IEICE TRANS. COMMUN., VOL.E94–B, NO.8 AUGUST 2011

PAPER

A New Multi-Path Routing Methodology Based on Logit-Type Probability Assignment

Yudai HONMA[†a], **Masaki AIDA**[††b], **Hideyuki SHIMONISHI**[†††c], ***and*** **Atsushi IWATA**[†††d], ***Members***

SUMMARY We present a new multi-path routing methodology, MLB-routing, that is based on the multinomial logit model, which is well known in the random utility field. The key concept of the study is to set multiple paths from the origin to the destination, and distribute packets in accordance with multinomial logit type probability. Since MLB-routing is pure multi-path routing, it reduces the convergence on some links and increases bandwidth utilization in the network. Unlike existing multi-path routing schemes, which pre-set alternate paths, the proposed method can dynamically distribute packets to every possible path and thus is more efficient. Furthermore, it should be mentioned that this methodology can be implemented as either a link-state protocol or a distance-vector protocol. Therefore, it well supports the existing Internet. Simulations show that this methodology raises network utilization and significantly reduces end-to-end delay and jitter.

key words: *multi-path routing, multinomial logit model, link-state protocol, distance-vector protocol*

1. Introduction

Most routing mechanisms in the Internet are based on single-shortest path routing and various metrics. This strategy seems rational because all packets on the networks should be delivered as soon as possible. Unfortunately, the shortest paths for many different node pairs tend to run through the same set of nodes. Hence, routing protocols based on the single-shortest path method tend to cause severe packet congestion.

The legacy approach to reducing traffic congestion is end-to-end control. The most famous example is the TCP mechanism [16], used by the Internet. This approach controls the “quantity” of packets on the network. A more recent approach to addressing traffic congestion is to control packet “routing”.

The PB-routing protocol proposed by Basu et al. identifies traffic potential fields on the network, and tries to avoid the congested links by setting detour routes [4]. However, even if they consider link traffic (i.e. adopting the traffic-aware routing), the basic concept of “single-shortest path” routing is not changed at all. That is, packets of same node pairs are allocated to same route, and cannot be transferred beyond the link-bandwidths. The virtual-circuit, one of the essential techniques for traffic engineering and QoS, also aims to control packet routes [2]. However, its goal is to establish alternate paths between node pairs through flow-base control which means that scalability is poor. From a practical point of view, the solutions that are scalable are of importance.

The multi-path routing approach has been receiving a lot of attention as another methodology to ease traffic congestion. The basic idea of multi-path routing is to spread the packets across multiple alternate paths. Hence, multi-path routing is supposed to encourage the effective utilization of network resources. First, the number of packets per path is decreased, so the emergence of bottlenecks is suppressed. In addition, multi-path routing are also expected the use of more links and better load-balancing. Consequently, multi-path routing would enhance various attributes such as the quality of service (QoS), delay, and delivery reliability.

It is well known that some protocols such as OSPF and BGP implement equal-cost multi-paths (ECMP) [13], [17]. However, it is unlikely that multiple paths with equal distance exist between each origin-destination pair, which means the full connectivity of the network is not used for load-balancing. Therefore, other multi-path routing methodologies, which also use “near-optimal paths”, have been discussed in a number of papers (see e.g. [3], [5], [14] and the references therein). There are two main types of algorithms for IP routing: distance-vector routing and link-state routing. Nevertheless, most multi-path routing protocols adopt the link-state approach which tracks the status from every router on the network. Unfortunately, few studies have examined distance-vector protocols [23]–[25].

In this paper, we present a new multi-path routing methodology based on the *m*ulti*n*omial *l*ogit model (MNL). We call it “(*m*ultinomial) *l*ogit *b*ased routing”, or “MLB-routing”. Since MLB-routing is pure multi-path routing, it reduces the severe convergence onto a few links and increases the bandwidth utilization in the network. It also should be mentioned that this methodology can be implemented as either a link-state protocol or a distance-vector protocol. Therefore, it works well with the current Internet mechanism.

The key idea of MLB-routing is to base packet routing on MNL. MNL is a classical model of *random utility theory*, and is used to assess an individual’s selection behavior [20]. Since the action of “selecting something” is triggered

Manuscript received April 7, 2010.
Manuscript revised February 24, 2011.
[†]The author is with Waseda University, Tokyo, 169-8555 Japan.
[††]The author is with Tokyo Metropolitan University, Hachioji-shi, 191-0065 Japan.
[†††]The authors are with NEC, Kawasaki-shi, 211-8666 Japan.
a) E-mail: yudai@aoni.waseda.jp
b) E-mail: maida@sd.tmu.ac.jp
c) E-mail: h-shimonishi@cd.jp.nec.com
d) E-mail: a-iwata@ah.jp.nec.com
DOI: 10.1587/transcom.E94.B.2282

by various psychological factors, it is difficult to predict exactly what will be selected. Hence, random utility theory estimates the "probability" of each alternative being selected; this is the most interesting point of the theory.

It must be emphasized that MNL has "a set of alternatives (candidate)", and selected-probability can be calculated as the pro rate allocation by "the exponential value of alternative's utility". Generalizing this idea, we construct the MLR-routing system to incorporate "all possible paths", and the pro rate allocation by "the exponential value of route-cost". These two strategy provides many advantages to MLR-routing over traditional multi-path routing techniques.

There are enormous candidate paths between a origin-destination pair. In lots of existing multi-path routing protocol, nevertheless, some of paths are only considered because of limitations of their algorithms. For example, MDVA, which is one of the multi-path routing implemented as distance-vector protocols, introduced directed acyclic graph, and some of candidate paths result to be neglected [23], [24]. Meanwhile, it is noteworthy that MLB-routing aims to consider "all of paths" as candidate routes. Thus, it contributes to more effective utilization of network resources.

It seems impractical to consider all possible paths in large scale networks. However, the above two features that we incorporate "all possible paths" and "the probability follows the exponential values of route-costs" paradoxically decreases the computational complexity. This discussion results from the condition of equivalency between MNL and Markov model in network flow [1]. In addition, they enable us to implement MLB-routing as distance-vector protocol. 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 explain the formulation of MNL and describe the MLB-routing model in greater detail. Then, we extend MLB-routing to hop-by-hop routing for the Internet network, and clarify the system structure analytically (Sect. 3). This is followed by Sect. 4 which describes our implementation of MLB-routing and evaluates its performance. Finally, we conclude and mention future works in Sect. 5.

This paper is an extensional article of our previous research [11]. In the previous proceeding, we discuss the basic concept of MLB-routing, so there are not so much differences. In this regard, however this paper also extends the discussion about the frequency of looped-route emergence, performance analysis in terms of network utilization, and appropriate parameter setting.

2. Concept of MLB-Routing

In this section, we present the concept of MLB-routing methodology. We start with a brief summary of the multinomial logit model (MNL). We then describe how to apply MNL to a routing system, and formulate that MLB-routing system. (An example of a simple MLB-routing system is given in the Appendix. It will help to understand the idea of this section.)

2.1 Multinomial Logit Model

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

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

and the individual is assumed to choose the alternative that maximizes his utility. Our first objective is to analyze which alternative the individual will choose.

If we "precisely" know his utility U_j, we can figure out which alternative the individual will choose. However, it is impossible to precisely determine the individual's utility, because utility U_j is influenced by various factors such as his character, preference, and so on. Therefore, it is inevitable that "the utility estimated by researchers" will contain some error.

In view of these facts, let us assume that the individual receives utility U_j upon choosing alternative j as follows:

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

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

The definition stated above indicates that utility U_j contains random variables, so it is impossible to identify just the alternative the individual would choose. This fact implies that we should analyze choice behavior as a stochastic event. Here, choosing alternative j means that U_j has the highest value. Therefore, we can formulate probability P_j that the individual chooses alternative j as follows:

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

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

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

$$F(\epsilon) = \exp[-\exp[-\gamma(\epsilon - \alpha)]], \quad (4)$$
$$f(\epsilon) = \gamma \exp[-\gamma(\epsilon - \alpha)] \exp[-\exp[-\gamma(\epsilon - \alpha)]]. \quad (5)$$

Here $F(\epsilon)$ is CDF and $f(\epsilon)$ is PDF. In this case, we can obtain probability P_j as follows:

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

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

2.2 MLB-Routing Assignment

As described above, MNL is not only simple but also quite persuasive, so it is used in various fields. One typical example is routing behavior in a transportation network model. The general objective of transportation network models is to describe transportation flows in road networks, and so it is essential to be able to predict driver's routing behavior [15]. MNL can be utilized to describe such routing behavior, and a number of studies have discussed effective calculation techniques. We considered their work in proposing MLB-routing.

In order to describe the MLB-routing protocol, we first define the system model considered in this study. We represent 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 N, and the set of edges by E. Furthermore, e_{ij} is 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 a sink. Finally, every node v has a set of $Z(v)$ neighbors denoted by $nbr(v)$.

Now we explain the concept of the MLB-routing protocol. Let us consider any packet p at origin node o going to destination node d. There are many paths, one of which is the shortest path, that reach destination d, and we try to use "all" paths (theoretically). This, however, doesn't mean that all paths are used equally; naturally the shortest path should be used most often, and path cost increases (its length), the less the path should be used. To incorporate this idea, let us define the probability that packet p uses the rth path P_r^{od} as follows:

$$P_r^{od} = \frac{\exp\left[-\gamma C_r^{od}\right]}{\sum_{r\in\Phi^{od}} \exp\left[-\gamma C_r^{od}\right]}. \tag{7}$$

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

$$C_r^{od} \overset{\text{def.}}{=} c_{ov_1} + \sum_{l=1}^{\Lambda-1} c_{v_l v_{l+1}} + c_{v_\Lambda d} \tag{8}$$

where the rth path is $o \to v_1 \to v_2 \to \cdots \to v_\Lambda \to d$.

Now let us discuss how to define the alternative set Φ^{od}. In the simple definition, the alternative set consists of "simple paths"; those that contain no loops. In this study, however, we eliminate this restriction, and consider all paths including "looped paths". This strategy has the possibility of generating very long detours which should not be assigned. Furthermore, it seems to be difficult to calculate the probability for large scale networks.

On the contrary, this definition is made to enhance computational feasibility. As shown in later sections, this definition enables us to perform stochastic assignment in large scale networks. Moreover, the definition is useful in some iterative algorithms for solving the assignment which means the moving to a distance-vector protocol.

2.3 Why Adopt MNL?

Here we address the reason for adopting MNL for assignment. First, we state that MLB-routing is an "optimal" system, because MNL is formulated to "maximize the utility" (see (3)). To state this more precisely, MLB-routing is an optimal strategy under the assumption that the cost of each candidate path is fluctuates and in doing so follows a Gumbel distribution.

There are various factors that should be considered in determining link costs (such as delay, bandwidth, and the number of flows). However, it is difficult to incorporate all factors and some might change frequently. Hence, we cannot determine the costs precisely, and some error must be included. It provides a plausible explanation that the estimated link cost randomly fluctuates, which is assumed in MLB-routing.

Furthermore, owing to the property of MNL, in MLB-routing, the choice probability of a path decreases exponentially as its length increases. This means that high utility paths are frequently selected and low utility ones are not. This structure inherently supports multi-path routing. Consequently, we consider that MLB-routing is favorable as a multipath routing system.

3. The Structure of MLB-Routing Model

In this section, we extend MLB-routing to cover hop-by-hop routing systems. In particular, we discuss the structure of MLB-routing systems and efficient algorithms in terms of calculation costs.

3.1 Extension to Hop-by-Hop Routing

First, we explain how to allocate packets to various paths. Obviously we can allocate packets to any arbitrary path by source routing. However, to apply MLB-routing to the present Internet network, it is better to implement MLB-routing as hop-by-hop routing.

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

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

$$W_{id} \overset{\text{def.}}{=} \sum_{r\in\Phi^{id}} \exp\left[-\gamma C_r^{id}\right] + \delta_{id}, \tag{10}$$

where $p(j|i)$ is the probability that a packet in node i will select the next hop j, and δ_{jd} is Kronecker delta. This formulation satisfies the condition

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

which is should be regarded as the conditional probability. Furthermore, the equivalency between two assignments is easily confirmed by

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

The existence of Markov assignment equivalency implies that we can apply the same simple rule to all nodes (routers). More clearly, if each node uses (9) to choose the next node to pass the packet to, the resulting network flow equals the assignment discussed in Sect. 2.2.

Since the above method distributes packets to every possible next node in a packet-by-packet manner, out-of-order packet delivery is possible. This paper does not address this problem, but solutions include 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, the transfer of all packets from arbitrary nodes spontaneously hold to (7). In general, existing routing tables also do not depend on the origin, so it is quite favorable feature of the MLB-routing. An example of the routing table assumed in MLB-routing is shown in Table 1. Since multi-path routing is used, there are multiple candidate nodes for the next hop even if packets are intended for the same destination.

3.2 How to Calculate W_{id}

To determine the value of $p(j|i)$, we have to calculate W_{id} which is included in $p(j|i)$. Since the summation in W_{id} means considering all paths from node i to d, the native approach is not feasible.

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

Table 1 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	$p_C(j_N|i)$	

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

The element of $\boldsymbol{A}^2$ yields

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

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

$$a_{ij}^{[L]} = \sum_{r \in \Phi_L} \exp\left[-\gamma C_r^{ij}\right]. \tag{15}$$

From the definition, we have

$$\Phi^{ij} = \Phi_1^{ij} \oplus \Phi_2^{ij} \oplus \cdots. \tag{16}$$

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

$$\boldsymbol{W} = \boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + \boldsymbol{A}^3 + \cdots \tag{17}$$

where the element of matrix $\boldsymbol{W}$ is

$$W_{ij} = \sum_{r \in \Phi^{ij}} \exp\left[-\gamma C_r^{ij}\right] + \delta_{id} \tag{18}$$

Furthermore, if matrix $\boldsymbol{A}$ satisfies the Hawkins-Simon condition [10], [18], the following formula is obtained:

$$[\boldsymbol{I} - \boldsymbol{A}]^{-1} = \boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + \boldsymbol{A}^3 + \cdots. \tag{19}$$

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

$$\boldsymbol{W} = [\boldsymbol{I} - \boldsymbol{A}]^{-1}. \tag{20}$$

As indicated above, determining the link-weighted matrix $\boldsymbol{A}$ yields transition probability $p(j|i)$ in a straightforward manner. This means that all routers (nodes) will implement MLB-routing. Since the link weighted matrix $\boldsymbol{A}$ is derived by assuming a link-state routing protocol (such as OSPF [13]), we implement MLB-routing as a link-state routing protocol.

3.3 Distance-Vector Routing Protocol

Although we describe MLB-routing as a link-state protocol above, RIP [9], a distance-vector routing protocol based on the Bellman-Ford algorithm, remains popular. Hence, it is worthwhile implementing MLB-routing as a distance-vector protocol. We explain how to do this below.

The idea of distance-vector routing protocols is to exchange information (such as routing tables) between neighboring routers. Thus, we calculate W_{id} (i.e. $p(j|i)$) by the iterative exchange of information. It is easy to understand that

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{I} + \boldsymbol{A}\left(\boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2 + \boldsymbol{A}^3 + \cdots\right) \\ &= \boldsymbol{I} + \boldsymbol{A}\boldsymbol{W} \end{aligned} \tag{21}$$

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

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

The recurrence formula (22) implies that each node can update its own weight W_{id} by the exchange of weight information between neighboring nodes. In other words, W_{id} can be calculated by the Bellman-Ford algorithm. As a result, MLB-routing can be implemented as a distance-vector protocol.

3.4 Traffic-Aware Case

We now show how MLB-routing can be used to construct traffic-aware routing algorithms. This expansion enables us to consider traffic congestion. In order to do this, we have to alter the link costs so that they include a traffic component. In this paper, we use the outgoing queue size to network link, ij, as a measure of traffic at each link.

Let Q_{ij} denote the queue length for 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 link e_{ij}. We redefine the cost of link e_{ij} as follows:

$$c_{ij} = d_{ij} + \frac{Q_{ij}}{B_{ij}}. \tag{23}$$

(23) represents the expectation time to path through the link e_{ij} under the condition of current queue length Q_{ij}. Hence, by using (23) in MLB-routing, packets will avoid traffic congestion spontaneously because congested links have high cost.

3.5 Properties of MLB-Routing

Finally, we discuss some properties of the MLB-routing approach. At first, we summarize its stability. Earlier, we mentioned that matrix $\boldsymbol{A}$ should satisfy the Hawkins-Simon condition for Eq. (20). This means that both W_{id} and $p(j|i)$ are stable. Furthermore, the Hawkins-Simon condition is equivalent to

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

where $\rho(\boldsymbol{A})$ is the spectral radius of matrix $\boldsymbol{A}$ and λ is the eigenvalue of $\boldsymbol{A}$. The value of $\max\{|\lambda|\}$ clearly decreases as γ becomes small. Therefore, setting the value of γ appropriately yields a 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 are non-existent even with the single-shortest path algorithm. In this situation, MLB-routing is less than optimal since some packets are allocated to non-optimal routes, which increases the average end-to-end delays.

Also, it should be mentioned how to set the value of γ which characterize the performance of MLB-routing. As discussed in Sect. 2.1, when $\gamma = 0$, all alternative paths are equally selected, and if $\gamma = \infty$, only the best path is selected. Normally, γ should be set between $0 < \gamma < \infty$. Hence, users of MLB-routing need to determine the value of γ depending on their purposes.

One of the techniques we recommend in the paper is to focus on the "Independence from Irrelevant Alternatives (I.I.A.)" property of MLB-routing (originally for MNL) [20]. Owing to I.I.A. property, the ratio of choice probability between alternative path r_1 and path r_2 is not affected other alternative path set, and is determined only by their path costs. Mathematically discussing, from (7), the ratio of $P^{od}_{r_1}$ and $P^{od}_{r_2}$ is calculated as

$$\begin{aligned} \frac{P^{od}_{r_1}}{P^{od}_{r_2}} &= \frac{\dfrac{\exp\left[-\gamma C^{od}_{r_1}\right]}{\sum_{r\in\Phi^{od}} \exp\left[-\gamma C^{od}_{r}\right]}}{\dfrac{\exp\left[-\gamma C^{od}_{r_2}\right]}{\sum_{r\in\Phi^{od}} \exp\left[-\gamma C^{od}_{r}\right]}} \\ &= \frac{\exp\left[-\gamma C^{od}_{r_1}\right]}{\exp\left[-\gamma C^{od}_{r_2}\right]} = \exp\left[-\gamma\left(C^{od}_{r_1} - C^{od}_{r_2}\right)\right]. \end{aligned} \tag{25}$$

From (25), note that it does not include alternative set Φ^{od}, and the ratio of probability between two paths only depends on "the difference of their path costs". For example, if choice probability would be decreased to a half by the increase of unit cost, γ should be set 0.693 ($\because \exp[-0.693] \fallingdotseq 0.5$). This ratio is also kept among all alternative path sets. That is to say, if there are three alternative paths, and $\gamma = 0.693, C^{od}_{r_1} = 5, C^{od}_{r_2} = 6, C^{od}_{r_3} = 7$, then choice probabilities become $P^{od}_{r_1} = 4/7, P^{od}_{r_2} = 2/7, P^{od}_{r_3} = 1/7$. Therefore using this property, we can control the performance of MLB-routing as we want.

4. Performance Evaluation

This section uses simulations to evaluate the performance of the MLB-routing algorithm. We first describe the implementation of the algorithm, explain the network topologies, and the results.

4.1 Implementation

First, we describe our implementation of the MLB-routing algorithm. As discussed above, MLB-routing can be implemented as either a link-state routing protocol or a distance-vector routing protocol.

We consider that specific implementation will be established by extending OSPF and RIP. General concepts are as follows. First, to implement MLB-routing as a link-state protocol, basic behavior would be same as OSPF (such as how to advertisement or flooding). However, we need to extend OSPF in terms of both the route computation algorithm and the information that must be disseminated. The route computation algorithm which MLB-routing uses is described in Sect. 3.2. MLB-routing as a link-state protocol has to use the above algorithm. Furthermore, MLB-routing requires not only link metric information but also queue length information. Hence, when flooding, length information also should be advertised.

To implement MLB-routing as a distance-vector protocol, basic behavior should be same as RIP (e.g. triggered updates). What we need to extend is the route update algorithm and the kept routing table. In a distance-vector protocol, routers running the Bellman-Ford algorithm compute routes based on information obtained from neighboring nodes. Thus, instead of updating hop counts, MLB-routing as a distance-vector protocol has to exchange and update each node weight W_{id}. This idea is summarized in Sect. 3.3. In addition, the routing table used in MLB-routing is described like Table 1, so MLB-routing as a distance-vector protocol should keep not only next routers but also their node weight W_{id}.

In this study, we implement the MLB-routing as distance vector protocol to reflect promptly the change of network state. However, it is known that distance vector convergence occurs rather slow. Thus, to reduce the convergence time, we first calculate the initial value of W_{id} from (20) (like a link-state protocol). Then, we update node weight W_{id} iteratively by running the Bellman-Ford algorithm.

4.2 Simulation Methodology

We ran simulations under the following methodology. To begin with, in order to evaluate the MLB-routing algorithm, we generated network topologies using the Waxman model [21]. Waxman model is one of random graphs to create the network topology for the purpose of evaluating routing algorithms, and hence is employed in many studies of routing performance [7], [8], [19], [22]. The suitability for this purpose has been recognized also in lots of context about Internet topology modeling. Especially, since nodes are randomly placed on a 2-dimensional plane link connection probability decreases exponentially as their distance increases, their link formation among routers appears to describe well the real world [12]. We distributed 100 nodes on the plane, and created the networks; the number of links was 392. The values of the Waxman-parameters were set to $\alpha = 0.5$ and $\beta = 0.15$. The network we used for evaluation is shown in Fig. 1. In the above example network, $\gamma > 0.875$ is needed to satisfy $\rho(\boldsymbol{A}) < 1$.

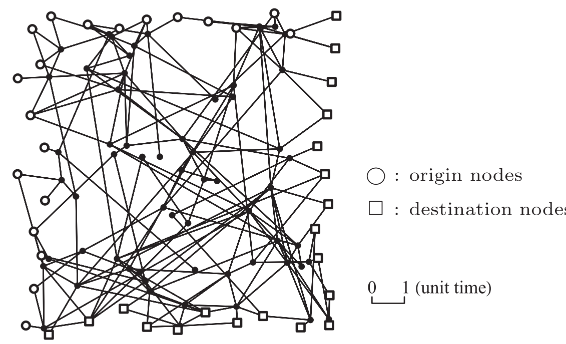


Fig. 1 Simulation network.

The link delay cost should be set uniformly if we set great store on the actual Internet, it means that the protocol employs the hop count as a routing metric like RIP. However in this study, we set the standard link delay cost in proportional to the separation distance of two nodes. This is mainly because one of the most important features in MLB-routing is that the choice probability of a path truly reflects each path cost by its exponential value. Therefore, it is more favorable for MLB-routing that each path cost clearly characterize their path length. We consider that a routing metric should be set flexibly in MLB-routing, though our routing enough works when the hop count is employed as a routing metric.

In the simulation, 20 of the top left nodes were set as origin nodes, and 20 of the bottom right nodes were taken to be destination nodes. Within the unit time period of 200, every node-pair sent/received 1 packet/unit time, which means that 400 packets were generated in each unit time. The simulation was continued until all data packets had reached their destinations. We set the link bandwidth B to a uniform value in all links, and examined various cases ($B = 5, 10, 15, \cdots 95, 100$ [packets/unit time]). In the following, we describe the results of the situation that the network is very clouded($B = 40$), the network is a little clouded($B = 50$), and the network is not clouded($B = 60$). When B is much less than 40, the network is too clouded even if we adopt MLB-routing, so we cannot evaluate our methodology appropriately. By contrast, when B is greater than 60, the network is not clouded at all, so there are no need to adopt multi-path routing.

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.

4.3 Frequency of Looped-Route Emergence

Using the condition stated above, we ran simulations of the MLB-routing algorithm. First, we discuss the frequency with which looped-routes emerged.

As previously mentioned, MLB-routing allocates some packets to looped-routes. Hence, we evaluated the amount of packets distributed to looped-routes by simulations. The number of such packets clearly depends on the value of γ. If γ is low, all candidate paths are used almost equally, so the ratio of looped-routes is relatively high. On the other hand, if γ is high, short paths are preferred, and the ratio becomes low.

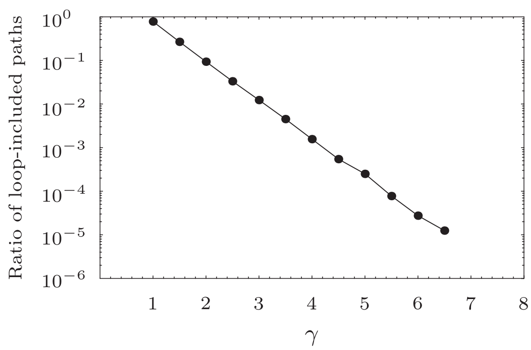


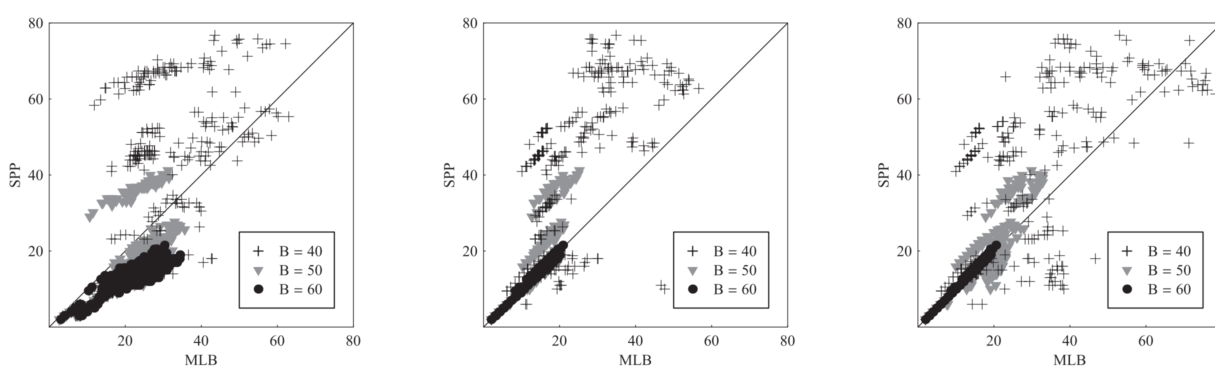
Fig. 2 Ratio of loop-included paths.

By verifying γ from 1.0 to 8.0 in steps of 0.5, we simulated the MLB-routing system and calculated the ratio of looped-routes. For simplicity, we assumed $B = \infty$ in this simulation. The result is plotted in Fig. 2. From Fig. 2, we can confirm that the ratio decreases exponentially against γ. Furthermore, in this simulation, looped-routes did not exist when $\gamma \geq 6.5$.

4.4 End-to-End Delay

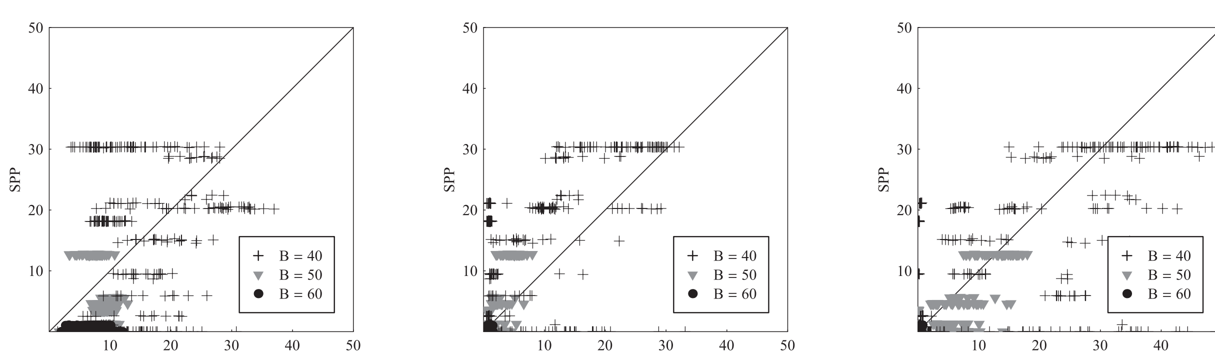
Next, we compare the MLB-routing algorithm to the single-shortest path algorithm (SPP) in terms of (i) end-to-end delay, and (ii) jitter times. Though we verified γ from 1.0 to 8.0 in steps of 0.5, here we show the result of MLB-routing for the three cases of $\gamma = 1.0$, 2.0, and 8.0. The simulation results are shown in Fig. 3 and Fig. 4.

First, we compare the end-to-end delay of each algorithm. Figure 3 provides a scatter plot of mean transit times



(a) $\gamma = 1.0$ (b) $\gamma = 2.0$ (c) $\gamma = 8.0$

Fig. 3 Scatter plot of mean transit times.



(a) $\gamma = 1.0$ (b) $\gamma = 2.0$ (c) $\gamma = 8.0$

Fig. 4 Scatter plot of standard deviation.

for each origin-destination pair. Each subfigure compares the result of MLB-routing to 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 which indicates that MLB-routing outperforms SPP with respect to transit times.

The efficiency of MLB-routing depends on the situation. As previously mentioned, MLB-routing is most 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 MLB-routing offers no improvement. The value of γ is also an important factor determining MLB-routing performance. If γ is too high, MLB-routing becomes equal-cost multi path routing, and cannot achieve its full potential. In the case of $\{\gamma, B\} = \{8.0, 40\}$, some plots lie under the diagonal line, which implies that some links have become bottlenecks. On the contrary, if γ is too low, MLB-routing allocates some packets to impractical paths. Actually, MLB-routing is inferior to SPP in the case of $\{\gamma, B\} = \{1.0, 60\}$; this is due to MLB's ineffective approach to detouring. This indicates the need to determine the value of γ appropriately. In this simulation, $\gamma = 2.0$ seems to enhance the performance of MLB-routing. Especially in the case of $\{\gamma, B\} = \{2.0, 40\}$ offers the best-performance; a 39.1% decrease in mean transit time is observed. In addition, MLB-routing are not inferior to SPP even when $B = 50, 60$, which means that MLB-routing with $\gamma = 2.0$ works well whether traffic is congested or not.

We now compare the two protocols in terms of jitter (standard deviation of transit times). The scatter plot of jitter times for each origin-destination pair is shown in Fig. 4. For γ values of 1.0 and 8.0, MLB-routing shows no significant advantage. For $\gamma = 2.0$ (Fig. 4(b)), however, MLB-routing can offer improved jitter by setting parameters appropriately.

4.5 Network Utilization

We now compare the MLB-routing algorithm to SPP in terms of network utilization. Same as previous subsection, though we verified γ from 1.0 to 8.0 in steps of 0.5, here we show the result of MLB-routing for the three cases of $\gamma = 1.0$, 2.0, and 8.0.

Several indices about network utilization is summarized in Table 2 ~ Table 4. In particular, we focused on all network links and measured how much of bands they are used during simulations. As first index for network utilization, we calculated all link's average of bandwidth use ratio. Furthermore, we also counted the number of links which is not used at all, and the number of links which are fully used (namely bottlenecks).

From Table 2 ~ Table 4, we can confirm that average use ratio of MLB $\gamma = 2.0$ is greater than that of SPP. It means that network resources are more utilized by multi-path routing mechanism. Additionally, both the number of "not used" links and the number of "bottlenecks" are decreased by MLB-routing of $\gamma = 2.0$. Especially, suppression of bottlenecks is favorable to ease traffic congestion.

Table 2 Link status of network utilization ($B = 40$).

	SPP	$\gamma = 1.0$	$\gamma = 2.0$	$\gamma = 8.0$
Average	13.2%	28.2%	16.2%	15.0%
Not used	246	104	187	212
Fully used	8	11	1	0

Table 3 Link status of network utilization ($B = 50$).

	SPP	$\gamma = 1.0$	$\gamma = 2.0$	$\gamma = 8.0$
Average	11.6%	23.4%	13.1%	12.3%
Not used	246	126	203	221
Fully used	7	11	0	0

Table 4 Link status of network utilization ($B = 60$).

	SPP	$\gamma = 1.0$	$\gamma = 2.0$	$\gamma = 8.0$
Average	10.0%	20.0%	10.5%	9.7%
Not used	256	135	223	242
Fully used	1	8	1	1

We consider that above facts describe the well performance of MLB-routing.

Though MLB-routing of $\gamma = 1.0$ enhances the average use ratio, it also results to the increase of bottlenecks. These are mainly because of ineffective detouring. Meanwhile, MLB-routing of $\gamma = 8.0$ indicates the similar features with $\gamma = 2.0$. However, indices of $\gamma = 8.0$ is inferior to that of $\gamma = 2.0$, which means that multi-path routing mechanism is not enough worked in $\gamma = 8.0$.

4.6 Appropriate Parameter Setting

From above discussion, we can now elaborate about the parameter setting for MLB-routing. As described, there would be appropriate parameter setting which enhance the performance of MLB-routing. In this simulation, $\gamma = 2.0$ performs better than SPP. Here, what we want to emphasize is that this parameter setting is favorable whether traffic is congested or not. In other word, parameter of MLB-routing is enough tolerant to traffic congestion and link bandwidth.

We should also state the fact that similar results are obtained when $\gamma = 2.5 \sim 4.0$. these results indicate that MLB-routing has a wide-range of appropriate parameter. Hence, even if the parameter is a little overshot, MLB-routing would still works with enough performance. That is to say, we do not have to be so nervous for parameter setting.

Finally, we discuss the parameter setting for different networks. It is certain that "the optimal" parameter depends on topology or size of networks. However, we consider that "relatively appropriate" parameter for different networks is similar with the above simulation result. This is because (i) MLB-routing parameter is tolerant to traffic congestion, (ii) appropriate parameter has a wide-range, and (iii) MLB-routing has I.I.A. property which means that basic performance is independent from alternative path set (= networks). From (25), $\gamma = 2.0$ is regarded that the choice probability becomes a half if path cost increase 0.347 ($\because\ \exp[-2.0 \times 0.347] \fallingdotseq 0.5$). Now, average link delay cost in our simulation is 2.78, so 0.347 correspond to 0.347/2.78=0.125 hops. Hence, if we change the metric of MLB-routing to the hop count, then appropriate parameter is transformed to $\gamma = 2.0 \times 2.78 = 5.56$ ($\because\ \exp[-5.56 \times 0.125] \fallingdotseq 0.5$). In this manner, we can discuss the appropriate parameter of MLB-routing in various networks and metrics.

5. 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. Its key concept is to use multiple paths from the origin to the destination, and distribute the packets according to multinomial logit type probability. We showed that MLB-routing can implemented as a link-state protocol as well as a distance-vector protocol. Furthermore, simulations showed that MLB-routing produces more efficient use of network resources and significantly reduces end-to-end delay and jitter. We analyzed the appropriate parameter setting in MLB-routing, however "best" value clearly depends on applied networks. In future works, we prefer to discuss the optimum parameter value for MLB-routing. In addition, avoiding the generation of looped-routes is also an important task in extending our model.

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, vol.30B, pp.369–386, 1996.
[2] D. Awduche, et al., "Overview and principles of Internet traffic engineering," RFC3772, 2002.
[3] R. Banner and A Orda, "Multipath routing algorithms for congestion minimization," Proc. IFIP Networking, 2005.
[4] A. Basu, et al., "Routing using potentials: A dynamic traffic-aware routing algorithm," Proc. SIGCOMM'03, 2003.
[5] I. Cidon, et al., "Analysis of multi-path routing," IEEE/ACM Trans. Netw., vol.7, no.6, pp.885–896, 1999.
[6] S. De and S.K. Das, "Dynamic multipath routing (DMPR): An approach to improve resource utilization in networks for real-time traffic," Proc. MASCOTS'01, 2001.
[7] H. De Neve and P. Van Mieghem, "TAMCRA: A tunable accuracy multiple constraints routing algorithm," Comput. Commun., vol.23, pp.667–679, 2000.
[8] L. Guo and I. Matta, "Search space reduction in QoS routing," Comput. Netw., vol.41, pp.73–88, 2003.
[9] C. Hendrick, "Routing information protocol," RFC1058, 1988.
[10] D. Hawkins and H.A. Simon, "Note: Some conditions of macroeconomic stability," Econometrica, 17, pp.245–248, 1949.
[11] Y. Honma, M. Aida, H. Shimonishi, and A. Iwata, "A new multi-path routing methodology based on logit type assignment," Proc. 2nd International Workshop on the Network of the Future (FutureNet II), 2009.
[12] A. Lakhina, et al., "On the geographic location of Internet resources," IEEE J. Sel. Areas Commun., vol.21, no.6, pp.934–948, 2003.
[13] J. Moy, "OSPF version 2," RFC2328, 1988.
[14] P. Narvaez and K.Y. Siu, "Efficient algorithms for multi-path link state routing," Proc. ISCOM'99, 1999.
[15] M. Patriksson, Traffic Assignment Problems: Models and Methods, Intl Science, 1994.
[16] J. Postel, "Transmission control protocol," RFC793, 1981.
[17] Y. Rekhter and T. Li, "A border gateway protocol version 4," RFC1771, 1995.
[18] E. Seneta, Non-negative Matrices And Markov Chains, Springer, 2006.
[19] A. Shaikh, et al., "Evaluating the impact of stale link state on quality-of-service routing," IEEE/ACM Trans. Netw., vol.9, pp.162–176, 2001.
[20] K. Train, Discrete Choice Methods with Simulation, Cambridge University Press, 2003.
[21] B. Waxman, "Routing of multipoint connections," IEEE J. Sel. Areas Commun., vol.6, no.9, pp.1617–1622, 1988.
[22] S. Verma, et al., "QoS based multicast routing algorithms for real time applications," Performance Evaluation, vol.34, pp.273–294, 1998.
[23] S. Vutukury and J.J. Garcia-Luna-Aceves, "MDVA: A distance-vector multipath routing protocol," Proc. INFOCOM, 2001.
[24] S. Vutukury, Multipath Routing Mechanisms for Traffic Engineering and Quality of Service in the Internet, Doctoral thesis of University of California Santa Cruz, 2001.
[25] W.T. Zaumen and J.J. Garcia-Luna-Aceves, "Loop-free multipath routing using generalized diffusing computations," Proc. IEEE INFOCOM, 1998.

Appendix: An Example of MLB-Routing

To promote understanding, we provide a brief example of MLB-routing.

We consider the computer network consisting of 4 nodes and 5 links described in Fig. A·1. For simplicity, all links are assumed to have unit cost (i.e. $c_{**} = 1$). The goal is to send packets from node 1 to node 4 by MLB-routing.

As is easily confirmed, there are three candidate paths from node 1 to node 4:

path I $1 \to 2 \to 4$,
path II $1 \to 3 \to 4$,
path III $1 \to 2 \to 3 \to 4$.

Total costs of path I and path II are both $C_{\mathrm{I}} = C_{\mathrm{II}} = 2$, and

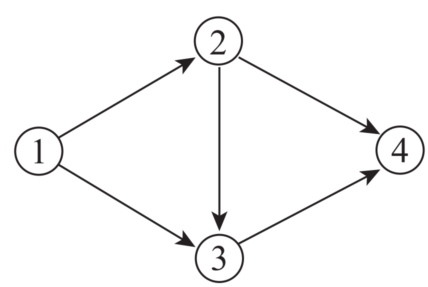


Fig. A·1 Sample network.

that of path III is $C_{\mathrm{III}} = 3$. Thus, from (7), the probability with which each path is used is derived as follows:

$$\textbf{path I} \quad \frac{\exp[-\gamma C_{\mathrm{I}}]}{\sum_{r=\mathrm{I}}^{\mathrm{III}} \exp[-\gamma C_r]} = \frac{x^2}{x^3+2x^2} \tag{A$\cdot$1}$$

$$\textbf{path II} \quad \frac{\exp[-\gamma C_{\mathrm{II}}]}{\sum_{r=\mathrm{I}}^{\mathrm{III}} \exp[-\gamma C_r]} = \frac{x^2}{x^3+2x^2} \tag{A$\cdot$2}$$

$$\textbf{path III} \quad \frac{\exp[-\gamma C_{\mathrm{III}}]}{\sum_{r=\mathrm{I}}^{\mathrm{III}} \exp[-\gamma C_r]} = \frac{x^3}{x^3+2x^2} \tag{A$\cdot$3}$$

where $x = \exp[-\gamma]$.

As discussed in Sect. 3.1, we next calculate the transition probability from node i to node j. From (13), matrix $\boldsymbol{A}$ for this network is

$$\boldsymbol{A} = \begin{bmatrix} 0 & x & x & 0 \\ 0 & 0 & x & x \\ 0 & 0 & 0 & x \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{A$\cdot$4}$$

and hence $\boldsymbol{W}$ is calculated from (20) as follows:

$$\boldsymbol{W} = \begin{bmatrix} 1 & x & x^2+x & x^3+2x^2 \\ 0 & 1 & x & x^2+x \\ 0 & 0 & 1 & x \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{A$\cdot$5}$$

Then, from (9), we obtain

$$\begin{aligned}\{p(2|1), p(3|1)\} &= \left\{\exp[-\gamma c_{12}]\frac{W_{24}}{W_{14}}, \exp[-\gamma c_{13}]\frac{W_{34}}{W_{14}}\right\} \\ &= \left\{x \times \frac{x^2+x}{x^3+2x^2}, x \times \frac{x}{x^3+2x^2}\right\},\end{aligned} \tag{A$\cdot$6}$$

$$\{p(3|2), p(4|2)\} = \left\{x \times \frac{x}{x^2+x}, x \times \frac{1}{x^2+x}\right\}, \tag{A$\cdot$7}$$

$$\{p(4|3)\} = \left\{x \times \frac{1}{x}\right\}. \tag{A$\cdot$8}$$

It is certain that the Markov assignment described by (A· 6)–(A· 8) is equivalent to (A· 1)–(A· 3), for example

$$\textbf{path I} \quad p(2|1) \times p(4|2) = \frac{x^2}{x^3+2x^2}. \tag{A$\cdot$9}$$

This is a brief example of MLB-routing and its extension to hop-by-hop routing.

Yudai Honma received his B.S., M.S., and Ph.D. degrees from Keio University, Tokyo, Japan, in 2004, 2006, and 2007, respectively. From April 2007 to September 2007, he was a Research Fellow of the Japan Society for the Promotion of Science (for Doctoral Course). From October 2007 to March 2008, he was a Research Fellow of the Japan Society for the Promotion of Science (for Postdoctoral). From April 2008 to March 2010, He was an Assistant Professor of the Faculty of System Design, Tokyo Metropolitan University. He has been an Assistant Professor of the Waseda Institute for Advanced Study, Waseda University since April 2010. His current interests are in the area of social systems and management. He is a member of the Operations Research Society of Japan.

Masaki Aida received his B.S. and M.S. degrees in Theoretical Physics from St. Paul's University, Tokyo, Japan, in 1987 and 1989, respectively, and the Ph.D. in Telecommunications Engineering from the University of Tokyo, Japan, in 1999. In 1989, he joined NTT Laboratories. In April 2005 to March 2007, he was an Associate Professor of the Faculty of System Design, Tokyo Metropolitan University. He has been a Professor of the Graduate School of System Design, Tokyo Metropolitan University since April 2007. His current interests include traffic issues in computer communication networks. He is a member of the IEEE and the Operations Research Society of Japan.

Hideyuki Shimonishi received his M.E. and Ph.D. degrees from Graduate School of Engineering Science, Osaka University, Osaka, Japan, in 1996 and 2002, respectively. He joined NEC Corporation in 1996 and has been engaged in research on traffic management in high-speed networks, traffic control protocols, switch and router architectures, and programmable/virtualized network architecture. He was a visiting scholar at Computer Science Department, University of California at Los Angeles, to study next generation transport protocols.

Atsushi Iwata received the B.E., and M.E., and Ph.D. degrees in electrical engineering from the University of Tokyo, Japan, in 1988, 1990, and 2001 respectively. In 1990, he joined NEC Corporation, Kanagawa, Japan. From 1997 to 1998, he was a Visiting Researcher at the University of California, Los Angeles. He has been a Senior Manager of IP Network Division since 2009, focusing on networking products for data center and fixed and mobile carrier network. His current research interest is the design and analysis of architectures, algorithms and protocols for computer communication networks. He is a member of the IEICE of Japan. He is an author of a book "Wide Area Ethernet Technology Overview" published by IEICE in 2005.