

A New Multi-path Routing Methodology Based on Logit Type Assignment



Yudai Honma
and Masaki Aida
Faculty of System Design
Tokyo Metropolitan University
Hino, Tokyo 191-0065
Email: (yudai, maida)@sd.tmu.ac.jp

Hideyuki Shimonishi
and Atsushi Iwata
NEC System Platforms Research Laboratories
Kawasaki, Kanagawa 211-8666
Email: h-shimonishi@cd.jp.nec.com
Email: a-iwata@ah.jp.nec.com

Abstract—We present a new multi-path routing methodology called MLB-routing based on multinomial logit model, which is well known as the random utility theory. The key concept of the study is to incorporate multiple paths from same origin to destination, and distribute packets followed by the multinomial logit type probability. Since MLB-routing is pure multi-path routing, it reduce the severe convergence to same links and increases the bandwidth utilization in the network. Compared to the existing multi-path routing schemes that select pre-determined alternate paths, the proposed method can dynamically distribute packets to every possible paths and thus is more efficient than them. Furthermore, it should be mentioned that this methodology could be implemented as both link-state protocol and distance-vector protocol. Therefore, it has enough affinity for present Internet mechanism. Using simulations, we have also shown that this methodology produces more efficient use of network and causes significant improvements in end-to-end delays and jitter times.

I. INTRODUCTION

Lots of routing mechanisms in the Internet network have been based on single-shortest path routing in various metrics. This strategy seems rational because all packets on the networks should be delivered as soon as possible. However, it is cautious that some of same links are heavily used as a shortest paths in different node pair. Hence, routing protocols based on single-shortest path method cause the severe convergence of packets to same links, and results to traffic congestion.

Traditionally, the main methodology to reduce traffic congestion is end-to-end control. It is well known that TCP mechanism [10], which is the most commonly used in Internet, has implemented end-to-end congestion controls. It basically means to control the “quantity” of packets on the networks. On the other hand, there are several idea to solve the traffic congestion by controlling the “routing” methodology. For example, PB-routing protocol proposed by Basu et al. considers the traffic potential fields on the network, and tries to avoid the congested links by making a detour to an alternate route [3]. However, even if they consider link traffic (i.e. adopting the traffic-aware routing), the fact of “single-shortest path” routing is not changed at all. That is, same links would be often used, and some alternative links become new bottlenecks.

Meanwhile, the techniques of multi-path routing have been focused as another methodology to ease traffic congestion. The basic idea of multi-path routing is to split the packets with

the same origin and destination along multiple alternate paths. Hence, multi-path routing encourages the effective utilization of network resources. First, the number of packets per each path is decreased, so the emergence of bottleneck will be prevented. In addition, multi-path routing means to use more links than single routing, and it is favorable in terms of the load-balancing of traffics. Consequently, multi-path routing contributes the various things, such as the improvement of the quality of service (QoS), minimizing delay, the increase of reliability of delivery, and so on.

It is well known that some protocols such as OSPF and BGP implement equal-cost multi-path (ECMP) [7], [11]. However, there is less likelihood that multiple paths with equal distance exist between each source-destination pair, which means the full connectivity of the network is still not used for load-balancing. Therefore, other multi-path routing methodologies, which also use “near-optimal paths”, have been discussed in a number of researches (see e.g. [2], [4], [8] and the references therein). Although, to calculate the multi-path routing, it is necessary to enumerate the multiple candidate paths to split. Hence, lots of multi-path routing protocols adopted the idea of the link-state protocols, and only a few researches are implemented as distance-vector protocols [14], [15].

In this paper, we present a new multi-path routing methodology which is based on *multinomial logit model* (MNL). We call this methodology “(multinomial) logit based routing”, or “MLB-routing”. Since MLB-routing is pure multi-path routing, it is supposed to reduce the severe convergence to same links and increases the bandwidth utilization in the network. It also should be mentioned that this methodology could be implemented as both link-state protocol and distance-vector protocol. Therefore, it has enough affinity for present Internet mechanism.

The key idea of MLB-routing is to determine the route of packets based on MNL. MNL is a classical model of *random utility theory*, which is used to assess the individual’s choosing behavior [12]. Since the behavior of “choosing something” is brought on by various psychological factors, it is difficult to estimate exactly what to be selected. Hence, in random utility theory, we estimate the “probability” to be chose toward to each alternatives, and it is the most interesting point of the

theory.

MLB-routing has many advantages over traditional multi-path routing system. There are enormous candidate paths between same origin-destination pair. In lots of existing multi-path routing protocol, nevertheless, some of paths are only considered because of limitations of their algorithms. Meanwhile, it is noteworthy that MLB-routing aims to consider “all of paths”. Thus, it contributes to more effective utilization of network resources. It seems impractical to consider all possible paths in large scale networks. However, this strategy paradoxically decreases the computational complexity, and enables us to implement MLB-routing as distance-vector protocol. This discussion results from the equivalency between MNL and Markov model in network flow [1].

Furthermore, owing to the property of MNL, in MLB-routing, the choice probability for each path decrease exponentially as the length of path becomes longer than shortest one. It means that high utility paths are frequently selected and less utility ones are not, which is a favorable structure we hope. Hence, we can avoid another problem that impractical paths are often used, because the probability for impractical path becomes quite low value. We believe that the general framework of MLB-routing could be adapted for various situation through the careful arrangement of algorithm.

The rest of the paper is organized as follows. In the next section, we describe the MLB-routing model in greater detail and clarify the structure of system analytically. This is followed by Sections 3 where we describe our implementation of the routing and evaluate its performance. Finally, we conclude and mention about future works in Section 4.

II. THE LOGIT BASED ROUTING METHODOLOGY

In this section, We present the basic idea of MLB-routing methodology. We first explain the brief summary of multinomial logit model (MNL) which is well known in random utility theory. We then describe how to apply the MNL to the routing system, and formulate MLB-routing system with an algorithm to calculate the routing effectively.

A. Multinomial Logit Model

Let us suppose a situation that an individual is going to choose an alternative j from an alternative set $N \in \{1, 2, \dots, J\}$. Furthermore, we assume that the individual recognizes the following utility

$$U_j \stackrel{\text{def}}{=} [\text{utility when he chooses the alternative } j], \quad (1)$$

and the individual is supposed to choose the alternative to maximize his utility. Our first objective is to analyze which alternative the individual would choose.

If we “precisely” know his utility U_j , then we can figure out which alternative the individual would choose. However, it is impossible to figure out the individual’s utility without the least difference, because utility U_j would be affected to various elements such as his character, preference, and so on. Therefore, it is undeniable that some errors are mixed in “the utility estimated by researchers”.

In view of these facts, let us assume the utility U_j when an individual choose the alternative j as follows:

$$U_j = V_j + \varepsilon_j, \quad (2)$$

where V_j is a function estimated by a researcher, and ε_j is a random variable. V_j depends on the observed characteristics of the alternative j , and generally $U_j \neq V_j$. Hence we introduce the term ε_j .

The definition stated above indicates that each utility U_j is a random variable, so it is impossible to identify one and only alternative the individual would choose. This fact implies that we should analyze the choice behavior as a stochastic event. Now, choosing the alternative j means that U_j is the highest value. Therefore, we can formulate the probability P_j that the individual choose the alternative j as follows:

$$P_j = \Pr [U_j > U_{j'} \text{ (for } \forall j' \in N, j' \neq j)]. \quad (3)$$

Specific form of (3) is identified when we define the distribution of random variable ε_j . the procedure explained above is the formulation of random utility model.

In the MNL, it is assumed that the random factors of utilities $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_J$ are independent random variables from the identical Gumbel distribution, where the Gumbel distribution with parameters (α, γ) is

$$\langle \text{c.d.f.} \rangle F(\epsilon) = \exp[-\exp[-\gamma(\epsilon - \alpha)]], \quad (4)$$

$$\langle \text{p.d.f.} \rangle f(\epsilon) = \gamma \exp[-\gamma(\epsilon - \alpha)] \exp[-\exp[-\gamma(\epsilon - \alpha)]]. \quad (5)$$

In this case, we can obtain the probability P_j as follows:

$$P_j = \frac{\exp[\gamma V_j]}{\sum_{j \in N} \exp[\gamma V_j]}. \quad (6)$$

This is the probability derived by the MNL. As understood by (6), when $\gamma = \infty$, only the best alternative is used, and if $\gamma = 0$, then all alternatives are used equally likely.

B. MLB-Routing Assignment

As described above, MNL is not only simple but also quite persuasive model, so it is used in various fields. One typical example is a routing behavior in transportation network model. The general objective of transportation network model is to describe transportation flows in road networks, and therefore it is an essential issue to predict driver’s routing behavior [9]. MNL is also utilized to describe such routing behavior, and the number of researches have been discussed for effective calculation. We incorporate their researches and propose MLB-routing in the following.

In order to describe the MLB-routing protocol, we first define a system model assumed in this study. We assume a network of nodes connected by directed links as a directed graph $G = (N, E)$. The set of nodes in the networks is labeled by the N , and the set of edges E . Furthermore, e_{ij} index a directed edge from vertex i to vertex j with cost metric c_{ij} . For the rest of this paper, we shall use the terms nodes (links)

and vertices (edges) interchangeably. Each node v can act as a traffic source and /or sink. Finally, every node v has a set of $Z(v)$ neighbors denoted by $nbr(v)$.

Now we explain the concept of MLB-routing protocol. let us consider any packet p at a origin node o going to destination node d . There are not only the shortest path but also several way of paths to reach the destination d , and we try to use “all” of paths (theoretically). However, it doesn't mean to use all route equally; naturally the shortest path should be most often used, and as the longer the cost of path becomes, the less the path used. To incorporate this idea, let us define the probability that packet p use the r th path P_r^{od} as follows:

$$P_r^{od} = \frac{\exp[-\gamma C_r^{od}]}{\sum_{r \in \Phi^{od}} \exp[-\gamma C_r^{od}]} \quad (7)$$

The above expression comes from the definition of MNL, where C_r^{od} denotes the costs for the r th path between nodes o and d , and Φ^{od} is the alternative set of paths from node o to d . In this study, the each pass cost is calculated by the summation of link cost, namely

$$C_r^{od} \stackrel{\text{def.}}{=} c_{ov_1} + \sum_{l=1}^{\Lambda-1} c_{v_l v_{l+1}} + c_{v_\Lambda d} \quad (8)$$

where r th pass is $o \rightarrow v_1 \rightarrow v_2 \rightarrow \dots \rightarrow v_\Lambda \rightarrow d$.

Now let us discuss how to define the alternative set Φ^{od} . The natural definition of the alternative set is “simple path”; the path which does not pass through same link more than once. In this study, however, we do not introduce any restriction, and consider all of path including “cyclic path”. This strategy has a possibility that the definition generates very long detours which should not be assigned. Furthermore, it seems to be difficult to calculate the probability in a large scale network.

On the contrary, this definition is made from the viewpoint of the computational feasibility. As shown in later sections, this definition enables us to calculate the stochastic assignment in large scale networks. Moreover, the definition is useful in some iterative algorithms for solving the assignment which means to expand to distance-vector protocol. From next subsection, we discuss the structure of MLB-routing system, and effective algorithms for calculation.

C. The Structure of MLB-routing Model

Let us now discuss the structure of the MLB-routing system. First, we explain how to allocate packets to various paths. Obviously we can allocate packets to any arbitrary paths by source routing. However, to apply MLB-routing to the present Internet network, it had also better to setup MLB-routing as hop-by-hop routing system.

Fortunately, it is proved that the network flow generated by (7) is equivalent to the following Markov assignment flow [1]:

$$p(j|i) = \exp[-\gamma c_{ij}] \frac{W_{jd}}{W_{id}}, \quad (9)$$

$$W_{id} \stackrel{\text{def.}}{=} \sum_{r \in \Phi^{id}} \exp[-\gamma C_r^{id}], \quad (10)$$

where $p(j|i)$ is a transition probability that a packet in node i select the next hop j . This formulation satisfy the condition

$$\sum_{j \in nbr(i)} p(j|i) = \frac{\sum_{j \in nbr(i)} \exp[-\gamma c_{ij}] W_{jd}}{W_{id}} = 1 \quad (11)$$

which is required to be regarded as the conditional probabilities. Furthermore, the equivalency between two assignments is easily confirmed by

$$\begin{aligned} P_r^{od} &= p(v_1|o) \times \prod_{l=1}^{\Lambda-1} p(v_{l+1}|v_l) \times p(d|v_\Lambda) \\ &= \exp[-\gamma c_{ov_1}] \frac{W_{v_1 d}}{W_{od}} \times \exp[-\gamma c_{v_1 v_2}] \frac{W_{v_2 d}}{W_{v_1 d}} \times \dots \\ &\quad \times \exp[-\gamma c_{v_{\Lambda-1} v_\Lambda}] \frac{W_{v_\Lambda d}}{W_{v_{\Lambda-1} d}} \times \exp[-\gamma c_{v_\Lambda d}] \frac{W_{dd}}{W_{v_\Lambda d}} \\ &= \exp\left[-\gamma \left(c_{ov_1} + \sum_{l=1}^{\Lambda-1} c_{v_l v_{l+1}} + c_{v_\Lambda d}\right)\right] \frac{W_{dd}}{W_{od}} \\ &= \frac{\exp[-\gamma C_r^{od}]}{\sum_{k=1}^{\infty} \exp[-\gamma C_r^{od}]} \end{aligned} \quad (12)$$

The existence of equivalent Markov assignment implies that we can apply the simple rule to all nodes (routers). More clearly say, if each node choose the next node to pass through the packet following (9), the network flow consequently equal to the assignment which discussed in II-B.

Since the above method distributes packets to every possible next nodes in a packet-by-packet manner, there would be a possibility of out-of-order packet delivery. In this paper, we will not further argue this problem, but we believe the problem can be solved by, for example, re-sequencing at the end host or using special random functions that select same next node for packets of the same flow.

Now, please note that $p(j|i)$ does not depend on the node of origin. It is confirmed that there are no subscript notation related to origin in (9). Thus, if each router follows the above Markov assignment, every packets which start from arbitrary node spontaneously hold by (7). Generally, existing routing table also does not depend on the origin, so it is quite favorable feature of the MLB-routing. The example of the routing table assumed in the MLB-routing is shown in Table I. Since it is multi-path routing, there are multiple candidate nodes for next hop even if packets go to same destination.

D. How to Calculate W_{id}

To determine the value of $p(j|i)$, we have to calculate the W_{id} which is included in $p(j|i)$. Since the summation in W_{id} means to consider all of path from node i to d , it is hard to derive them natively.

Consider a matrix \mathbf{A} with N rows and N columns, whose $[i, j]$ element is given as follows:

$$a_{ij} = \begin{cases} \exp[-\gamma c_{ij}] & (\text{if node } i \text{ to } j \text{ is connected}) \\ 0 & (\text{otherwise}) \end{cases} \quad (13)$$

TABLE I
AN EXAMPLE OF ROUTING TABLE

destination	next router	probability	weight
network A	router j_1	$p_A(j_1 i)$	W_{iA}
	\vdots	\vdots	
	router j_N	$p_A(j_N i)$	
network B	router j_1	$p_B(j_1 i)$	W_{iB}
	\vdots	\vdots	
	router j_N	$p_B(j_N i)$	
network C	router j_1	$p_C(j_1 i)$	W_{iC}
	\vdots	\vdots	
	router j_N	$p_C(j_N i)$	

The element of A^2 yields

$$\begin{aligned}
 a_{ij}^{[2]} &= \sum_{k=1}^N a_{ik} a_{kj} \\
 &= \sum_{\{k|e_{ik}, e_{kj} \in E\}} \exp[-\gamma(c_{ik} + c_{kj})] \\
 &= \sum_{r \in \Phi_2^{ij}} \exp[-\gamma C_r^{ij}], \quad (14)
 \end{aligned}$$

where Φ_L^{ij} is the set of paths which connect nodes i and j in L steps. Similarly, the typical element of A^L is given by

$$a_{ij}^{[L]} = \sum_{r \in \Phi_L} \exp[-\gamma C_r^{ij}]. \quad (15)$$

From the definition, we have

$$\Phi^{ij} = \Phi_1^{ij} \oplus \Phi_2^{ij} \oplus \dots \quad (16)$$

Consequently, we can obtain the value of W_{id} as follows:

$$\mathbf{W} = \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \dots \quad (17)$$

where the element of matrix \mathbf{W} is

$$W_{ij} = \sum_{r \in \Phi^{ij}} \exp[-\gamma C_r^{ij}] \quad (18)$$

Furthermore, if the matrix \mathbf{A} satisfied the Haskins-Simon condition, the following formula is obtained:

$$[\mathbf{I} - \mathbf{A}]^{-1} = \mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \dots \quad (19)$$

Therefore, we derive the value of W_{ij} as follows:

$$\mathbf{W} = [\mathbf{I} - \mathbf{A}]^{-1} - \mathbf{I}. \quad (20)$$

As indicated above, if we know the link weighted matrix \mathbf{A} , then transition probability $p(j|i)$ is clearly determined. It means that all routers (nodes) will work appropriately for MLB-routing. Since the link weighted matrix \mathbf{A} is derived by assuming a link-state routing protocol (such as OSPF [7]), we obtained the methodology to implement MLB-routing as link-state routing protocol.

E. As a Distance-vector Routing Protocol

From the above discussion, we represented the methodology to implement MLB-routing as a link-state protocol. Meanwhile, such as RIP [6], distance-vector routing protocol which is based on Bellman-Ford algorithm is still a major method. Hence, it is worth enough to implement MLB-routing as distance-vector protocol. We explain the methodology to achieve this idea in the following.

The idea of distance-vector routing protocol is that exchanging information (such as routing tables) between neighborhood routers. Thus, we present how to calculate W_{id} (and it means $p(j|i)$) by iterative exchange of information. It is easy to understand that

$$\begin{aligned}
 \mathbf{W} &= \mathbf{A} (\mathbf{I} + \mathbf{A} + \mathbf{A}^2 + \mathbf{A}^3 + \dots) \\
 &= \mathbf{A} (\mathbf{I} + \mathbf{W}) \quad (21)
 \end{aligned}$$

is satisfied from (17). Furthermore, it is equivalent to

$$W_{id} = \sum_{j \in nbr(i)} \exp[-\gamma c_{ij}] (W_{jd} + \delta_{jd}), \quad (22)$$

where δ_{jd} is Kronecker delta. The recurrence formula (22) implies that each node can update their own weight W_{id} by exchanging the weight information for neighborhood nodes. In other word, W_{id} could be calculated by Bellman-Ford algorithm. As a result, we obtain another method to calculate MLB-routing as a distance-vector protocol.

F. Traffic-aware Case

We now show how MLB-routing could be used to construct traffic-aware routing algorithms. This expansion enables us to incorporate the traffic congestion in the model. In order to do this, we have to design the link costs to include a traffic component. In this paper, we use the outgoing queue sizes to a network link ij as a measure of traffic at each link.

Let Q_{ij} denote the queue length on the outgoing link e_{ij} and B_{ij} be the bandwidth associated with e_{ij} . Furthermore, we assume that d_{ij} is the standard delay in the link e_{ij} . Then, we redefine the cost of link e_{ij} as follows:

$$c_{ij} = d_{ij} + \frac{Q_{ij}}{B_{ij}}. \quad (23)$$

The above expression (23) represents the expectation time to path through the link e_{ij} under the condition of current queue length Q_{ij} . Hence, using (23) when we calculate the MLB-routing, packets are supposed to avoid the traffic congestion spontaneously (because the cost of congested link will take a great value).

G. Properties of MLB-routing

Finally, we discuss some properties of MLB-routing methodology. At first, we summarize the stability of MLB-routing algorithm. Earlier, we mentioned that the matrix \mathbf{A} should be satisfied the Haskins-Simon condition for equation (20). It means that W_{id} and $p(j|i)$ are both stable. Furthermore, Haskins-Simon condition is equivalent to

$$\rho(\mathbf{A}) = \max\{|\lambda|\} < 1, \quad (24)$$

where $\rho(\mathbf{A})$ is the spectral radius of the matrix \mathbf{A} and λ is the eigenvalue of \mathbf{A} . The value of $\max\{|\lambda|\}$ clearly decrease as γ become small. Therefore, setting the value of γ appropriately, we can construct the stable routing system.

The advantages of MLB-routing depend on the situation. Clearly, if the sending rates are low compared to the link capacities, then end-to-end delays do not happen even in single-shortest path algorithm. In such situation, there are no significant improve by using MLB-routing. Some packets are allocated to non-optimal route in MLB-routing, it results to the increase of end-to-end delays.

III. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the MLB-routing algorithm using simulations. We first describe the implementation of algorithm, and then explain the network topologies, followed by the experimental results.

A. Implementation

First, we describe our implementation of the MLB-routing algorithm. As discussed above, MLB-routing methodology is able to be implemented as both link-state routing protocol and the distance-vector routing protocol.

To implement the MLB-routing as link-state protocol, we need to determine both the route computation algorithm and the information to disseminate. The route computation algorithm described in Section II-D requires not only the link metric information but also the queue length information. Therefore, in addition to other link-state information, the queue length information should be added to the information dissemination process. Similarly, routers running the Bellman-Ford algorithm compute routes based on information obtained from neighborhood nodes. To calculate the route based on Section II-E, besides the routing table which is assumed in RIP, node weight W_{id} should be also exchanged (see Table I).

In this study, we implement the MLB-routing protocol as follows. First, we calculate the initial value of W_{id} by implementing the MLB-routing as link-state protocol. Then, update the each node of weight W_{id} iteratively by calculating the MLB-routing based on the Bellman-Ford algorithm.

B. Experimental Methodology

We ran simulation under the following methodology. To begin with, in order to evaluate the MLB-routing algorithm, we generated the network topologies using the Waxman model [13]. In this model, nodes are randomly placed on a 2-dimensional plane, and links are added incrementally. The values of the Waxman-parameters were set to $\alpha = 0.5$ and $\beta = 0.15$ respectively. We distributed 100 nodes in plane, and created the networks. The number of links is 392.

The link cost (delay) is set in proportion to the distance of two nodes, and the link bandwidths B is set to uniform value in all links; we calculated in three cases ($B = 40, 50, \text{ and } 60$ [packet/unit time]). In addition, top left 20 nodes is set as origin nodes, and bottom right 20 nodes as destination nodes. During 200 [unit time], every node-pair is sent 1 [packet/unit

time], which means the 400 packet generation at each unit time. The simulation is continued till all the data packets in transit reached their destinations.

To compare the performance of the MLB-routing algorithm to the single-shortest path algorithm (SPP), we set the maximum queue size at each network node to infinity. In other words, no packet is lost at any node. This assumption enables us to make meaningful comparisons between the delay values for viable source-destination pairs when the two different routing schemes are used.

C. Performance Analysis

Using the condition stated above, we ran simulation for both the MLB-routing algorithm and single-shortest path algorithm (SPP). In this paper, we compare two methodologies in terms of (i) end-to-end delay, and (ii) jitter times, respectively.

Now, let us briefly discuss the value of γ for MLB-routing. In the above example network, $\gamma > 0.875$ is required to satisfy $\rho(\mathbf{A}) < 1$. Hence, we calculated MLB-routing in three cases that $\gamma = 1.0, 2.0, \text{ and } 8.0$. The simulation result is shown in Figure 1 and Figure 2.

First, we compare end-to-end delay in each algorithm. Figure 1 indicates the scatter plot of mean transit times for each origin-destination pair. Each subfigure compares the result of MLB-routing and that of SPP, where the diagonal line indicates the break-even point. From the figures, we can confirm that a large majority of plots lie above the diagonal line for lots of simulation cases. It means that MLB-routing outperforms SPP with respect to transit times.

The efficiency of MLB-routing scheme depends on the situation. As previously mentioned, MLB-routing is effective when traffic is congested ($B = 40$). On the other hand, when the link bandwidths is high enough ($B = 60$), end-to-end delays do not occur in SPP and there are no improve by MLB-routing. The value of γ is also an important factor for MLB-routing. If γ is too high, MLB-routing becomes equal-cost multi path, and cannot achieve its full potential. In the case of $\{\gamma, B\} = \{8.0, 40\}$, some plots are scattered under the diagonal line, and it implies that some other links become bottlenecks. On the contrary, if γ is too low, MLB-routing allocates some packets to impractical path. Actually, MLB-routing is inferior to SPP in the case of $\{\gamma, B\} = \{1.0, 60\}$, it result from the MLB's ineffective detouring. We are supposed to determine the value of γ with respect to the congestion of network (the value of B). In this simulation, $\gamma = 2.0$ seems to enhance the potential of MLB-routing. Especially, the case of $\{\gamma, B\} = \{2.0, 40\}$ is best-performed, and 39.1% decrease of mean transit times are observed.

We now compare both protocol in terms of jitter times (standard deviation of transit times). Just as Figure 1, the scatter plot of jitter times for each origin-destination pair is shown in Figure 2. In the case of $\gamma = 1.0$ and 8.0, though significant improvements by our methodology are not observed, the scatter plot for $\gamma = 2.0$ (Figure 2-(b)) lies above the diagonal line. That is, MLB-routing has a possibility to improve not only transit times but also jitter times by setting

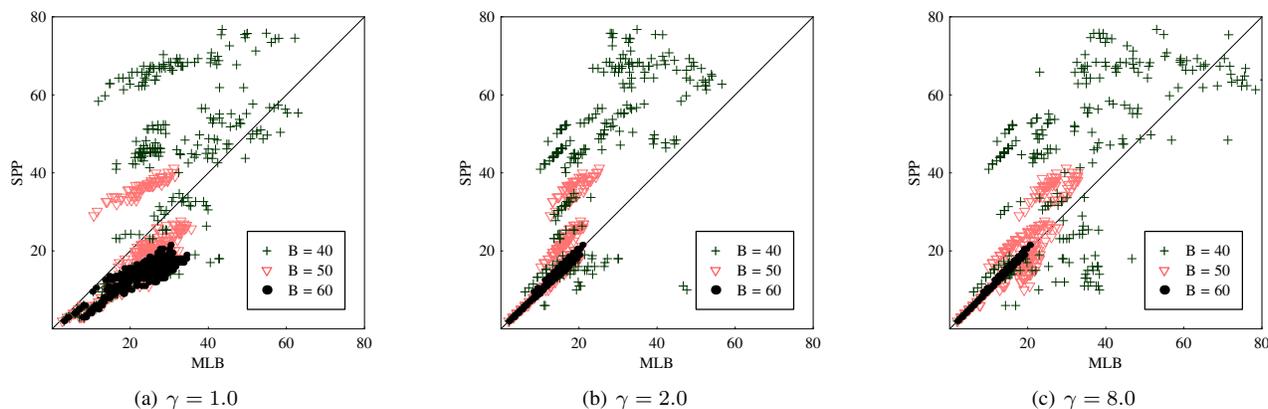


Fig. 1. Scatter Plot of Mean Transit Times

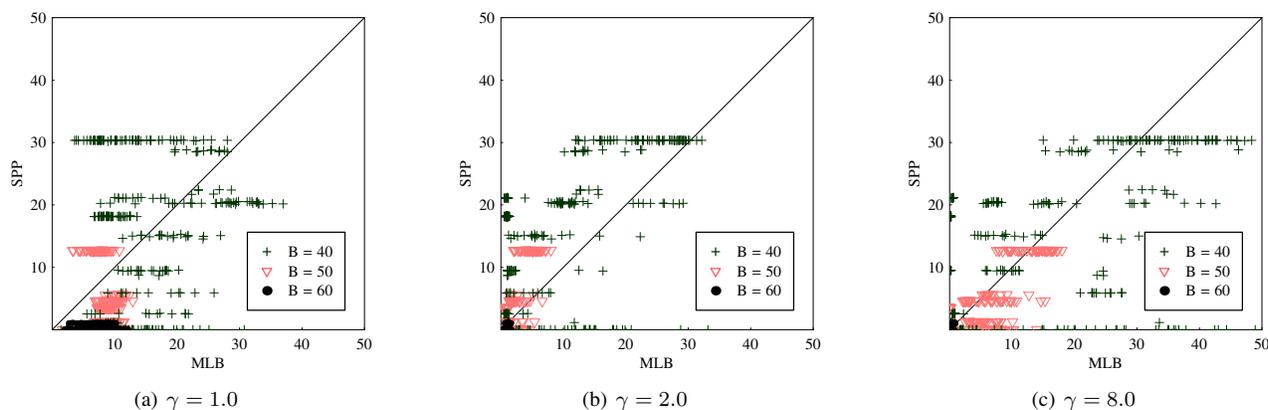


Fig. 2. Scatter Plot of Standard Deviation

parameters appropriately. Therefore, we can conclude that there is an appropriate value of γ which ensures maximum performance of MLB-routing.

IV. CONCLUSION AND FUTURE WORKS

In this paper, we have applied the idea of random utility theory to Internet routing to develop a multi-path routing methodology called MLB-routing. The key concept of the study is to incorporate the multiple paths from same origin to destination, and distribute the packets followed by the multinomial logit type probability. We have described that MLB-routing is implemented as not only link-state protocol but also distance-vector protocol. Furthermore, we have shown that our methodology produces more efficient use of network and causes significant improvements in end-to-end delays and jitter times. The performance of MLB-routing clearly depends on the value of parameter γ , so it would be the subject of future work how to determine the appropriate value.

ACKNOWLEDGMENT

This research was partially supported by the Grant-in-Aid for Scientific Research (B) No. 21300027 (2009-2011) from the Japan Society for the Promotion of Science.

REFERENCES

- [1] T. Akamatsu, Cyclic Flows, Markov Process and Transportation Stochastic Assignment, *Transportation Research*, 30B, pp.369–386, 1996.
- [2] R. Banner and A Orda, Multipath Routing Algorithms for Congestion Minimization, In *Proceedings of IFIP Networking*, 2005.
- [3] A. Basu, *et al.*, Routing Using Potentials: A Dynamic Traffic-Aware Routing Algorithm, In *Proceedings of SIGCOMM'03*, 2003.
- [4] I. Cidon, *et al.*, Analysis of Multi-Path Routing, *IEEE/ACM Trans. on Networking*, 7(6), pp.885–896, 1999.
- [5] S. De and S. K. Das, Dynamic Multipath Routing (DMPR): An Approach to Improve Resource Utilization in Networks for Real-Time Traffic, In *Proceedings of MASCOTS'01*, 2001.
- [6] C. Hendrick, Routing Information Protocol, RFC1058, 1988.
- [7] J. Moy, OSPF version 2, RFC2328, 1988.
- [8] P. Narvaez and K. Y. Siu, Efficient Algorithms for Multi-Path Link State Routing, In *Proceedings of ISCOM'99*, 1999.
- [9] M. Patriksson, *Traffic Assignment Problems: Models and Methods*, Intl Science, 1994.
- [10] J. Postel, Transmission Control Protocol, RFC793, 1981.
- [11] Y. Rekhter and T. Li, A Border Gateway Protocol version 4, RFC1771, 1995.
- [12] K. Train, *Discrete Choice Methods with Simulation*, Cambridge University Press, 2003.
- [13] B. Waxman, Routing of Multipoint Connections. *IEEE Journal on Selected Areas in Communications*, 6, pp.1617–1622, 1998.
- [14] S. Vutukury and J. J. Garcia-Luna-Aceves, MDVA: A Distance-Vector Multipath Routing Protocol, In *Proceedings of INFOCOM*, 2001.
- [15] W. T. Zaumen and J. J. Garcia-Luna-Aceves, Loop-free multipath routing using generalized diffusing computations, In *Proceedings of IEEE INFOCOM*, 1998.