Centralized and distributed heuristic algorithms for application-level traffic routing

Kazuhito MATSUDA*, Go HASEGAWA*, Satoshi KAMEI† and Masayuki MURATA*
*Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
Email: {k-matuda, hasegawa, murata}@ist.osaka-u.ac.jp
†NTT Service Integration Laboratories, 3-9-11 Midori-cho, Musashino, Tokyo 180-8585, Japan
Email: kamei.satoshi@lab.ntt.co.jp

***Abstract*—Recent studies have revealed that the adoption of application-level routing that chooses an end-to-end traffic route relaying other end-hosts can improve certain user-perceived performance factors. However, selfish route selection performed by each application user can lead to a decrease in the route performance due to route overlaps, as well as an increase in the inter-ISP transit cost as a result of utilizing more transit links than in the case of IP routing. In the present paper, we strictly define an optimization problem for selecting application-level traffic routes with the aim to improve end-to-end network performance and reduce transit cost. We then propose centralized and distributed heuristic algorithms based on simulated annealing in order to obtain near-optimal solutions to the problem. We evaluate the performance of the proposed method by assuming that the PlanetLab nodes utilize application-level traffic routing. We show that application-level traffic routing with the proposed algorithms can result in a considerable improvement of network performance. In particular, in the case of using the available bandwidth as the routing metric, the end-to-end network performance can be improved by 87% on average.**

***Index Terms*—network optimization, overlay network, routing, inter-ISP transit cost, simulated annealing**

I. Introduction

Application-level traffic routing where end-to-end underlayer routes are regarded as application-level virtual links, as shown in Fig. 1, has recently received much attention. Recent studies have revealed that such routing can improve user-perceived performance factors such as end-to-end latency, available bandwidth and packet loss ratio [1-4]. However, selfish route selection performed by each application user to improve their own performance can lead to a decrease in the route performance due to route overlaps. For example, in [5], a number of overlay networks that are not operated in a coordinated manner cause oscillations in route selection due to the concentration of traffic at certain links. Furthermore, route selection aimed at improving only user-perceived performance 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 monetary cost as *transit cost* in this paper) [6]. In the rest of this paper, we refer to end-hosts, which can be senders, relay hosts and destinations, as AL nodes.

In [7], we demonstrated that the number of transit links traversed by user traffic routed by application-level traffic routing could be controlled by using end-to-end network performance values as routing metrics. The results showed the possibility of reducing the transit cost generated by application-level traffic routing. However, in [7], we did not evaluate the influence of route overlaps.

When end users utilize application-level routing in a coordinated manner, the route selection process can be controlled considering the routes utilized by other AL nodes. In general, there are two candidate of coordinated manner for routing. One is a centralized algorithm where one supernode gathers all information needed for route calculation and communicates the appropriate route to each AL node; the other method is a distributed algorithm where each AL node calculates each route used by that node. Both of them have advantages and disadvantages, and the more appropriate one of them should be chosen according to the target situation.

In this paper, with the aim to improve end-to-end network performance, we focus on application-level traffic routing operated in a coordinated manner by all AL nodes. First, we formulate the application-level traffic routing and strictly define an optimization problem for selecting application-level traffic routes with various route selection metrics. We then propose centralized and distributed heuristic algorithms that produce near-optimal solutions for the optimization problem. We propose two different algorithms to ensure that the requirements in a wide range of situations are met. For example, in the case of the centralized algorithm, we can imagine a situation where an Application Service Provider (ASP) controlling a supernode communicates the selected routes to end users. On the other hand, in the case of the distributed algorithm, we can imagine a situation where AL nodes are controlled independently by ISPs and the routes are communicated to each ISPs' customers. We evaluate the proposed algorithms by assuming that the PlanetLab nodes utilize application-level routing by using the end-to-end measurement results of the network performance value. First, we evaluate the differences between the network performance achieved with the optimal solution and that achieved with the proposed centralized algorithm to confirm the effectiveness of the proposed heuristic algorithms. After that, we show the evaluation results for the network performance and confirm that the performance can be substantially improved by application-level routing with the proposed algorithms.

II. Application-level route optimization problem

In this section, we begin by explaining the network model assumed in this paper. We then formulate the application-level routing and define the optimization problem for selecting application-level routes.

A. Network model

We assume a network model as depicted in Fig. 2. The underlay IP network is constructed from a number of IP-level routers, each of which is located at one of the AS. 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 links or

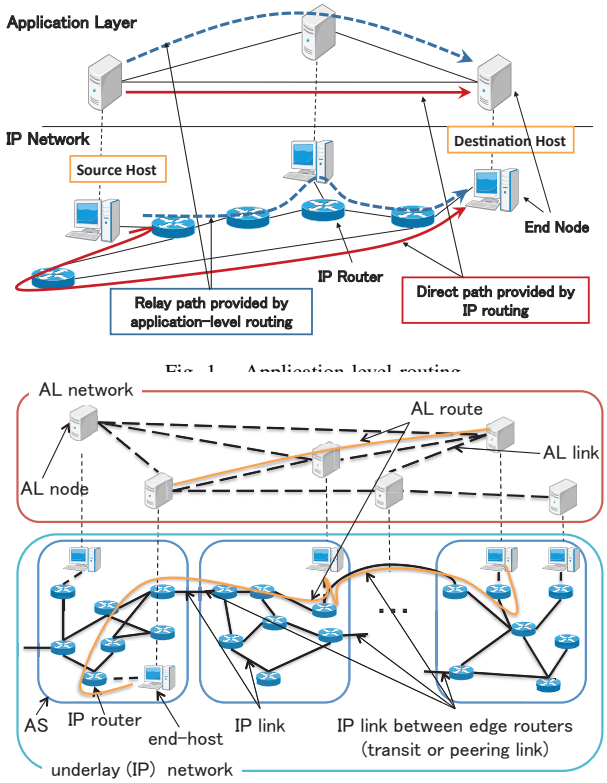


Fig. 2. Network model

peering links. Note that a transit cost is incurred when traffic traverses transit links. We ignore the hierarchical AS-level topology since we focus on the total amount of the transit cost for all the traffic in the network. Application-level (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. AL routing is performed on the AL network and determines the AL network-level routes (AL routes) between AL node pairs that demand traffic.

B. Optimization problem

First, 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 from 1 to M to each link.

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

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

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 the 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.

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 this 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 in this paper we assume that the AL routing determines the AL routes only for AL node pairs that demand traffic. Here, we define the AL routing matrix as follows.

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

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 the value of the element becomes zero if node pair j does not demand traffic.

We divide traffic traversing the entire network into two parts: traffic only carried by IP routing and traffic carried by AL routing. We describe the traffic demands 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 demands 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 loads on the links between routers, as follows.

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

We introduce a function f_D, which calculates the latencies of all AL links by using 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.

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

For the available bandwidth, we define a function f_B, which calculates the available bandwidths for all AL routes

by using Y and A^{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.

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

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 latency.

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

As mentioned above, the AL routing determines AL routes only for AL node pairs that demand traffic. When we describe the set of identifiers of AL node pairs that demand traffic as Θ, the problem of minimizing the average of the latencies of the AL routes between AL nodes that demand traffic is described as follows, where the AL routes between AL nodes $r_j (j \in \Theta)$ are treated as variables.

$$\begin{aligned} &\text{minimize :} && (\textstyle\sum_{j\in\Theta} d_j^{\mathrm{AL}})/|\Theta| \\ &\text{subject to :} && r_j \in \Gamma_j^{\mathrm{AL}} (\forall j | j \in \Theta) \end{aligned} \tag{9}$$

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

$$\begin{aligned} &\text{maximize :} && (\textstyle\sum_{j\in\Theta} b_j^{\mathrm{AL}})/|\Theta| \\ &\text{subject to :} && r_j \in \Gamma_j^{\mathrm{AL}} (\forall j | j \in \Theta) \end{aligned} \tag{10}$$

Similarly, the minimization problem for the transit cost can be described as follows. Note that the targets of this problem are not the AL routes but all the AL links.

$$\begin{aligned} &\text{minimize :} && (\textstyle\sum_{j=1}^{(N-1)N} c^{\mathrm{AL}_j})/((N-1)N) \\ &\text{subject to :} && r_j \in \Gamma_j^{\mathrm{AL}} (\forall j | j \in \Theta) \end{aligned} \tag{11}$$

III. Proposed method

In this section, we propose centralized and distributed heuristic algorithms for obtaining near-optimal solutions to the problem described in Section II. For this purpose, we utilize a popular heuristic algorithm known as *simulated annealing* (SA).

A. Centralized algorithm

We show the centralized algorithm utilizing SA in Algorithm 1, and the definitions of symbols and functions used in the algorithm are shown in Table I. The process of SA progresses as it decides whether to change the current state to its neighbor state. For the proposed algorithm, the *state* is the set of AL routes selected for all AL node pairs that demand traffic.

The parameters required for Algorithm 1 are the initial state S_{init}, the neighbor state generation function $\mathrm{Neighbor}()$, the cost function $\mathrm{Cost}()$, the transition probability function $\mathrm{Probability}()$, the initial temperature T_{init} and the cooling schedule function $\mathrm{Cooling}()$. We explain these parameters in detail below.

Initial state

TABLE I
NOTATIONS FOR CENTRALIZED AND DISTRIBUTED SA ALGORITHMS

I	iteration count of simulated annealing
T	temperature of simulated annealing
S	state of simulated annealing
$\mathrm{Neighbor}(S)$	function that returns a neighbor state of S
$\mathrm{Cost}(S)$	function that returns the cost of S
$\mathrm{Random}(x, y)$	function that returns a value between x and y
$\mathrm{Probability}(T, S, S_{tmp})$	function that returns the probability of transition in the direction of reducing the cost
$\mathrm{Cooling}(T)$	function that returns the temperature of the subsequent iteration
$\mathrm{Update}(S)$	function that updates the state with information about other AL nodes received until the present time
$\mathrm{Neighbor}_{\mathrm{dsa_i}}(S)$	function that returns a neighbor state of S, changing only the AL routes related to AL node i
$\mathrm{SendNeighbor(S)}$	function that sends the state S to neighbor AL nodes

Algorithm 1 Centralized algorithm

```
I ← 0,   T ← T_init,   , S ← S_init
while T > 0 do
  S_tmp ← Neighbor(S)
  if Cost(S) ≥ Cost(S_tmp) then
    S ← S_tmp
  else
    r ← Random(0, 1)
    if r < Probability(T, Cost(S), Cost(S_tmp)) then
      S ← S_tmp
    end if
  end if
  I ← I + 1
  T ← Cooling(T, I)
end while
```

In the initial state, all AL node pairs that demand traffic select routes identical to that without AL routing; the selected routes correspond to one-hop routes in the AL network, because we believe the network performance of the routes to be identical to the case without AL routing, which is the network performance baseline for AL node pairs.

Neighbor state generation function

A *neighbor state* of state S is defined to be where some of the AL routes are changed, and as a result the function generates a state that is slightly different from S for SA process.

Cost function

We use Eqs. (9)–(11) for the cost function.

Transition probability function

We utilize the general equation used for SA. The equation is described as follows.

$$\mathrm{Probability}(T, S, S_{tmp}) = e^{-\frac{\mathrm{Cost}(S_{tmp}) - \mathrm{Cost}(S)}{T}} \tag{12}$$

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

Initial temperature and cooling schedule function

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 [8]. The actual value of the initial temperature should be set according to the target situation. We use the following general cooling schedule function for SA.

$$\mathrm{Cooling}(T, I) = \gamma T \ (0 < \gamma < 1) \tag{13}$$

Algorithm 2 Distributed algorithm for AL node i

$I_i \leftarrow 0, \quad T_i \leftarrow T_{i_{init}}, \quad S_i \leftarrow S_{i_{init}}$
while $T_i > 0$ **do**
  $\mathrm{Update(S_i)}$
  $S_{i_{tmp}} \leftarrow \mathrm{Neighbor_{dsa_i}}(S_i)$
  if $\mathrm{Cost}(S_i) \geq \mathrm{Cost}(S_{i_{tmp}})$ **then**
    $S_i \leftarrow S_{i_{tmp}}$
  else
    $r_i \leftarrow \mathrm{Random}(0, 1)$
    if $r_i < \mathrm{Probability}(T_i, \mathrm{Cost}(S_i), \mathrm{Cost}(S_{i_{tmp}}))$ **then**
      $S_i \leftarrow S_{i_{tmp}}$
    end if
  end if
  $I_i \leftarrow I_i + 1$
  $T_i \leftarrow \mathrm{Cooling}(T_i, I_i)$
  if S_i has been updated **then**
    $\mathrm{SendNeighbor}(S_i)$
  end if
end while

B. Distributed algorithm

We present the distributed algorithm in Algorithm 2, which is based on the algorithm proposed in [9]. The symbols in Algorithm 2 are the same as those in Algorithm 1, although we attach a subscript i indicating that the algorithm is run on AL node i. Each AL node runs the algorithm independently, where the term "independently" signifies that each AL node measures the network performance values only for AL links related to itself, and each AL node can perform state transition only with respect to AL routes related to itself. Due to these features of the distributed algorithm, each AL node must gather the network performance values and the AL routes determined by the other AL nodes. In addition, we must transform the neighbor generation function into a form that is appropriate to the distributed algorithm. Although the distributed algorithm enables each AL node to determine the AL routes related to its own self, the AL nodes must gather information related to the other AL nodes, and doing so generates communication overhead. However, decreasing the frequency of gathering information related to the other AL nodes in order to reduce the communication overhead leads to an increase in the gap between the actual current state and the current state used by each AL node. In other words, the distributed algorithm has a trade-off between communication overhead and accuracy of the current state for SA.

Based on these features of the distributed algorithm, the parameters required for Algorithm 2 are the update function $\mathrm{Update}()$, which updates the state according to information about the other AL nodes, the notification function $\mathrm{SendNeighbor}()$, which communicates the AL route to the other AL nodes, and the neighbor state generation function $\mathrm{Neighbor_{dsa_i}}()$, which is appropriately modified to the distributed algorithm.

Update function

The update function gathers the network performance values and the AL routes determined by the other AL nodes, after which it updates the current state of the own node with the gathered information.

Notification function

The notification function sends information about the AL route related to own AL node to the other AL nodes. In this paper, the AL routes related an AL node indicate the AL routes containing that AL node as the source node. Although the cost function requires the current state of all AL nodes to calculate the cost accurately, the communication overhead becomes high if the AL routes are gathered from the other AL nodes on each update of the state of each AL node. For this reason, the frequency of notification of AL routes to other AL nodes should be determined by considering the communication overhead due to notification.

Modified function for neighbor state generation

The modified function for neighbor state generation targets only the AL routes related to the own AL node, unlike the case of the centralized algorithm, where the neighbor generation function targets all AL routes.

IV. Evaluation

In this section, we show the results of evaluating the proposed algorithms described in Section III by assuming that the PlanetLab nodes constitute an AL network and conduct AL routing.

A. Dataset and settings

1) Dataset: For the evaluation, we used the results of measuring 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 and used the `traceroute` 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) [10]. S^3 provides a summary of the measurement results among PlanetLab nodes every 4 hours. In this paper, we used the measurement results obtained on October 19, 2010.

AS-level routes

We converted the IP-level routes to AS-level routes by using the relationships between IP address prefixes and AS numbers, which are made available at the Route Views Project [11]. We used the data obtained on April 16, 2009. Although the data of AS numbers is rather older than other data, we think it does not affect the evaluation results because attached AS numbers are not changed frequently.

Information about the relationships between ASes

We utilized the information about the relationships between ASes as provided by CAIDA [12] on January 20, 2010.

2) Settings for cost functions: In the evaluation, we determined the functions f_D, f_B and f_C in Eqs. (6)–(8) as follows. The function f_D calculates the sum of propagation delays and queuing delays at the routers to obtain the end-to-end latency. We adopted the M/M/1 queuing model in calculating the queuing delays with the following assumptions.

- None of the AL links share any IP links.
- 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.
- The queuing delay at an AL link occurs mainly at the tight IP link, and queuing delays at other IP links are negligible.

With the above assumptions, we calculated the queuing delay of the AL link between AL node pair j, d_j^q, as follows.

$$d_j^q = \frac{\frac{c_j - a_j + x_j}{c_j}}{1 - \frac{c_j - a_j + x_j}{c_j}} \times \frac{P}{c_j} \quad (14)$$

where c_j, a_j and x_j are the physical capacity, the available bandwidth and the traffic demand of the AL link between AL node pair j, respectively. Here, P is the average packet size. The end-to-end latency of the AL link between AL node pair j, d_j, was calculated as follows by using d_j^q, which is obtained from Eq. (14), and the measured propagation delay d_j^p.

$$d_j = d_j^q + d_j^p \quad (15)$$

The function f_B determines the available bandwidths of AL links. We utilized a simple max-min discipline to calculate the available bandwidths. In other words, when we calculated the following values $f_{\mathcal{B}}(i)$ $(0 \leq i \leq (N-1)N)$ for the AL links, the AL routes shared the available bandwidths of the AL links in ascending order of the values of the AL links.

$$f_{\mathcal{B}}(i) = (a_i - \sum\nolimits_{j \in S^i} b_j^{\mathrm{AL}}) / |\{k | b_k^{\mathrm{AL}} = 0, k \in S^i\}| \quad (16)$$

where a_i represents the available bandwidth of the AL link between AL node pair j and S^i represents the set of node pairs that utilize the AL link between AL node pair i.

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

$$c_j^{\mathrm{AL}} = \beta_j x_j \quad (17)$$

where x_j represents the traffic demand on the AL link between AL node pair j. Here, β_j determines the transit cost per unit amount of traffic, which is calculated as the sum of transit costs of all IP links traversed by the AL link between AL node pair j. In the evaluation, we used the following function to determine the transit cost of IP link i on the AL link between AL node pair j, v_i^j.

$$v_i^j = \begin{cases} 1 & (IP_i^j = 1 \text{ and } i \text{ is a transit link}) \\ 0.05 & (IP_i^j = 1 \text{ and } i \text{ is a peering link}) \\ 0 & (IP_i^j = 1 \text{ and } i \text{ is not a AS-level link}) \\ 0 & (IP_i^j = 0) \end{cases} \quad (18)$$

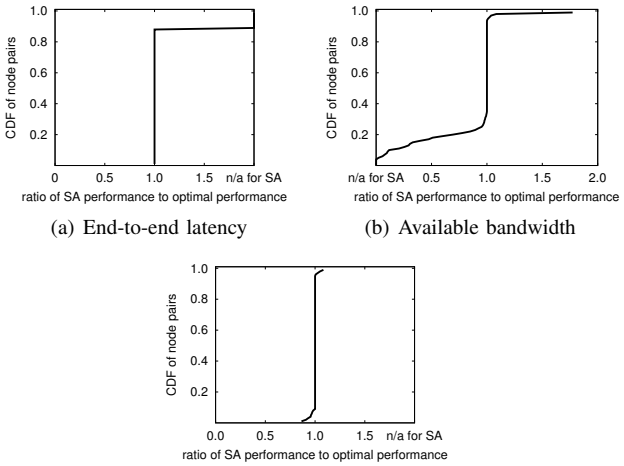
The value of β_j was calculated from Eq. (19) as follows.

$$\beta_j = \sum\nolimits_{i=1}^{M} v_i^j \quad (19)$$

Note that in cases where we were unable to obtain the measurement results of the network performance values and thus could not calculate the end-to-end latencies and the available bandwidths of AL links with the methods described above, we regarded the end-to-end latencies as infinite and the available bandwidths as zero. As a result, we did not use those AL links in AL routing.

3) Other settings: In the proposed algorithms, we normalized the state cost by the initial state cost. The proposed algorithms completed when the temperature became lower than 10^{-6}. The initial temperature was 0.15, and the parameter γ for the cooling schedule function was 0.999. The traffic demand on the AL node pairs, which were used to calculate end-to-end latency and transit cost, was 1000 kbps. Also, the average packet size used in Eq. (14) was 770 bytes, which is the average value calculated with the general maximal packet size of 1500 bytes and the TCP ACK packet size of 40 bytes. In addition, we considered only one-hop and two-hop AL routes because AL routes with more than two AL links do not contribute to the improvement of end-to-end network performance [13]. The neighbor generation function randomly selected AL routes from among the one-hop and two-hop AL route candidates for 1% of the AL node pairs demanding traffic.

For the distributed algorithm, we assumed that all AL nodes exchanged all of the measured network performance values and all information about AL routes every 100 iterations of the SA process.



(a) End-to-end latency (b) Available bandwidth

(c) Transit cost

Fig. 3. Comparison with optimal solutions

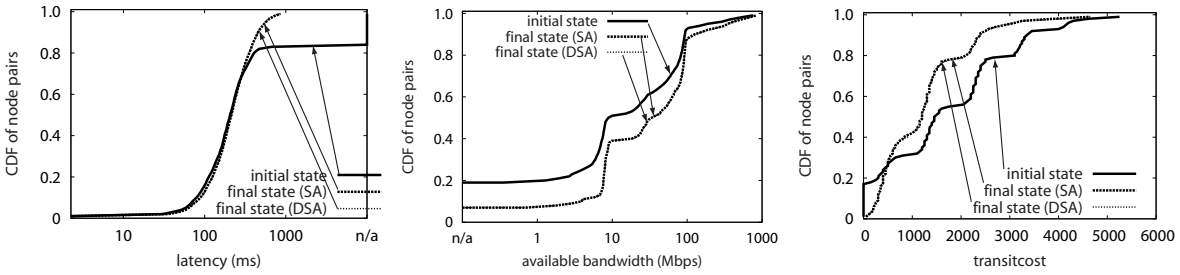
B. Comparison with optimal solutions

First, we evaluated the differences between the end-to-end network performance achieved with the optimal solution and that achieved with the proposed centralized algorithm. Here, the optimal solution was obtained by exhaustive search. We randomly chose 10 PlanetLab nodes as AL nodes and assumed that 7 AL node pairs demanded traffic. We selected 10 AL nodes and 10 AL node pairs, after which we performed 100 evaluation trials for each network performance metric. For the evaluation metric, we utilized the ratio of the end-to-end network performance achieved with the proposed algorithm to that achieved with the optimal solution, I, as follows.

$$I = \delta_{sa} / \delta_{opt} \quad (20)$$

where δ_{sq} and δ_{opt} represent the end-to-end network performance achieved with the proposed algorithm and that achieved with the optimal solution, respectively.

Fig. 3 shows the distribution of the values of I. The results for the end-to-end latency, the available bandwidth and the transit cost utilized as routing metric are shown in Figs. 3(a), 3(b) and 3(c), respectively. Note that in the proposed algorithm, a portion of the AL node pairs exhibit performance higher than that obtained with the optimal solution since the proposed algorithm utilizes the object functions, which minimize/maximize the average value of the metrics. Fig. 3 shows that the end-to-end network performance achieved with the optimal solution and that achieved with the proposed algorithm are almost the same in more than 70% of the AL



(a) End-to-end latency (b) Available bandwidth (c) Transit cost

Fig. 4. End-to-end network performance achieved by the proposed algorithms

node pairs. From this result, we can conclude that the proposed algorithm can produce near-optimal solutions.

C. Performance evaluation for the centralized and distributed algorithms

Next, we assessed the end-to-end performance improvement achieved with the proposed algorithms. We randomly chose 30 PlanetLab nodes as AL nodes and assumed that 10% of the AL node pairs demanded traffic. We conducted 100 trials, as in the previous subsection.

Fig. 4 shows the distribution of the end-to-end network performance of the initial state and final states for the centralized and the distributed algorithm. The results for the end-to-end latency, the available bandwidth and the transit cost utilized as routing metric are shown in Figs. 4(a), 4(b), and 4(c), respectively. Note that the initial states for both algorithms were identical. Below, we explain the results for the centralized algorithm and the distributed algorithm, in this order.

1) Centralized algorithm: From Fig. 4(a), AL routing results in almost no improvement in the end-to-end latency, which has been pointed out in the literature [7, 13]. In contrast, for the available bandwidth (Fig. 4(b)), a substantial improvement by 84% on average was achieved, which mirrors the tendency in [7, 13]. Also, the transit cost (Fig. 4(c)) was reduced by 27% on average, which clearly shows that the transit cost can be reduced with the centralized algorithm, even though the AL routing generally increases the number of traversed IP links by detouring.

2) Distributed algorithm: We focus on a comparison between the centralized and the distributed algorithm. From the results shown in Fig. 4, the distributed algorithm achieves almost the same performance as the centralized algorithm. This indicates that the distributed algorithm, which changes the AL routes independently on each AL node, can produce almost the same results as the centralized algorithm.

V. Conclusion

In this paper, we proposed centralized and distributed heuristic algorithms for application-level traffic routing. First, we formulated the application-level routing and defined an optimization problem for selecting AL routes, after which we introduced centralized and distributed heuristic algorithms based on simulated annealing. Assuming that PlanetLab nodes perform AL routing, we confirmed that the proposed algorithms could achieve near-optimal solutions as well as considerable improvement in end-to-end network performance. Furthermore, we showed that the distributed algorithm could provide almost the same results as the centralized algorithm.

In future research, we will evaluate the proposed method with more than one metrics such as minimizing end-to-end latency under a constraint on transit cost. In addition, we intend to evaluate the distributed algorithm in the case of limited information exchange, which can occur, for instance, when ISPs operate AL nodes and each AL node runs the distributed algorithm independently. In such situations, information exchange might be limited due to control exerted by each ISP. We will also aim to extend the proposed algorithms appropriately to the protocol developed by Application-Layer Traffic Optimization (ALTO), which is an architecture where ISPs control the application user traffic by communicating network information through a server-client model. We believe that the proposed algorithms can achieve high compatibility with the ALTO protocol and can realize traffic optimization across a number of ISPs which are not considered in ALTO at present.

Acknowledgments

This work is supported in part by the Ministry of Internal Affairs and Communications (MIC), Japan, under the Promotion program for Reducing global Environmental loaD through ICT innovation (PREDICT).

References

[1] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *Proceedings of INFOCOM 2003*, Apr. 2003.
[2] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *Proceedings of 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, 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, Apr. 2006.
[5] R. Keralapura, N. Taft, C. nee Chuah, and G. Iannaccone, "Can ISPs take the heat from overlay networks," in *Proceedings of HotNets-III Workshop*, Nov. 2004.
[6] S. Seetharaman and M. Ammar, "Exit policy violations in multi-hop overlay routes: Analysis and mitigation," in *Proceedings of GLOBECOM 2007*, pp. 87–92, Nov. 2007.
[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," in *NETWORKS 2010*, pp. 250–255, Sept. 2010.
[8] J. Hromkovic, *Algorighmics for Hard Problems*. Springer, 2005.
[9] M. Arshad and M. C. Silaghi, "Distributed simulated annealing and comparison to DSA," in *Proceedings of the Fourth Workshop on DCR*, Aug. 2003.
[10] Hewlett-Packard Laboratories Scalable Sensing Service. available at http://networking.hpl.hp.com/s-cube/.
[11] University of Oregon Route Views Project. available at http://www.routeviews.org/.
[12] University of California CAIDA. available at http://www.caida.org/home/.
[13] G. Hasegawa, Y. Hiraoka, and M. Murata, "Effectiveness of overlay routing based on delay and bandwidth information," *IEICE Transactions on Communications*, vol. E92-B, pp. 1222–1232, Apr. 2009.