PAPER

An Application-Level Routing Method with Transit Cost Reduction Based on a Distributed Heuristic Algorithm

Kazuhito MATSUDA[†a)], **Go HASEGAWA**[††b)], ***and*** **Masayuki MURATA**[†c)], ***Members***

SUMMARY Application-level routing that chooses an end-to-end traffic route that relays other end hosts can improve user-perceived performance metrics such as end-to-end latency and available bandwidth. However, selfish route selection performed by each end user can lead to a decrease in path performance due to overload by route overlaps, as well as an increase in the inter-ISP transit cost as a result of utilizing more transit links compared with native IP routing. In this paper, we first strictly define an optimization problem for selecting application-level traffic routes with the aim of maximizing end-to-end network performance under a transit cost constraint. We then propose an application-level traffic routing method based on distributed simulated annealing to obtain good solutions to the problem. We evaluate the performance of the proposed method by assuming that PlanetLab nodes utilize application-level traffic routing. We show that the proposed routing method can result in considerable improvement of network performance without increasing transit cost. In particular, when using end-to-end latency as a routing metric, the number of overloaded end-to-end paths can be reduced by about 65%, as compared with that when using non-coordinated methods. We also demonstrate that the proposed method can react to dynamic changes in traffic demand and select appropriate routes.

key words: *overlay network, overlay routing, inter-ISP transit cost, PlanetLab, simulated annealing*

1. Introduction

Application-level (AL) traffic routing, as shown in Fig. 1, is a routing mechanism that works on the application layer and chooses an end-to-end route to other end hosts. Recent studies have revealed that such routing can improve user-perceived performance metrics such as end-to-end latency, available bandwidth and packet loss ratio compared with those in native IP routing [1]–[4]. In the rest of this paper, we refer to end hosts (which can be senders, relay hosts, and destinations) as AL nodes.

Such AL route selection can inflate the monetary cost incurred by Internet service providers (ISPs) as a consequence of increasing the number of transit links along the route, where monetary cost is determined according to the amount of traffic traversing the links (we refer to the monetary cost as *transit cost*) [5]. Such a situation can be expected because the AL route that relays an AL node includes more than one IP-level path. Furthermore, selfish AL route selection performed by multiple application users can lead to a decrease in path performance due to overload by route overlaps. For example, in [6] a number of non-coordinated overlay networks cause oscillations in route selection due to concentration of traffic at certain links.

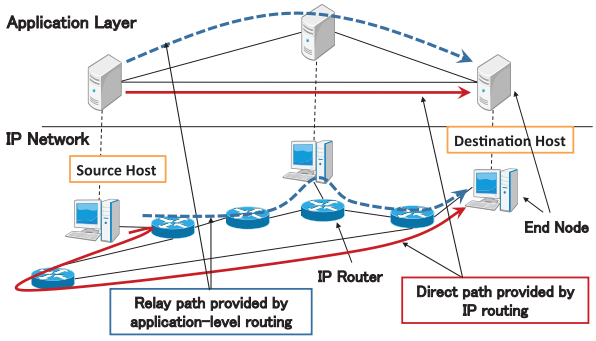

Fig. 1 AL traffic routing.

In [7], we demonstrated that the number of transit links traversed by AL-routed traffic could be estimated, and then controlled by using end-to-end network performance metrics. The results showed the possibility of reducing the transit cost generated by AL traffic routing without precise information about transit and peering links on the routes. In that study, however, we did not evaluate the influence of route overlaps, which means that the proposed method in [7] is a selfish AL route selection method. Hence a coordinated method of addressing the influence of route overlaps is still needed.

In this paper, we focus on AL traffic routing based on coordination performed by AL nodes, with the aim of improving end-to-end network performance without increasing transit cost. First, we formulate the AL traffic routing and strictly define an optimization problem for selecting AL traffic routes with various route selection metrics and a constraint on the transit cost. In general, there are two candidates of coordinated algorithms to achieve good solutions for the optimization problem: centralized and distributed algorithms. In this work, we assume that the operator of each AL node wants to decide the AL route on its own. For example, we can easily imagine a use case where each AL node is independently controlled by an ISP, and routes are provided to each of the ISP's customers. In such a case, a distributed algorithm is more desirable than a centralized one. Therefore, we propose an AL traffic routing method based on a

Manuscript received July 23, 2012.
Manuscript revised February 4, 2013.

[†]The authors are with the Graduate School of Information Science and Technology, Osaka University, Suita-shi, 560-0871 Japan.
[††]The author is with Cybermedia Center, Osaka University, Toyonaka-shi, 560-0043 Japan.
a) E-mail: k-matuda@ist.osaka-u.ac.jp
b) E-mail: hasegawa@cmc.osaka-u.ac.jp
c) E-mail: murata@ist.osaka-u.ac.jp
DOI: 10.1587/transcom.E96.B.1481

distributed heuristic algorithm that produces good solutions to the optimization problem. We also design the proposed method to perform route selection not only for a fixed AL traffic demand, but also in reaction to dynamic AL traffic demand changes.

We evaluate the proposed method by assuming that PlanetLab nodes utilize AL routing using the end-to-end measurement results of the network performance values. We first evaluate the proposed method assuming fixed amounts of traffic demand between each AL node pair. Next, we evaluate the effectiveness of the proposed method in a situation where the amount of AL traffic demand fluctuates over time. In both cases, we compare performance between the proposed and non-coordinated methods, and confirm the effectiveness of the proposed method.

The remainder of this paper is organized as follows: In Sect. 2, we describe the background of the present research. In Sect. 3, we define the optimization problem for AL route selection. In Sect. 4, we propose a novel AL traffic routing method. In Sect. 5, we show the results of evaluating the proposed method. Finally, in Sect. 6, we present our conclusions and describe avenues of future research.

2. Pros and Cons of AL Traffic Routing

2.1 Improvement of End-to-End Network Performance

The advantage of AL traffic routing (simply AL routing, below) for end-to-end network performance is mainly from the policy mismatch between native IP routing and AL routing. Native IP routing is based primarily on metrics such as router-level and AS-level hop counts, which do not always correlate to user-perceived performance. In addition, ISPs have their own cost structures based on commercial contracts with their neighboring ISPs affecting the IP routing. Two types of links are common between ASes: transit links that connect upper- and lower-level ISPs, and peering links used for peering relationships[†]. The monetary cost of the transit link is usually determined by the amount of traffic traversing the link. In contrast, there is almost no monetary cost for peering links, except for that of the physical link facilities. ISPs make IP-level routing decisions by considering such differences between transit and peering links.

Figure 1 shows a typical example of improving network performance by AL routing. We assume that IP routing uses the direct path and AL routing chooses the relay path. The length of the arrows represents the end-to-end latency value. Comparing the direct and relay paths, the direct path has a lower router-level hop count but a higher end-to-end latency. Therefore, AL routing provides lower end-to-end latency than the IP routing. For example, [8] showed from their evaluation results for a PlanetLab environment that AL routing could reduce end-to-end latency in over 80% of end-to-end paths.

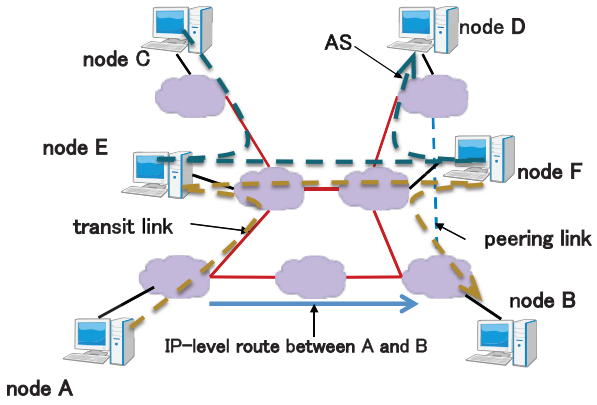

Fig. 2 Problems on AL routing.

2.2 Route Overlaps and Impact on ISP Cost Structure

Although AL routing can improve user-perceived performance, we can expect situations where certain AL links that can provide high network performance are utilized by many AL routes, because AL routing users make selfish routing decisions. This situation degrades the benefits of AL routing. Figure 2 shows a simple example of this problem in which six end hosts, each of which works as an AL node, are connected by AL links. We assume that Node A generates traffic that is routed to Node B, and Node C generates traffic to Node D. When the AL link between Nodes E and F provides high network performance, both node pairs A–B and C–D try to use the AL link between Nodes E and F. As a result, the network performance of both pairs may degrade, for example increasing end-to-end latency or decreasing available bandwidth.

Furthermore, this may also generate traffic that does not follow the ISPs' cost structure (the IP routing policy provided by ISPs), so ISPs may incur additional monetary costs. If these costs accumulate, the transit cost over the entire network increases. For example, in Fig. 2 each AL link includes multiple inter-AS links, each of which is either a transit link (solid line) or a peering link (dashed line). We assume that Node A generates traffic that is routed to Node B. When using native IP routing or AL routing that chooses the direct path, the traffic traverses two transit links. Conversely, when AL routing utilizes the relay path via Nodes E and F, the traffic traverses three transit links: those between Nodes A and E, those between Nodes E and F, and those between Nodes F and B. Therefore, the sum of the transit links traversed by the relay path is increased by one compared with the direct path and, as a consequence, the transit cost over the entire network increases.

3. AL Route Optimization Problem

We begin this section by explaining the network model as-

[†]We ignore sibling links because they connect ASes belonging to the same organization.

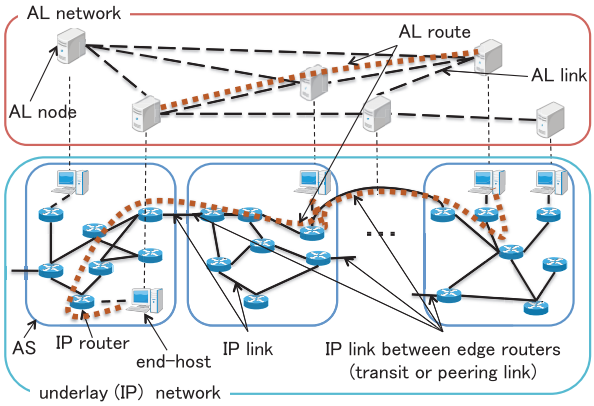

Fig. 3 Network model.

sumed in this paper. We then formulate the AL routing and define the optimization problem for selecting AL routes.

3.1 Network Model

We assume a network model as depicted in Fig. 3. The underlay IP network is constructed from a number of IP-level routers, each of which is located at one of the ASes. There is at most one link between each IP-level router pair. IP-level routers located at the edge of an AS connect to IP-level routers located at the edge of one or more ASes by transit or peering links. Note that a transit cost is incurred when traffic traverses transit links. AL nodes that utilize AL routing reside on end hosts connected to IP-level routers. The AL nodes are connected to each other by AL links, which constitute the AL network. Each AL link equals to the native IP-level path between the corresponding AL node pair. AL routing is performed on the AL network and determines the AL routes between AL node pairs that have traffic demand. For example, in Fig. 3, the AL route drawn with a dotted line consists of two AL links, each of which is a native IP-level path between the end hosts.

3.2 Optimization Problem for AL Routing

We formulate the IP routing in an underlay IP network. Here, N represents the number of IP-level routers and M represents the number of links in the underlay network. We assign an identifier $1 \ldots M$ to each link.

Since there are N routers, we can consider $(N-1)N$ IP-level routes between all router pairs. We then assign an identifier $1 \ldots (N-1)N$ to each pair of source and destination routers. Note that the order of router pairs is irrelevant to the following discussion. We define the IP routing matrix R^{IP} as below. The subscripts and superscripts respectively assign rows and columns in the order of $1, 2, \ldots, (N-1)N$ and $1, 2, \ldots, M$.

$$R^{\mathrm{IP}} = \begin{pmatrix} IP_1^1 & \cdots & IP_1^{(N-1)N} \\ \vdots & \ddots & \vdots \\ IP_M^1 & \cdots & IP_M^{(N-1)N} \end{pmatrix} \tag{1}$$

When link i exists on the route for router pair j, the value of element IP_i^j is one, otherwise zero.

Next, we consider an AL network constructed from AL nodes and AL links. We assume that end hosts can be connected to all IP-level routers, which can be AL nodes. Therefore, we can consider $(N-1)N$ AL links between all possible AL node pairs. Note that we consider the direction of AL links. We assign an identifier to each AL node pair, which is the same as the corresponding IP-level router pair whose source and destination routers connect to the source and destination AL nodes. The AL network topology $\mathcal{E}$ can be expressed as follows:

$$\mathcal{E} = \{e_1^{\mathrm{AL}}, e_2^{\mathrm{AL}}, \ldots, e_{(N-1)N}^{\mathrm{AL}}\} \tag{2}$$

where the value of e_j^{AL} is one when the source and destination AL nodes exist and they are connected through the AL link between AL node pair j, otherwise zero.

Here, we describe an AL route for AL node pair j as $r_j = (p_1, p_2, \ldots, p_h)$, which indicates that the AL route utilizes the AL links between AL node pairs $p_1, p_2, \ldots, p_h$, in that order.

The set of available AL routes for AL node pair j in the AL network, Γ_j^{AL}, is described as follows:

$$\begin{aligned} \Gamma_j^{\mathrm{AL}} = \{(p_1, p_2, \ldots, p_h) | h \geq 1,\ & s_{p_1} = s_j, t_{p_h} = t_j, \\ & t_k = s_{k+1}\ (2 \leq h,\ 1 \leq k \leq h-1), \\ & e_{p_k}^{\mathrm{AL}} = 1\ (1 \leq k \leq h)\} \end{aligned} \tag{3}$$

where s_j and t_j respectively represent the source and the destination nodes of AL node pair j.

As for the AL links, we can consider $(N-1)N$ AL routes and use the same identifiers for AL node pairs of AL routes as for AL links. Note that we assume that the AL routing determines the AL routes only for AL node pairs that have traffic demand. Here, we define the AL routing matrix as follows:

$$R^{\mathrm{AL}} = \begin{pmatrix} AL_1^1 & \cdots & AL_1^{(N-1)N} \\ \vdots & \ddots & \vdots \\ AL_{(N-1)N}^1 & \cdots & AL_{(N-1)N}^{(N-1)N} \end{pmatrix} \tag{4}$$

When the AL link between AL node pair i exists on the AL route for AL node pair j, the value of element AL_i^j is one, otherwise zero. Note that AL_i^j $(\forall i | i \in \{1, 2, \ldots, (N-1)N\})$ become zero if node pair j has no traffic demand.

We divide the whole network traffic into two parts, traffic carried only by IP routing and traffic carried by AL routing. We describe the traffic demand on router pairs carried by IP routing as $\mathcal{X}^{\mathrm{IP}} = (x_1^{\mathrm{IP}}\ x_2^{\mathrm{IP}} \cdots x_{(N-1)N}^{\mathrm{IP}})$, and the traffic demand on AL node pairs carried by AL routing as $\mathcal{X}^{\mathrm{AL}} = (x_1^{\mathrm{AL}}\ x_2^{\mathrm{AL}} \cdots x_{(N-1)N}^{\mathrm{AL}})$. Here, x_j^{IP} and x_j^{AL} denote the traffic demand corresponding to router pair j and AL node pair j, respectively. Then, we can calculate the matrix $\mathcal{Y}$, which represents the load on the links between routers:

$$\mathcal{Y} = R^{\mathrm{IP}}\mathcal{X}^{\mathrm{IP}} + R^{\mathrm{IP}}R^{\mathrm{AL}}\mathcal{X}^{\mathrm{AL}} \tag{5}$$

We introduce a function f_D, which calculates the latencies of all AL links under traffic load $\mathcal{Y}$. Then, we can calculate the latencies of all AL routes. The matrix $\mathcal{D}^{\mathrm{AL}} = (d_1^{\mathrm{AL}}\ d_2^{\mathrm{AL}} \cdots d_{(N-1)N}^{\mathrm{AL}})$, where the latencies of the AL routes are set in rows, can be described as follows (note that each element d_j^{AL} represents the latency of the AL route between AL node pair j):

$$\mathcal{D}^{\mathrm{AL}} = f_D(\mathcal{Y})R^{\mathrm{AL}} \tag{6}$$

For the available bandwidth, we define a function f_B, which calculates the available bandwidths for all AL routes under traffic load $\mathcal{Y}$ and AL routing matrix R^{AL}. Note that f_B directly calculates the available bandwidths for the AL routes, because the available bandwidths are determined not by the sum of values of the used AL links but by the value of the narrowest AL link. Using f_B, we can express $\mathcal{B}^{\mathrm{AL}} = (b_1^{\mathrm{AL}}\ b_2^{\mathrm{AL}} \cdots b_{(N-1)N}^{\mathrm{AL}})$ as follows:

$$\mathcal{B}^{\mathrm{AL}} = f_B(\mathcal{Y}, R^{\mathrm{AL}}) \tag{7}$$

We assume that the transit cost of an AL route is determined by the traffic load and the number of transit links on the route. Based on that assumption, in the case of the transit cost, $\mathcal{C}^{\mathrm{AL}} = (c_1^{\mathrm{AL}}\ c_2^{\mathrm{AL}} \cdots c_{(N-1)N}^{\mathrm{AL}})$, can be expressed as follows in the same way as the available bandwidth.

$$\mathcal{C}^{\mathrm{AL}} = f_C(\mathcal{Y}, R^{\mathrm{AL}}) \tag{8}$$

We now define the constraint on transit cost in AL route selection. We treat the transit cost of the direct paths as a baseline, and constrain the increase in transit cost of the AL paths compared with that of the direct paths. Here, the routing matrix of the direct paths can be described as follows:

$$R^{\mathrm{DR}} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \tag{9}$$

The transit cost of the direct paths, $\mathcal{C}^{\mathrm{DR}} = (c_1^{\mathrm{DR}}\ c_2^{\mathrm{DR}} \cdots c_{(N-1)N}^{\mathrm{DR}})$, can be described with R^{DR} as follows:

$$\mathcal{C}^{\mathrm{DR}} = f_C(\mathcal{Y}, R^{\mathrm{DR}}) \tag{10}$$

Then, we can describe the constraint on the increase in the transit cost of AL path between node pair j as follows:

$$c_j^{AL} \leq \alpha c_j^{DR}\ (\forall j | j \in \Theta) \tag{11}$$

where Θ is the set of identifiers of AL node pairs that have traffic demand. Equation (11) means that AL routing can select only the AL paths whose transit cost is lower than that of the corresponding direct paths multiplied by α. We use the equation when the AL routing selects AL routes to limit the increase in the transit cost.

The AL routing determines AL routes only for AL node pairs that have traffic demand. The problem of minimizing the average latency of the AL routes between AL nodes that have traffic demand is described as follows, where the AL routes between AL nodes $r_j (j \in \Theta)$ are treated as variables:

$$\begin{aligned} \text{minimize}: \quad & \left(\sum_{j\in\Theta} d_j^{\mathrm{AL}}\right) \Big/ |\Theta| \\ \text{subject to}: \quad & r_j \in \Gamma_j^{\mathrm{AL}} \\ & c_j^{AL} \leq \alpha c_j^{DR}\ (\forall j | j \in \Theta) \end{aligned} \tag{12}$$

We can also describe the maximization problem for the available bandwidth as follows:

$$\begin{aligned} \text{maximize}: \quad & \left(\sum_{j\in\Theta} b_j^{\mathrm{AL}}\right) \Big/ |\Theta| \\ \text{subject to}: \quad & r_j \in \Gamma_j^{\mathrm{AL}} \\ & c_j^{AL} \leq \alpha c_j^{DR}\ (\forall j | j \in \Theta) \end{aligned} \tag{13}$$

4. Proposed Method

In this section, we propose an AL routing method, based on obtaining good solutions to the problem described in Sect. 3. For this purpose, we take advantage of a popular heuristic algorithm known as *simulated annealing* (SA). As described in Sect. 1, because the distributed algorithm is desirable for application scenarios of AL routing, we utilize the distributed simulated annealing (DSA) proposed in [9]. In the remainder of this section, we propose two algorithms for the AL routing method, one for static AL traffic demand and the other as the algorithm reacting to dynamic AL traffic demand changes.

4.1 Algorithm for Static Route Selection

In general, the SA process continues through the decision of whether to change the *state*, which is a solution to the target problem, to its neighbor that is slightly different from the current state. The decisions are made stochastically based on two parameters, namely, *temperature* and *cost*. The cost represents the goodness of the state and determines the probability of accepting the state. The temperature also determines the probability, and it gradually decreases as the process continues. The process finishes when the temperature becomes sufficiently low. DSA is a distributed heuristic algorithm, in which an individual agent has the right to decide a part of the state. Utilizing DSA, we can search for a good solution in a cooperative manner under an environment consisting of a number of individual domains. In the DSA process, agents generate a neighbor state by changing the associated parts of the current state and determine whether to accept the neighbor state. Then, the agents exchange their associated parts of the state with each other to share the whole state, and calculate the cost of the obtained state. After that, the agents repeat these steps as in SA. The approach used in DSA, which entails dividing up the problem and solving it with a number of distributed agents, can be easily applied to AL route selection by distributed AL nodes. DSA cannot guarantee a near-optimal solution, but we have verified that AL route selection using DSA gives a

good solution, even when compared with the optimal solution in an environment containing few AL nodes [10].

To apply the DSA algorithm to AL routing, we define a state as a set of AL routes of all AL node pairs that have traffic demand, and the cost as the estimated network performance obtained by the AL route selections of the state. Each AL node handles the AL links and AL routes originating from itself. To share the network status among AL nodes, each node measures the performance of AL links without the AL traffic and shares the measurement results, as well as the AL traffic demands, among all AL nodes at the beginning of AL routing. Using the exchanged information, each AL node estimates the performance of AL links when AL traffic is added to the network, and conducts the DSA algorithm to determine the AL routes. Note that, in regard to measurement overhead, AL routing normally requires $n(n-1)$ measurements among the n AL nodes. Some existing methods can reduce this measurement overhead (e.g., [11], [12]). However, a specific measurement method is beyond the scope of the present paper.

The pseudocode for the algorithm (called the static algorithm below) is shown in Algorithm 1. The function Random(x) returns a random positive value less than x. T_{low} is set to a sufficiently small positive value nearly equal to zero. Note that a subscript i in the algorithm indicates that the algorithm is run on the i-th AL node. In what follows, we omit the subscript for simplicity. We describe the parameters and functions required for Algorithm 1 in detail.

Initial state S_{init}

The initial state is the state used at the beginning of the algorithm. Each AL node has its own initial state in which direct routes are utilized for all AL node pairs. The direct routes are the same as the IP-level routes. Therefore, when all AL nodes are IP-reachable, there exists a route between any two AL nodes, which corresponds to the direct route. In addition, the information about direct routes is exchanged among AL nodes. Consequently, each AL node can construct the initial state without a global view of the network. On these grounds, we assume that the initial state consisting of the direct routes is feasible.

Neighbor-generation function Neighbor()

This function takes a state as its argument and returns a neighbor state. A *neighbor state* of state S for an AL node is defined as a state where some AL routes in S originating from itself are changed. Randomly according to uniform distribution, the function first chooses a portion of AL routes in the state S originating from the node. Then, the function generates a neighbor state by changing the selected AL routes to randomly selected AL routes from the candidate AL routes. Note that the candidate AL routes are restricted by the constraint conditions in Eqs. (12) and (13).

Cost function Cost()

This function estimates the network performance obtained by the given state as the argument and returns the average end-to-end latency or average available bandwidth of all AL routes. These costs correspond to the optimization problems (Eqs. (12) and (13)). In addition, we normalize the state cost by the initial state cost so that the transition probability is not affected by the absolute value of the cost.

Algorithm 1 Algorithm for static route selection on AL node i

```
1: I_i ← 0,   T_i ← T_init,   S_i ← S_init
2: while T_i > T_low do
3:     Update(S_i)
4:     S_tmp ← Neighbor(S_i)
5:     if Cost(S_i) ≥ Cost(S_tmp) then
6:         S_i ← S_tmp
7:     else
8:         r_i ← Random(1)
9:         if r_i < Probability(T_i, Cost(S_i), Cost(S_tmp)) then
10:            S_i ← S_tmp
11:        end if
12:    end if
13:    I_i ← I_i + 1
14:    T_i ← Cooling(T_i, I_i)
15:    if I_i mod U_i = 0 then
16:        Notification(S_i)
17:    end if
18: end while
```

Transition probability function Probability()

Here, we utilize a typical function in SA. The equation is as follows:

$$\text{Probability}(T, S, S_{tmp}) = e^{-\frac{\text{Cost}(S_{tmp})-\text{Cost}(S)}{T}} \tag{14}$$

where T, S, and S_{tmp} are the current temperature, the current state, and the neighbor state of the current state when the function is executed, respectively.

Initial temperature T_{init} **and cooling schedule function** Cooling()

In the general SA algorithm, the initial temperature must be set sufficiently high to induce a transition from the current state to its neighbor state regardless of the cost of the neighbor state [13]. We use the following typical cooling schedule function in SA:

$$\text{Cooling}(T, I) = \gamma T \ (0 < \gamma < 1) \tag{15}$$

Update function Update()

This function updates the current state of the AL node with the AL routes and AL traffic demands received from other AL nodes.

Notification function Notification()

This function sends to the other AL nodes the currently-selected AL routes and AL traffic demands originating from itself. Although the cost function requires the AL routes selected by all AL nodes, the communication overhead becomes high if the AL routes are gathered on each update of the state at each AL node. Then, the function is executed every U iterations of SA.

4.2 Algorithm for Dynamic Route Selection

Next, we propose a route selection algorithm, which we call the dynamic algorithm below, that dynamically reacts to AL traffic demand changes. We construct the dynamic

Algorithm 2 Algorithm for dynamic route selection

```
1: T_i ← T_init
2: C_i ← 0
3: loop
4:     StaticAlgorithm(T_i)
5:     while T_i = 0 do
6:         C_i ← CountChanges(C_i)
7:         if C_i ≥ C_th then
8:             T_i ← T_re
9:             C_i ← 0
10:        end if
11:    end while
12: end loop
```

algorithm by extending the static algorithm. The dynamic algorithm first runs the static algorithm. After that, the algorithm enters an idle state until the accumulation of traffic changes exceeds a threshold, at which time it executes the static algorithm again.

When developing a dynamic algorithm, we need to consider the changes in the performance of AL links without AL traffic, which are caused by fluctuations in the background traffic. In the proposed method, each AL node measures the performance of AL links when their estimated performance is far from actual performance.

The pseudocode for the dynamic algorithm is shown in Algorithm 2, where StaticAlgorithm() means the execution of Algorithm 1. In what follows, the additional parameters and function required for Algorithm 2 are described in detail.

Function for counting traffic changes CountChanges() **and threshold** C_{th}

This function observes traffic changes between AL nodes. When AL traffic demand originates from the node itself, the function returns the same value as the threshold C_{th}, meaning that StaticAlgorithm() is immediately executed to determine a better route for the new traffic. On the other hand, when AL traffic originating itself terminates, or when AL traffic demand originating another AL node occurs or terminates, the function counts these events and StaticAlgorithm() is executed when the count reaches C_{th}.

Temperature for re-execution of the static algorithm T_{re}

The temperature for re-execution of the static algorithm should be equal to or lower than the initial temperature, because the state at the beginning of re-execution is the result of the previous execution of Algorithm 1.

5. Evaluation

In this section, we show the evaluation results of the proposed algorithms described in Sect. 4, assuming that the PlanetLab nodes constitute an AL network and conduct AL routing.

5.1 Dataset and Settings

5.1.1 Dataset

To construct the IP-level and AS-level network topologies and determine the network performance between each AL node pair for performance evaluation, we use the measurement results of the network performance values for the 657 PlanetLab nodes. Below, we describe the process of obtaining the network performance values.

End-to-end latencies, IP-level routes

We conducted `traceroute` commands for all PlanetLab nodes. We use results obtained on October 19, 2010.

Available bandwidths and physical capacities

We obtained the available bandwidths and physical capacities between all PlanetLab nodes from the Scalable Sensing Service (S^3) [14]. S^3 provides the measurement results among PlanetLab nodes every 4 hours. In this paper, we use the measurement results obtained on October 18–19, 2010.

AS-level routes

We converted the IP-level routes into AS-level routes by using the relationships between IP address prefixes and AS numbers, available at the Route Views Project [15]. We use the data obtained on April 16, 2009. Although the AS number data are older than the other data, we believe this does not affect the evaluation results because attached AS numbers are not changed frequently.

The relationships between ASes

We utilize the relationships between ASes as provided by CAIDA [16] on January 20, 2010, to calculate the transit cost of AL routes, as described below.

5.1.2 Cost Functions

In what follows, we explain the functions f_D, f_B, and f_C in Eqs. (6)–(8) for evaluating the state cost and the transit cost. We define f_D as the function that derives the sum of the propagation delay and the queuing delay that may occur by the current state in the process of the proposed method. For details, we first make the following assumptions:

- None of the AL links share any IP links.
- For each AL link, the tight link for the available bandwidth and the narrow link for the physical capacity are identical, and we can measure these values with end-to-end measurement methods. Note that we do not consider the effect of the traffic generated by measurement on the network.
- The queuing delay at an AL link occurs only at the tight IP link.
- The queuing delay included in the measured delay is negligibly small compared to that caused by the AL traffic. This is because the utilization ratio derived by the obtained measurement result is low, which indicates that the queuing delay is very small.

With the above assumptions, we can regard the delay that is measured by each AL node in the absence of AL traffic as propagation delay. We also calculate the queuing delay of AL links based on the M/M/1 queuing model. The queuing delay of the AL link between AL node pair j, d_j^q is calculated as follows:

$$d_j^q = \frac{\frac{g_j - a_j + x_j}{c_j}}{1 - \frac{g_j - a_j + x_j}{c_j}} \cdot \frac{P}{g_j} \tag{16}$$

where g_j, a_j, and x_j are the physical capacity, the measured available bandwidth, and the AL traffic demand of the AL link between AL node pair j, respectively. P is the average packet size. We use 770 bytes as the value of P, which is the average value calculated with the typical maximal packet size of 1500 bytes and the TCP ACK packet size of 40 bytes. Then, the end-to-end latency of the AL link between node pair j, d_j is calculated as follows:

$$d_j = d_j^q + d_j^p \tag{17}$$

We define f_B as the function that derives the bandwidth that can be achieved by the AL routes when sharing the measured available bandwidth of AL links among other AL routes, which is based on simple max-min bandwidth sharing [17]. The bandwidth achieved by an AL route on the AL link between the AL node pair i, $f_{\mathcal{B}}(i)$ is calculated as follows:

$$f_{\mathcal{B}}(i) = \left(a_i - \sum\nolimits_{j \in \Theta^i} b_j^{\mathrm{AL}}\right) / |\{k | b_k^{\mathrm{AL}} = 0, k \in \Theta^i\}| \tag{18}$$

where a_i represents the measured available bandwidth of the AL link between AL node pair j, z_j is the number of traffic flows, and Θ^i represents the set of node pairs that utilize the AL link between AL node pair i. The calculation of Eq. (18) progresses in ascending order of the available bandwidth of the AL links. The bandwidth of the AL route between node pair j is determined using $f_{\mathcal{B}}(i)$ according to the following equation:

$$b_j^{\mathrm{AL}} = f_{\mathcal{B}}(i) \ \ (j | b_j^{\mathrm{AL}} = 0, j \in \Theta^i) \tag{19}$$

The function f_C calculates the transit costs of the AL routes based on the amount of AL traffic demand between AL nodes and inter-AS relationships (transit or peering). We calculated the transit cost of the AL route between the AL node pair j, c_j^{AL} as follows:

$$c_j^{\mathrm{AL}} = \beta_j x_j \tag{20}$$

where x_j represents the AL traffic demand on the AL link between AL node pair j. Here, β_j determines the transit cost per unit amount of traffic, which can be determined by the types of IP links traversed by the AL route. In the evaluation, we used the values of IP link i on the AL route between AL node pair j, v_i^j as follows:

$$v_i^j = \begin{cases} 1 & (IP_i^j = 1 \text{ and } i \text{ is a transit link}) \\ 0.05 & (IP_i^j = 1 \text{ and } i \text{ is a peering link}) \\ 0 & (IP_i^j = 0 \text{ or } i \text{ is not AS-level link}) \end{cases} \tag{21}$$

The value of β_j was calculated from Eq. (22) as follows:

$$\beta_j = \sum\nolimits_{i=1}^{M} v_i^j \tag{22}$$

Note that in cases where we are unable to obtain the measurement results of the network performance values of the AL links, we do not use those AL links in the AL routing.

5.1.3 Evaluation Scenarios and Metrics

The evaluation scenarios for the static and dynamic algorithms are as follows. For the static algorithm with end-to-end latency as a routing metric, we assume an AL traffic demand of 1 Mbps for 50% of AL node pairs, 3 Mbps for 30% of pairs, 5 Mbps for 15% of pairs, and 10 Mbps for 5% of pairs. For performance evaluation metrics, we use the average end-to-end latency, the number of AL routes that use overloaded AL links (referred to as overloaded AL routes below), and the transit cost of all AL node pairs that have traffic demand. We also observe a part of AL routes to confirm where the effectiveness of the proposed method comes from. For the case of available bandwidth as a routing metric, we assume that 50% of all AL node pairs have traffic demand and require bandwidth. We then evaluate the distribution of the available bandwidth between all AL node pairs.

For the dynamic algorithm with end-to-end latency as a routing metric, we set the AL traffic demand among AL node pairs according to [18]. That is, traffic flows where each of them requires 100 kbps are generated in accordance with a Weibull distribution, and their durations are determined by the log-normal distribution. The source and destination of each flow is randomly chosen from all AL node pairs. Because the parameters for the two distributions shown in [18] are achieved by the observation on the access link of only one AS, we accumulate a number of traffic flows for generating inter-AS traffic. We refer to the number of traffic flows as the *traffic accumulation degree*. We assume that the AL traffic demand changes at 2 sec intervals, and that the degree changes in the order of 1, 3, 10, and 5. Using this setting, we evaluate the end-to-end latency performance of AL routes selected by the proposed method. We also observe that the AL route changes when the AL traffic demand changes to demonstrate that the proposed method can react to AL traffic demand changes. For the case of available bandwidth as a routing metric, we consider the situation where the number of AL node pairs that require bandwidth changes over time. We assume that the changes occur at two-second intervals, and that the ratio of AL node pairs that require the bandwidth changes in the order of 10%, 80%, and 40%. We then evaluate the changes in the average available bandwidth.

5.1.4 Other Settings

We assume that at the beginning of the proposed method,

all parts of the initial state and the measured performance of AL links have been already exchanged among all AL nodes. In the evaluation for the dynamic algorithm, we assume the background traffic is not changed during the entire evaluation. The neighbor-generation function randomly changes AL routes of 1% of AL node pairs originating from itself. We considered only one- and two-hop AL routes as candidate AL routes because AL routes with more than two AL links do not contribute to the improvement of end-to-end network performance [8]. Other parameters for the proposed method are shown in Table 1. Note that, under the settings, the number of iteration is roughly $2,400$ for the selection of one route by the proposed method, and AL nodes exchange the currently selected routes once every 20 iterations. Here, we assume that the information size of one AL route is 12 bytes, which includes the IP addresses of the source, destination, and relay AL nodes. A rough calculation of the traffic volume generated by each AL node every route selection is $12 \times (30 \times 29) \times (2400/20) = 1,252,800$ bytes, where (30×29) is the number of AL node pairs. In fact, the actual overhead depends on the interval of AL route updates, but it is easily executable on the current Internet at a realistic interval, for example, from a few minutes to an hour.

For comparison, we show the evaluation results for a non-coordinated route selection method. That is, each AL node pair independently selects the AL route that has the best network performance based on the measurement results of AL links before the route selection. We refer to this method as the *non-cooperative* method below.

5.2 Evaluation Results

5.2.1 Static Algorithm

We first show the evaluation results using end-to-end latency as a routing metric. Table 2 shows the average end-to-end latencies of the AL routes selected by the proposed and non-cooperative methods, which are classified by traffic demand values. We also show the number of overloaded AL routes to investigate the degree of congestion. For the proposed method, we show the results without the constraint on transit cost ($\alpha = \infty$ in Eq. (12)) and those with the constraint ($\alpha = 1$). From Table 2, we can observe that the proposed method provided slightly larger end-to-end latencies than did the non-cooperative method. However, the non-cooperative method generated the overloaded AL routes twice as much as the proposed method without the constraint on transit cost. Note that the average end-to-end latency was calculated excluding the overloaded AL routes. Therefore, the average end-to-end latency obtained by the non-cooperative method was smaller than that obtained by the proposed method.

The reason for the difference in the number of overloaded AL routes can be explained by Table 3, which presents the samples of the AL route selection results. Next to each AL node pair, values in parenthesis, (x, y), represent the number of overlapped utilizations of the AL link by the selected AL routes and the bandwidth utilization of the AL link, which is the ratio of the sum of background and AL traffic on the AL links to the physical capacity. From Table 3, we can see that the proposed method avoided overloaded AL routes in several patterns. For example, in the case of the AL route between *planetlab-2.ssvl.kth.se* and *planetlab1.ci.pwr.wroc.pl*, the proposed and non-cooperative methods selected the direct route. However, the number of overlaps in the proposed method was smaller than that in the non-cooperative method. This is because the proposed method shares the

Table 1 Parameters for the evaluation.

Number of AL nodes	30
T_{init} and T_{re}	0.15
γ in Eq. (15)	0.995
U	20
T_{low}	10^{-6}
C_{th}	10

Table 2 Average end-to-end latency classified by AL traffic demand and number of overloaded AL routes.

traffic demand	proposed method ($\alpha = \infty$)	proposed method ($\alpha = 1$)	non-cooperative method
1 Mbps	210 ms	217 ms	203 ms
3 Mbps	226 ms	226 ms	215 ms
5 Mbps	206 ms	209 ms	198 ms
10 Mbps	204 ms	195 ms	193 ms
number of overloaded AL routes	16	24	35

Table 3 Samples of selected AL routes with number of overlaps and bottleneck link utilization ratio of AL links.

source node destination node	(overlaps, utilization) or	source node relay node destination node	(overlaps, utilization) (overlaps, utilization)
proposed method ($\alpha = \infty$)		non-cooperative method	
planetlab-2.ssvl.kth.se planetlab1.ci.pwr.wroc.pl	(2, 0.78)	planetlab-2.ssvl.kth.se planetlab1.ci.pwr.wroc.pl	(4, 1.01)
planetlab2.cs.columbia.edu planetlab4.cs.duke.edu planetlab6.cs.cornell.edu	(1, 0.76) (5, 0.51)	planetlab2.cs.columbia.edu planetlab6.cs.cornell.edu	(2, 1.35)
ricepl-1.cs.rice.edu planetlab1.utep.edu planetlab-2.ssvl.kth.se	(3, 0.94) (2, 0.46)	ricepl-1.cs.rice.edu planetlab1.utep.edu planetlab-2.ssvl.kth.se	(3, 1.57) (2, 0.46)

AL route selection at other AL nodes, enabling avoidance of excessively overlapped utilization. For the case between *planetlab2.cs.columbia.edu* and *planetlab6.cs.cornell.edu*, the proposed method selected a detour AL route to avoid using the direct route that was overloaded. For the case between *ricepl-1.cs.rice.edu* and *planetlab-2.ssvl.kth.se*, although both methods used the same host as a relay node, the bandwidth utilization of the selected AL links by the proposed method was lower than that by the non-cooperative method. These samples indicate that the proposed method can select AL routes by considering the bandwidth utilization on AL links including AL traffic demand of other AL node pairs in a coordinated manner, thereby avoiding overlaps and overload on AL links.

Table 5 shows the average transit cost of the selected AL routes by the proposed method with and without the constraint on the transit cost. Although the transit cost could be reduced by 20% for the case with the constraint, we can observe that the overloaded AL routes increased compared with the case without the constraint in Table 2. This is because the number of AL links available for the proposed method with the constraint is smaller than those without the constraint.

We next present the evaluation results using the available bandwidth as a routing metric. Figure 4 shows the distribution of the available bandwidth of paths between AL node pairs. The average available bandwidth in the proposed method was 54,738 kbps, while that in the non-cooperative method was 26,570 kbps. We can observe from this figure that almost all AL node pairs could achieve considerable improvement by the proposed method compared with the non-cooperative method, which stems from the advantage of the proposed method in avoiding AL route overlaps as in the case of end-to-end latency shown in Table 3.

From the above results, we conclude that the proposed method, which implements coordination between AL nodes, can reduce congestion and avoid overlaps on AL links. The constraint on transit cost can efficiently decrease the cost incurred by the AL routing. On the other hand, the number of candidate AL links that can be used decreases compared with the case without the constraint.

To examine the effect of the frequency of AL route exchange, we set the value of U to 20, 200, and 2000. Table 4 shows the results in the same manner as Table 2. From the results, we can see that the difference due to the value of U is very small, though a larger value of U generates a few additional overloaded AL routes.

Table 4 Effect of the frequency of AL route exchange.

	$U = 20$	$U = 200$	$U = 2000$
1 Mbps	210 ms	215 ms	211 ms
3 Mbps	226 ms	229 ms	227 ms
5 Mbps	206 ms	214 ms	227 ms
10 Mbps	204 ms	100 ms	201 ms
number of overloaded AL routes	16	18	18

Table 5 Average transit cost of the AL routes.

proposed method ($\alpha = \infty$)	proposed method ($\alpha = 1$)
7,117	5,747

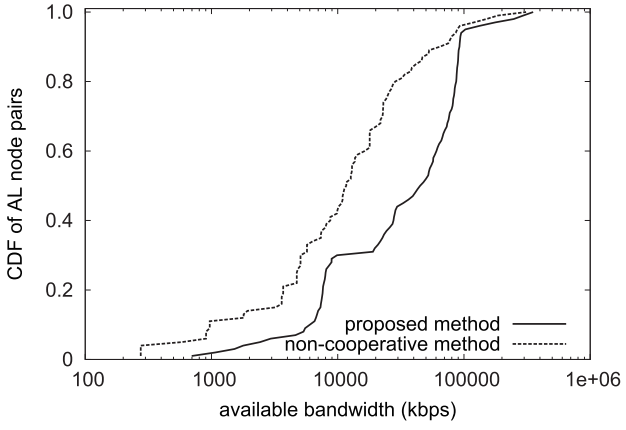

Fig. 4 Distribution of available bandwidth between the AL node pairs.

5.2.2 Dynamic Algorithm

Figures 5 and 6 show the average end-to-end latency and the

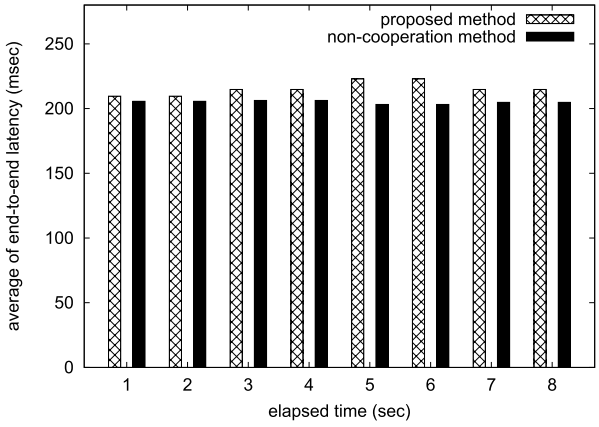

Fig. 5 Average end-to-end latency over time.

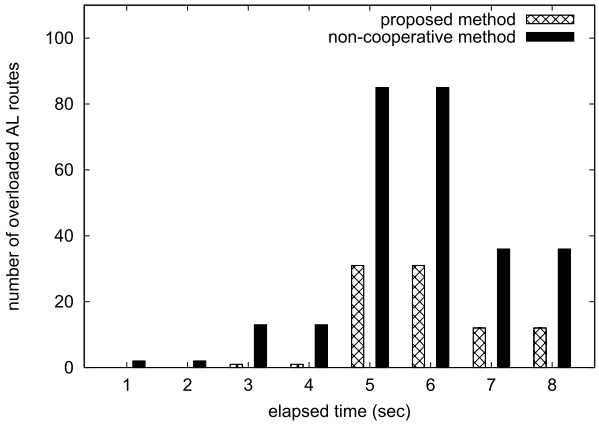

Fig. 6 Number of overloaded AL routes over time.

Table 6 Samples of changes in selected AL routes.

time and traffic value	proposed method ($\alpha = \infty$)		non-cooperative method	
2 s 300 kbps	planetlab1.di.unito.it ricepl-1.cs.rice.edu	(1, 0.81)	planetlab1.di.unito.it ricepl-1.cs.rice.edu	(2, 0.84)
3 s 1600 kbps	planetlab1.di.unito.it deimos.cecalc.ula.ve ricepl-1.cs.rice.edu	(1, 0.65) (2, 0.12)	planetlab1.di.unito.it planetlab2.cs.columbia.edu ricepl-1.cs.rice.edu	(3, 1.39) (2, 0.08)

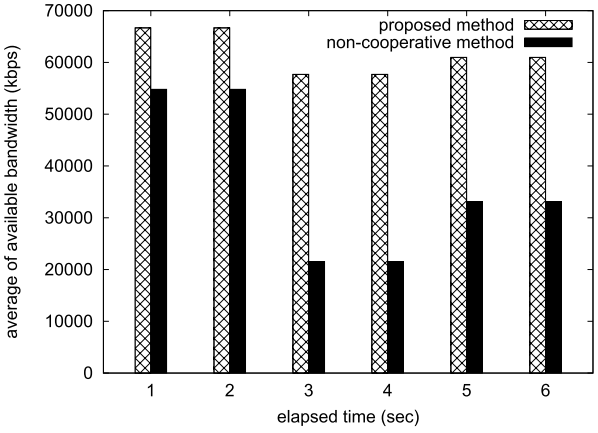

Fig. 7 Average available bandwidth over time.

number of overloaded AL routes as a function of simulation time, where end-to-end latency is taken as a routing metric. The trend in the results at each second in Fig. 5 is similar to that in Table 2, where the end-to-end latency achieved by the proposed method was slightly larger than that achieved by the non-cooperative method. On the other hand, from Fig. 6, the number of overloaded AL routes was significantly affected by the AL traffic demand changes. Between 2 sec and 3 sec, and between 4 sec and 5 sec, the number of overloaded AL routes increased in both methods. However, when comparing two methods, the proposed method gave a significantly smaller increase in the number of overloaded AL routes. Specifically, the proposed method reduced the overloaded AL routes by roughly 65% from that by the non-cooperative method at all times.

Table 6 shows samples of changes in selected AL routes between 2 sec and 3 sec in the same manner as Table 3. For the AL route between *planetlab1.di.unito.it* and *ricepl-1.cs.rice.edu*, the direct route selected by both methods at 2 sec had become overloaded at 3 sec, so both methods tried to change the AL route. However, the non-cooperative method could not avoid overlaps, which caused the overloaded AL routes. The proposed method selected the AL route where the number of overlaps was small and the bandwidth utilization was low. From these results, we can confirm that the proposed method can select AL routes while reacting to AL traffic demand changes.

Figure 7 shows the changes in the available bandwidth, presented in the same manner as Fig. 5, when available bandwidth is utilized as a routing metric. The trend in the results at each second is similar to that for the static algorithm. The proposed method with the dynamic algorithm could achieve more bandwidth than the non-cooperative method, regardless of the changes in the number of AL node pairs that required the bandwidth. This is because the proposed method shares information about which AL node pairs require the bandwidth at each time, and changes the AL routes taking account of the sharing the bandwidth among these AL node pairs.

From these results, we confirm that the proposed method with the dynamic algorithm can react to changes in AL traffic demand, while achieving almost the same effectiveness as the static algorithm.

6. Conclusion

In this paper, we proposed an ALrouting method that works in a coordinated manner based on DSA. First, we formulated the AL routing and defined an optimization problem for selecting AL routes. Second, we proposed an AL routing method based on DSA with two algorithms, one for static AL traffic demand and the other for dynamic changes in AL traffic demand. Assuming that PlanetLab nodes perform AL routing, we confirmed that the proposed algorithms could avoid overlaps and overload on AL links, which resulted in reduced congestion and less performance degradation on the AL routes.

In recent years, some extremely large content providers called *hyper giants* have emerged. They are likely to construct direct peering relationships to a number of edge ISPs that provide access service to end users, utilizing Internet exchanges to reduce the transit cost. The utilization of Internet exchanges by the edge ISPs facilitates the peering contracts between the edge ISPs. The proposed method can exploit peering links to improve user-perceived performance and reduce transit cost. Therefore, as the peering links between the edge ISPs increase in the future, the proposed method will become more efficient.

In the future, we plan to combine two routing metrics, namely, end-to-end latency and available bandwidth, to achieve both effectiveness in avoiding congestion and significant improvement of available bandwidth. We will also try to realize an application scenario where the proposed method is implemented by many ISPs by constructing a policy for exchanging the information about IP network performance and an architecture to balance ISP profits.

Acknowledgments

This work is supported in part by National Institute of Information and Communications Technology (NICT), Japan.

References

[1] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," Proc. INFOCOM 2003, April 2003.
[2] D.G. Andersen, A.C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," Proc. IMC 2003, Oct. 2003.
[3] C.L.T. Man, G. Hasegawa, and M. Murata, "Monitoring overlay path bandwidth using an inline measurement technique," IARIA International Journal on Advances in Systems and Measurements, vol.1, no.1, pp.50–60, Feb. 2008.
[4] Y. Zhu, C. Dovrolis, and M. Ammar, "Dynamic overlay routing based on available bandwidth estimation: A simulation study," Computer Networks Journal, vol.50, pp.739–876, April 2006.
[5] S. Seetharaman and M. Ammar, "Exit policy violations in multi-hop overlay routes: Analysis and mitigation," Proc. GLOBECOM 2007, pp.87–92, Nov. 2007.
[6] R. Keralapura, N. Taft, C. nee Chuah, and G. Iannaccone, "Can ISPs take the heat from overlay networks," Proc. HotNets-III Workshop, Nov. 2004.
[7] K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "Performance evaluation of a method to reduce inter-ISP transit cost caused by overlay routing," Proc. NETWORKS 2010, pp.250–255, Sept. 2010.
[8] G. Hasegawa, Y. Hiraoka, and M. Murata, "Effectiveness of overlay routing based on delay and bandwidth information," IEICE Trans. Commun., vol.E92-B, no.4, pp.1222–1232, April 2009.
[9] M. Arshad and M.C. Silaghi, "Distributed simulated annealing and comparison to DSA," Proc. Fourth Workshop on DCR, Aug. 2003.
[10] K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "Centralized and distributed heuristic algorithms for application-level traffic routing," Proc. ICOIN 2012, Feb. 2012.
[11] Y. Chen, D. Bindel, H. Song, and R.H. Katz, "An algebraic approach to practical and scalable overlay network monitoring," SIGCOMM Comput. Commun. Rev., vol.34, no.4, pp.55–66, Aug. 2004.
[12] N. Hu and P. Steenkiste, "Exploiting Internet route sharing for large scale available bandwidth estimation," Proc. IMC 2005, pp.187–192, Oct. 2005.
[13] J. Hromkovic, Algorighmics for Hard Problems, Springer, 2005.
[14] Hewlett-Packard Laboratories Scalable Sensing Service, available at http://networking.hpl.hp.com/s-cube/
[15] University of Oregon Route Views Project, available at http://www.routeviews.org/
[16] University of California CAIDA, available at http://www.caida.org/home/
[17] X.R. Wu and A.A. Chien, "A distributed algorithm for max-min bandwidth sharing," Tech. Rep., University of California, San Diego, 2006.
[18] M. Pustisek, I. Humar, and J. Bester, "Empirical analysis and modeling of peer-to-peer traffic flows," Proc. MELECON2008, pp.169–175, May 2008.

Kazuhito Matsuda received an M.E. degree in Information Science and Technology in 2010 from Osaka University, Japan, where he is now a doctoral student. His research work is in the area of overlay networks especially from the aspect of monetary cost.

Go Hasegawa received the M.E. and D.E. degrees in Information and Computer Sciences from Osaka University, Japan, in 1997 and 2000, respectively. From July 1997 to June 2000, he was a Research Assistant of Graduate School of Economics, Osaka University. He is now an Associate Professor of Cybermedia Center, Osaka University. His research work is in the area of transport architecture for future high-speed networks and overlay networks. He is a member of IEEE and IPSJ.

Masayuki Murata received the M.E. and D.E. degrees in Information and Computer Science from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984, he joined Tokyo Research Laboratory, IBM Japan, as a Researcher. From September 1987 to January 1989, he was an Assistant Professor with Computation Center, Osaka University. In February 1989, he moved to the Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University. In April 1999, he became a Professor of Cybermedia Center, Osaka University, and is now with Graduate School of Information Science and Technology, Osaka University since April 2004. He has more than five hundred papers of international and domestic journals and conferences. His research interests include computer communication network architecture, performance modeling and evaluation. He is a member of IEEE and ACM. He is a chair of IEEE COMSOC Japan Chapter since 2009. Also, he is now partly working at NICT (National Institute of Information and Communications Technology) as Deputy of New-Generation Network R&D Strategic Headquarters.