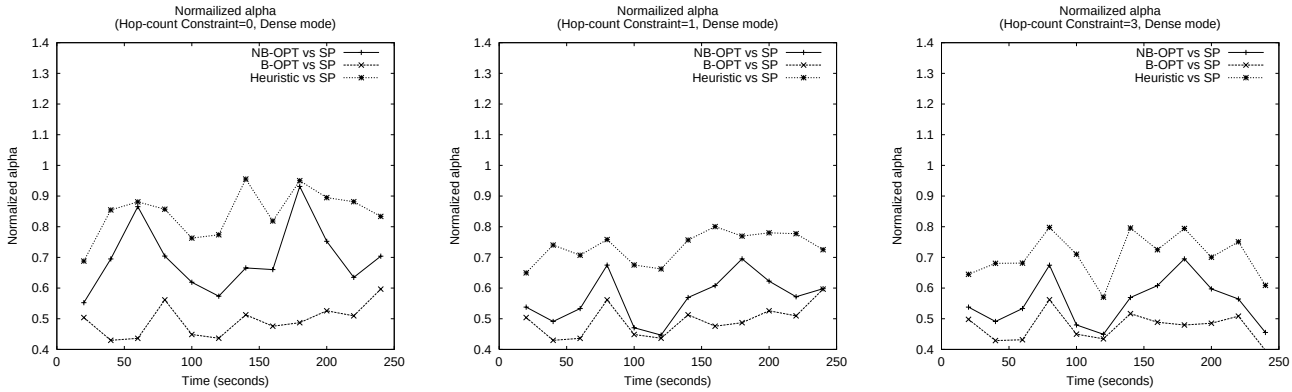
mum of α in sparse mode and dense mode respectively. The optimal solution of traffic non-bifurcation case not only almost halves the maximum of α of the shortest path algorithm(SP), but also improves the average α, in both sparse and dense mode. Also, we can see that the proposed heuristic shows similar results to optimal solutions. There is only a small difference less than 3% between heurustic and optimal solutions. Although the solution of traffic bifurcation case is better than that of traffic non-bifurcation case, the degree of improvement is very low relative to the overhead for implementation.

Figure 6 shows the *normalized* α which was obtained by dividing α of the both optimal MIP solution and α of our heuristic algorithm in Figure 2 by α of simple shortest path solution, in sparse mode. In Figure 6, we can see that the proposed heuristic constrained on zero, one or three additional hop performs better than the shortest path case (*normalized* $\alpha < 1$). When the hop-count constraint is changed, the optimal solution α of the traffic bifurcation



(a) Constraint(H=0) (b) Constraint(H=1) (c) Constraint(H=3)

Fig. 6. Maximum of link utilization (α) with the hop-count constraints in sparse mode



(a) Constraint($H = 0$) (b) Constraint($H = 1$) (c) Constraint($H = 3$)

Fig. 7. Maximum of link utilization (α) with the hop-count constraints in dense mode

TABLE I
MAXIMUM OF LINK UTILIZATION (α) IN SPARSE MODE

	SP	NB-OPT	B-OPT	Heuristic
AVG	13.23	8.84	7.20	10.24
MAX	30	15.72	14.37	18.32

TABLE II
MAXIMUM OF LINK UTILIZATION (α) IN DENSE MODE

	SP	NB-OPT	B-OPT	Heuristic
AVG	18.44	9.78	8.55	12.65
MAX	47.85	20.85	20.82	23.25

case is largely decreased.

Figure 7 shows the result of simulation in dense mode. In dense mode, each traffic request has a number of receivers. So total resource utility is increased, but we can see that the proposed heuristic also performs better than the shortest path case. Especially, Figure 7 (a), (c) show the performance improvement of the optimal MIP solutions since the hop-count constraint is changed from zero to three.

Since the optimal MIP formulation is $NP-hard$, we propose the heuristic algorithm whose complexity is polynominal time. Both simulations show results that the proposed heuristic algorithm efficiently minimizes the maximum link utilization, α and the rejected request ratio compared to the shortest path algorithm. However, the performance of both the optimal MIP and the proposed heuristic algorithm may not be further enhanced although the number of hop-count constraint increases more than three.

V. CONCLUSION

In this paper, we propose dynamic traffic engineering schemes for multicast routing that minimize the maximum of link utilization, α by finding multicast tree with the hop-count constraints. We classify this routing problem into two categories according to the use of a traffic splitting. We formulate each problem as the mixed-integer programming problem by using network flow model. Because finding solution of this optimal MIP problem is $NP-hard$, we propose the heuristic algorithm that calculates a constrained muticast tree in polynominal time. The simulation

results show that the proposed algorithm solves the problem of multicast routing with nearly same α as that of the optimal solution. Also we can see that the average rejected request ratio of the proposed algorithm outperforms the shortest routing algorithm and presents the nearly optimal result. Therefore, the proposed traffic engineering scheme is practical and will be useful for reducing the probability of congestion by minimizing the utilization of the most heavily used link in the network.

References

[1] D. Bertsekas, and R. Gallager, Data Networks, Prentice Hall, 1992

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

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

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

[5] M. Charikar, C. Chekuri, T. Cheung, Z.Dai, A. Goel, S.Guha, and M. Li, "Approximation Algorithms for Directed Stiner Problems," SODA, 1988.

[6] R. Guerin, D. Williams, A. Orda. "QoS Routing Mechanisms and OSPF Extensions," Globecom 97.

[7] N. S. V. Rao, and S. G. Batsell, "QoS Routing Via Multiple Paths Using Bandwidth Reservation," INFOCOM'98

[8] A. Shaikh, J. Rexford, and K. G. Shin, "Load-Sensitive Routing of Long-Lived IP Flows," SIGCOMM'99

[9] P. Bhaniramka, W. Sun, and R. Jain, "Quality of Service using Traffic Engineering over MPLS: An Analysis," LCN'2000

[10] H. Saito, Y. Miyao, and M. Yoshida, "Traffic Engineering using Multiple Multipoint-to-Point LSPs," INFOCOM'2000

[11] Y. Wang, and Z. Wang, "Explicit Routing Algorithms for Internet Traffic Engineering," ICCCN'99

[12] Y. Seok, Y. Lee, Y. Choi, and C. Kim, "Dynamic Constrained Multipath Routing for MPLS Networks," ICCCN'2001

[13] E. Dinan, D. O. Awduche, and B. Jabbari, "Analytical Framework for Dynamic Traffic Partitioning in MPLS Networks," ICC'2000

[14] M. Kodialam, and T. V. Lakshman, "Minimum Interference Routing with Applications to MPLS Traffic Engineering," INFOCOM'2000

[15] M. Kodialam, T. V. Lakshman, and S. Sengupta, "Online Multicast Routing with Bandwidth Guarantees: A New Approach using Multicast Network Flow,", SIGMETRICS'2000

[16] Z. Wang, Y. Wang, and L. Zhang, "Internet Traffic Engineering without Full Mesh Overlaying," INFOCOM'2001

[17] A. Elwalid, C. Jin, S. Low, and I. Widjaja, "MATE: MPLS Adaptive Traffic Engineering," INFOCOM'2001

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