Dynamic Constrained Multipath Routing for MPLS Networks

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—**Multipath routing employs multiple parallel paths between a traffic source and destination in order to relax the most heavily congested link in Internet backbone. A Large bandwidth path can be easily set up too. Although multipath routing is useful, the total network resources, i.e. sum of link bandwidths consumed, could be wasted when the acquired path is larger(in terms of number of hops) than the conventional shortest path. In addition, even though we can accommodate more traffic by establishing more paths between the same node pair, it is advised to limit the number of paths for practical reason such as manageability. This paper presents a heuristic algorithm for hop-count and path-count constrained dynamic multipah routing. The objective we adopted in this paper is to minimize the maximum of link utilization. We also obtain the traffic split ratio among the paths, for routers based on traffic partitioning by hashing at flow level. The extensive simulation results show that the proposed algorithm always minimizes the maximum of link utilization and reduces the number of blocked requests.**

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 multicommodity network flow formulation, that leads to splitting traffic over multiple paths between source-destination pairs.

Multipath routing provides increased bandwidth, and the network resources are more efficiently used than the single shortest path 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 one, which is known as Equal-Cost Multi-Path(ECMP) routing. This is explicitly supported by several routing protocols such as Open Shortest Path First(OSPF)[3] and Intermediate System to Intermediate System(IS-IS)[4]. Some router implementations allow equal-cost multipath with Routing Information Protocol(RIP) and with some other routing protocols. In Multi-Protocol Label Switching (MPLS) networks[2] where IP packets are switched through the pre-established Label Switched Paths (LSPs) by signaling protocols, multiple paths can be used to forward packets belonging to the same "forwarding equivalent class (FEC)" by explicit routing.

However, multiple paths 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 multipath routing scheme in order not to waste bandwidth. In addition, as the number of paths will be restricted between a source-destination pair in the real network topology, the maximum path-count constraint should be considered in multipath routing.

This paper proposes a practical heuristic algorithm that finds hop-count and path-count constrained multiple paths that minimize the maximum of link utilization, while satisfying the requested traffic demand. 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.

Traffic split ratios for the calculated paths are also obtained from the proposed algorithm. The split ratio is fed to the routers for dividing the traffic of the same source-destination pair to multiple paths. Partitioning a traffic demand will be done by adjusting the output range of the hashing function[5]. In multipath routing, routers should also provide a flow-level forwarding mechanism not to cause the out-of-order packet delivery problem which will degrade end-to-end performance.

Through splitting a traffic demand, it is expected that the maximal revenue can be achieved by increasing the probability that more traffic demand requests will be accepted in the future. Also, the utilization of the total network resource will be maximized, while guaranteeing requested bandwidth by reservation. For the sake of users, it would be better that the queueing delay is minimized, especially for expedited forwarding (EF) in differentiated services (Diffserv)[13]. However, when minimizing the queue-

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.

ing delay for a newly arrived request, 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. This is not suitable in the real-time environment, even causes a lot of path disruptions. Therefore, instead of minimizing the sum of the queueing delay on a link, we use the objective of minimizing the maximum of link utilization which makes little difference in routing performance[1].

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

In connection-oriented networks, [6] analyzed the performance of multipath routing algorithms and showed that the connection establishment time for a reservation is significantly lowered in the multipath case. However, they did not really consider the path computation problem. [7] 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 their work, only connection or call-level, not flow-level routing and forwarding are considered. In [8], Quality-of-Service(QoS) routing via multiple paths under 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 the dynamic buffer adjustment at the receiver. The enhanced routing scheme for load balancing by separating long-lived and short-lived flows is proposed in [9], and it is shown that congestion can be greatly reduced. In [10], 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 path calculation problem.

For the MPLS network, a traffic engineering method using multiple multipoint-to-point LSPs is proposed in [11], where backup routes are used against failures. Hence, the alternate paths are used only when primary routes do not work. In [12], the traffic bifurcation linear programming (LP) problem is formulated and heuristics for the non-bifurcating problem are proposed. Although [12] minimizes the maximum of link utilization, it does not consider the total network resources and constraints. Wang and et al. showed that the traffic bifurcation LP problem can be transformed to the shortest path problem by adjusting link weights in [15].

In [14], 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.

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

III. Hop-count and Path-count Constrained Multipath Routing

A. Problem Definition

In this section, we define the hop-count and path-count constrained routing problem in mixed integer programming (MIP) formulation.

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 (t_k). For each traffic demand, there are the maximum number of hop counts constraint, H_k[1] and the maximum number of multiple paths constraint, P. The variable $X^k_{ij}(h,p)$ represents the fraction of the traffic demand k assigned to link (i, j) where j is h hops far from s_k and the path p, $1 \leq p \leq P_k$, includes (i, j) link. The integer variable $Y^k_{ij}(h,p)$ tells whether link (i, j) is used or not for the path p, where j is within h hops from s_k, for the traffic demand k. Let d_k be a scaling factor to normalize the total traffic demand from the source to become 1. The mixed integer programming (MIP) problem is formulated as follows.

$$\textit{Minimize } \alpha$$
$$\text{subject to}$$

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

$$\sum_{j:(i,j)\in E} X^k_{ij}(h+1,p) - \sum_{j:(j,i)\in E} X^k_{ji}(h,p) = 0, \tag{2}$$
$$k \in K, i \neq s_k, t_k, 1 \leq p \leq P_k, 1 \leq h < H_k$$

$$\sum_{h=1}^{H_k} \sum_{j:(j,i)\in E} X^k_{ji}(h,p) = 1, k \in K, i = t_k, \forall p \tag{3}$$

$$\sum_{p=1}^{P_k} \sum_{h=1}^{H_k} \sum_{k\in K} d_k X^k_{ij}(h,p) \leq c_{ij}\alpha, \forall (i,j) \in E \tag{4}$$

$$\sum_{j:(i,j)\in E} Y^k_{ij}(h,p) = 1, k \in K, i \neq t_k, \forall h, p \tag{5}$$

$$X^k_{ij}(h,p) \leq Y^k_{ij}(h,p), \forall k, h, p, i, j \tag{6}$$

$$\textit{where}, \quad 0 \leq X^k_{ij}(h,p) \leq 1, 0 \leq \alpha,$$
$$Y^k_{ij}(h,p) \in \{0,1\}, h, p \in Z.$$

The objective is to minimize the maximum of link utilization, α. Constraint (1) says that the sum of total outgoing traffic of each path over the first hop from the source is 1,

[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. H is additional hop-count that is added to H_{MH_k}.

and the all node over the first hop from the source never receive the traffic without 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 the amount of total traffic comes into the destination is 1, which is the same as that of goes out of the source. Constraint (4) means that the maximum link utilization among all paths for traffic demand k is α. Constraint (5),(6) shows that for a path, there is only one outgoing edge from a node. This problem is $NP\!-\!hard$ because it includes the constrained integer variable.

B. Proposed Heuristic

We propose a heuristic algorithm to find multiple paths and their split ratios for each traffic demand request on the ingress and egress pair. The proposed algorithm consists of three parts: 1) modifying the original graph to the hop-count constrained one, 2) finding path-count constrained multiple paths, and 3) calculating the load split ratios for multiple paths.

Step 1 : **Hop-count constrained graph conversion**
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 Fig. 1. Fig. 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 Fig. 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 Fig. 1 (b) does not exceed three hop counts.

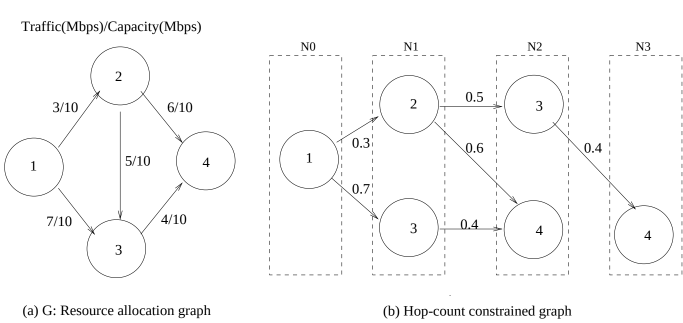


Fig. 1. Topology conversion example

Step 2 : **Finding multiple paths**
On the modified graph G', the link metric(c_{ij}) is given with the current utilization ratio (allocated bandwidth / 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[18]. 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

The M shortest paths are obtained by selecting the adjacent node with the minimum cost. The cost of a node i reached from source is denoted as $dist(i)$,

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

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

- M widest paths

The M widest paths are selected in order to minimize the usage of the bottle neck 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.

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

Step 3 : **Calculating load split 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 to minimize the amount of used total resources(i.e., the number of links and routers). Then, the path with the smallest hop counts is selected among the 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 repeated until there remains no traffic demand or there exists no available path. Yet, if the traffic demand is not fully assigned, it is allocated on M multiple paths in proportion to the sum of the link utilization of each path.

The detailed algorithm is explained in Fig. 2. Therefore, the proposed heuristic is either the shortest path(SP) based algorithm or the widest path (WP) based one. Fig. 3 explains the results of the multiple paths and their load split ratios on the graph in Fig. 1 (a). It is seen that the paths found by the single shortest path algorithm(SSP) wastes more network resources, while the proposed algorithm can find the optimal solution.

IV. Performance Evaluation

A. Simulation Environment

The network topology shown in Fig. 4 represents the abstract US backbone topology[17]. On this topology we assume that the background traffic demands are given as in [17].

In this network condition, we generate ten random requests of traffic demands between two nodes selected ran-

Heuristic : *Find constrained multiple paths and their split ratios*

- Set α to be current maximum link utilization;
- Set d_k to be the normalized traffic demand k, $0 < d_k < 1$;
- Modify G to G' satisfying H_k hops;
- Find M shortest(widest) paths from s_k to t_k;
- Set P to be a set of candidate multiple paths previously found;
- 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-count path of P_k;
  - Set $d_k(p)$ to be $min(\alpha - \alpha_M, d_k)$;
  - Assign $d_k(p)$ to each link along the path p;
  - Set d_k to be $d_k - d_k(p)$;
  - Delete p from P;

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

endwhile

Fig. 2. The flow of proposed Heuristic.

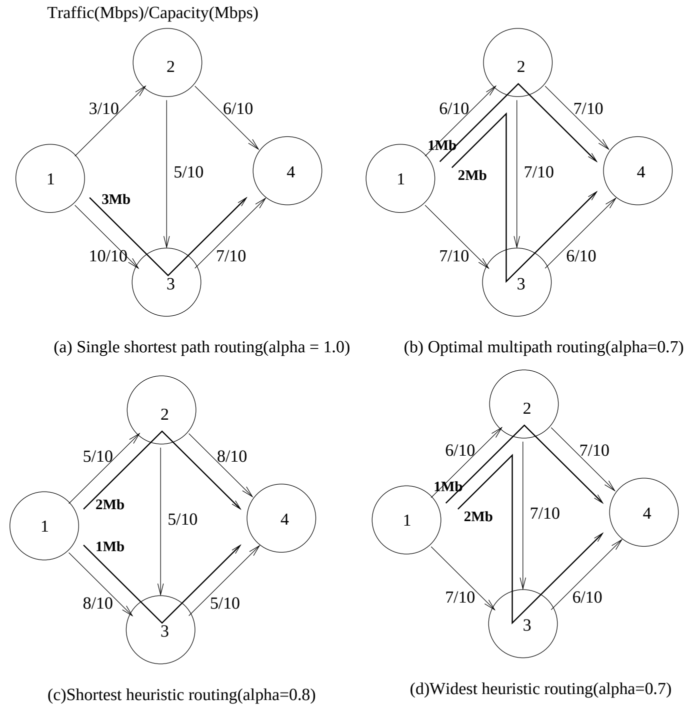


Fig. 3. The result of several multiple path calculation methods

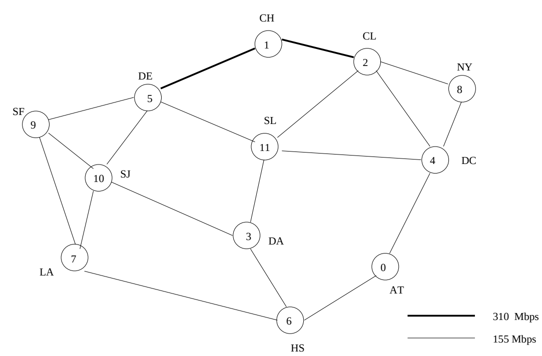


Fig. 4. Abstract US Network

domly. Therefore, 1,320 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 one hundred seconds. The average rate of each traffic demand is set to 2 Mbps.

The proposed heuristics are compared with the simple shortest path algorithm and the optimal MIP solution. 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. As the number of multiple paths(P_k) will be usually restricted, the path-count constraint is set to be one of 1, 3, and 5.

As shown in Table I with the average of α, the shortest path(widest path) based heuristic increases only 4.12(3.01) % when compared to the optimal solution by MIP formulation. Yet, the maximum link utilization by the single shortest path(SSP) algorithm increases by 11.34 %.

Fig. 5 shows the *normalized* α which was obtained by dividing α of our algorithm in Fig.2 by α of optimal solution of the MIP. In Fig. 5 (a), it is seen that although only one additional hop is constrained on the single path, the proposed heuristic performs better than the shortest path. As the path-count constraint is increased to three (Fig. 5 (b)), the maximum of link utilization is greatly reduced. Also, even when the hop-count constraint is zero (i.e., the equal cost multiple paths) (Fig. 5 (c)), multiple paths are well utilized, giving the similar α to the case with $(H, P_k) = (1, 3)$

However, the performance of the proposed algorithm may not be more enhanced although the number of hop-count or path-count constraint increases, because many multiple paths are overlapped. With the path-count constraint of five(Fig. 6 (a)) or with H of two (Fig. 6 (b)), the proposed algorithm shows the similar result under $(H, P_k) = (1, 3)$.

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. 7. It is shown that the number of rejected requests is reduced by the proposed heuristic.

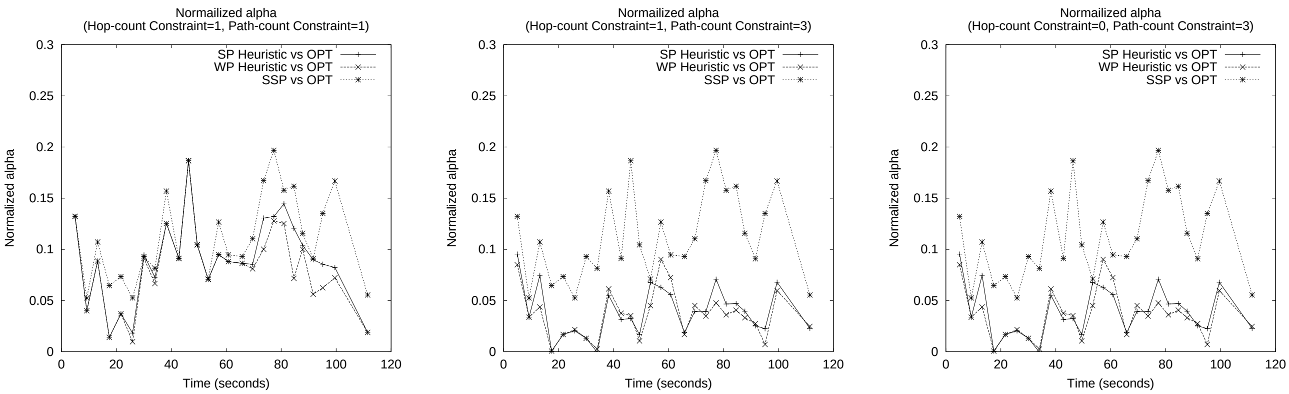
V. Conclusion

In this paper, we propose dynamic multipath traffic engineering schemes for MPLS networks that minimize the maximum of link utilization, α by finding multiple paths with the hop-count and path-count constraints. The proposed heuristic approximates the traffic bifurcation MIP problem that is $NP-hard$, by calculating constrained mul-

TABLE I

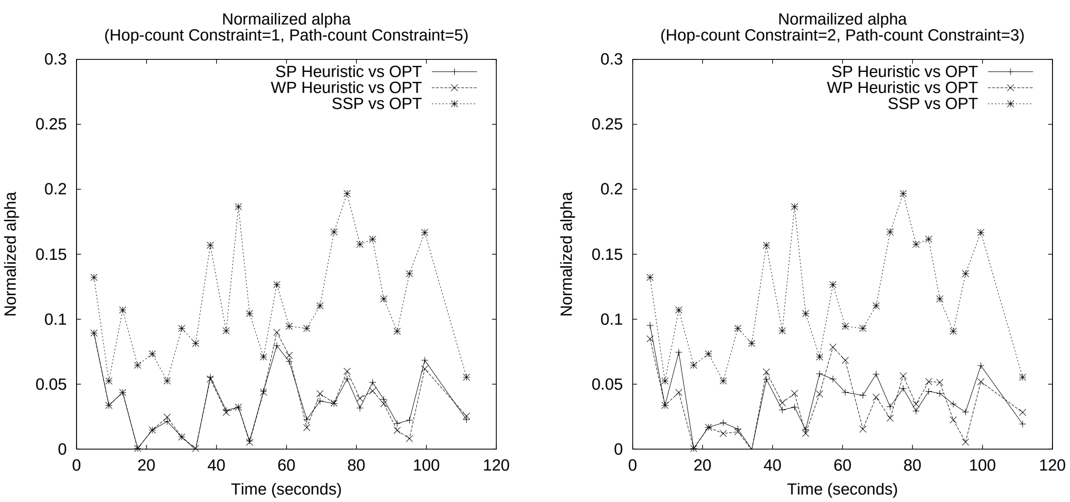
Average of Maximum of Link Utilization (α)

	SSP	SP Heuristic	WP Heuristic	OPT
α	1.08	1.01	1.00	0.97



(a) (H, P_k) = (1,1) (b) (H, P_k) = (1,3) (c) (H, P_k) = (0,3)

Fig. 5. Maximum of link utilization (α) with the hop-count constraint and the path-count constraint



(a) (H, P_k) = (1,5) (b) (H, P_k) = (2,3)

Fig. 6. Maximum of link utilization (α) with the hop-count constraint and the path-count constraint: when α is not more enhanced

tiple paths and their split ratios efficiently in polynomial time. The simulation results show that the proposed algorithm solves nearly the same α of the optimal solution even with only one additional hop and three paths because many new candidate paths are derived. In addition, the number of blocked requests is reduced. 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] Z. Cao, Z. Wang and E. Zegura, "Performance of Hashing-Based Schemes for Internet Load Balancing," INFOCOM'2000

[6] 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

[7] S. Bahk, and M. Zarki, "Dynamic Multi-path Routing and How it Compares with other Dynamic Routing Algorithms for High Speed Wide Area Networks," Computer Communications Review, vol. 22, no. 4, Oct. 1992

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

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

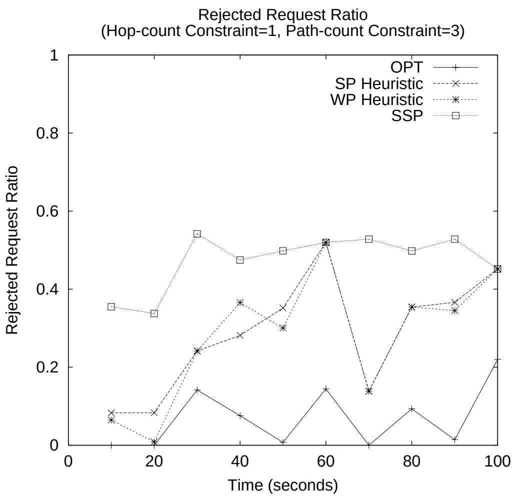


Fig. 7. Rejected Traffic Demand Request Ratio

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

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

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

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

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

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

[18] E. L. Lawler, Combinatorial Optimization: Newtorks and Matroids, Holt, Rinehart and Winston, 1976