Reducing ISPs' cost
by application-level path selection
and in-network caching

Kazuhito MATSUDA

Graduate School of Information Science and Technology
Osaka University

January 2013

List of Publications

Journal Papers

1. K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "A method to reduce inter-ISP transit cost caused by overlay routing based on end-to-end network measurement," to appear in *IEICE Transactions on Information and Systems*, vol. E96-D, no. 2, Feb. 2013.

Refereed Conference Papers

1. K. Matsuda, G. Hasegawa, and M. Murata, "Decreasing ISP transit cost in overlay routing based on multiple regression analysis," in *Proceedings of International Conference on Information Networking (ICOIN) 2010*, Jan. 2010.

2. 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 *14th International Telecommunications Network Strategy and Planning Symposium (NETWORKS 2010)*, pp. 250–255, Sept. 2010.

3. K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "Centralized and distributed heuristic algorithms for application-level traffic routing," in *Proceedings of International Conference on Information Networking (ICOIN) 2012*, Feb. 2012.

Non-Refereed Technical Papers

1. K. Matsuda, G. Hasegawa, and M. Murata, "Decreasing ISP transit cost in overlay routing based on multiple regression analysis," *Technical Report of IEICE* (ICM2009-25), vol. 109, no. 120, pp. 67–72, July 2009. (in Japanese)

2. K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "Reducing inter-ISP transit cost caused by overlay routing," *Technical Report of IEICE* (NS2010-17), vol. 110, no. 39, pp. 7–12, May 2010. (in Japanese)

3. K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "An application-level routing method for improving end-to-end network performance based on heuristic algorithm," *Technical Report of IEICE* (NS2011-65), vol. 111, no. 196, pp. 23–28, Sept. 2011. (in Japanese)

4. K. Matsuda, G. Hasegawa, and M. Murata, "A dynamic application-level routing method reacting traffic changes based on distributed heuristic algorithm," *Technical Report of IEICE* (IN2012-36), vol. 112, no. 134, pp. 19–24, July 2012. (in Japanese)

Preface

The current Internet consists of a numerous number of Internet Service Providers (ISPs), each of which operates own network to maximize its own benefit. Since each individual ISP cannot connect to whole part of the Internet directly, it connects to other ISPs with monetary contracts, so that it can provide full connectivity to the Internet for its customers. The contracts between ISPs are determined mainly by the magnitudes of ISPs. The lower-level ISP connects to the upper-level ISP to ensure full IP reachability. Such inter-ISP link is called as a *transit link*. The lower-ISP pays monetary cost to the upper-ISP according to the amount of traffic traversing the transit link, regardless of the traffic direction. Such monetary cost accounts for a greater part of ISP's expense. Therefore, to suppress the monetary cost, ISPs interconnect with other ISPs whose magnitude is comparable to themselves, by making different type contracts. A link based on such contract is called as a *peering link*. It is usually used only for the traffic within the interconnected ISPs. In general, the monetary cost is not incurred when the traffic traverses the peering link, except for the cost paid to carrier companies for the physical link facilities. Each ISP configures the IP-level routes for the network traffic according to those differences on monetary cost of inter-ISP links.

In recent years, on the other hand, new types of traffic routing mechanisms are getting much attention, which largely impact on the ISPs' cost structure. *Application-level traffic routing* is one of such mechanisms. Extensive previous researches have revealed that the application-level traffic routing can improve end-to-end network performance without any modification in the current underlay network. However, because the application-level traffic routing is conducted by end-to-end approach without any care of the monetary costs of inter-ISP links, it can be harmful to the ISPs' cost structure. *Content-centric networking* (CCN), which routes packets based on content name,

also largely affects to the ISPs' cost structure. On the contrary to application-level traffic routing, CCN brings positive effect on inter-ISP transit cost due to its in-network caching mechanism, because that it can reduce the traversing traffic on transit links by replying the cached contents when there is a cache hit. However, CCN is not developed considering ISPs cost structure directly, and in-network caching of CCN does not help reducing the transit cost of ISPs with peering relationships. In this thesis, therefore, we focus on the impact of such new traffic routing technologies on the ISPs' cost structure, and develop comfortable methods to decrease ISPs' network cost, while preserving end-to-end network performance, in respect to traffic routing.

This thesis begins by developing a method to reduce transit cost of application-level traffic routing conducted by individual end user. Since ISPs and end users have their own objectives respectively regarding traffic routing, application-level traffic routing should be operated considering both standpoints, while existing methods focused only on improving end-to-end network performance. Therefore, we propose a method to reduce inter-ISP transit cost caused by application-level traffic routing, while maintaining end-to-end network performance gain, considering both standpoints of ISPs and end users. To determine the relationships among ASes, which are required for ISP cost-aware routing, we first construct a method to estimate the transit cost of application-level paths from end-to-end network performance values. Utilizing the estimation results, we then propose a novel method that controls application-level traffic routes satisfying both objectives of ISPs and end users. Through extensive evaluations using measurement results from the actual network environments, we confirm the advantage of the proposed method, meaning that we can reduce the transit cost while preserving the merit of application-level routing for end-to-end network performance.

In the next part of this thesis, we aim to realize an application-level traffic routing conducted by individual operators in multiple ISPs in a distributed fashion. In the proposed method, we assume that the operators of application-level traffic routing cooperate with each other on their route selection, so that we can avoid performance degradation caused by route overlaps due to their selfish decisions. As preparatory to construct the proposed routing method, we first strictly define an optimization problem for selecting application-level traffic routes with the aim of maximizing end-to-end network performance under transit cost limitation. We then propose an application-level traffic routing method based on distributed simulated annealing to obtain near-optimal solutions

to the problem. We evaluate the performance of the proposed routing method by assuming that PlanetLab nodes utilize application-level traffic routing. From the results, we indicate that the proposed routing method can result in considerable improvement of network performance without increasing transit cost. In particular, in the case of using end-to-end latency as routing metric, the number of overloaded end-to-end paths can be reduced by around 65% compared that with non-coordinated methods. We also exhibit that the proposed method can react to dynamic traffic demand changes and select appropriate routes.

The third part of this thesis, we focus on the in-network caching mechanism in CCN and propose a new mechanism to reduce the transit cost by cache sharing mechanism. The in-network caching mechanism in CCN can suppress traffic volume along the route to the host that has the content. However, the memory spaces for caching at CCN routers are relatively small comparing to the amount of contents required by end users. In addition, any initial access to a content from users must use the transit link even when nearby CCN routers outside the route have the cached content. Also, the current CCN does not have the cooperation mechanism among ISPs interconnected by peering links. Therefore, we propose an architecture of cooperative cache sharing among CCN routers in multiple ISPs. It aims to improve cache hit ratio, which leads the further reduction in the inter-ISP transit cost. In the proposed architecture, the CCN routers share the memory space for content caching. A request packet for the cached contents is forwarded to the CCN router who has the content, even when it is not located on the route to the original content holder. We also extend the mechanism to accommodate multiple ISPs under peering relationships to reduce transit cost by using peering links. For that purpose, the proposed architecture considers the balance of network traffic between cooperating ISPs by controlling the memory size for cache sharing. We evaluated the proposed architecture by simulation experiments using the actual ISPs' IP-level network topologies, and showed that the inter-ISP transit cost could reduce significantly compared to the normal caching behavior in the CCN while ensuring the fairness between the ISPs.

Finally, we discuss future works with some ideas to polish up the proposed methods in this thesis.

Acknowledgments

This thesis would not have reached completion without generous supports by the professors and colleagues. First of all, I would express my great gratitude to Professor Masayuki Murata of the Graduate School of Information Science and Technology, Osaka University, for his valuable advises and schooling.

I would give my sincere appreciation to Professor Hirotaka Nakano of the Cyber Media Center, Osaka University, for his pointed advices. His approach to research has favorable influence to me extensively.

I am grateful to the members of my thesis committee, Professor Koso Murakami and Professor Teruo Higashino of the Graduate School of Information Science and Technology, Osaka University, for reviewing my thesis and offering a lot of valuable comments.

I would express my special gratitude to Associate Professor Go Hasegawa of the Cyber Media Center, Osaka University. His innumerable critical comments and earnest support for my research are invaluable for me. Again, give my gratitude to him.

I am grateful to Mr. Satoshi Kamei of NTT, for his comments from a different perspective and supports for my research.

I am thankful to all the members of Ubiquitous Network Laboratory at the Graduate School of Information Science and Technology, Osaka University, especially, Assistant Professor Yoshiaki Taniguchi and my colleague Masahumi Hashimoto. The laboratory is the precious environment for me thanks to them, which encourages me numerous times.

Finally, I give a special thanks to my parents, my grandmother, and my sister, for their support and understanding to my academic life, which enables me to apply myself to my studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

1.1.1 Hierarchical Internet structure

The current Internet consists of a numerous networks constructed by Internet Service Providers (ISPs). A network constructed by an ISP, which is composed of a number of IP routers, is called as Autonomous System (AS), and each ISP may operate more than one AS. In general, each individual ISP cannot connect to the whole Internet directly. Therefore, each ISP makes monetary contracts with other ISPs, and interconnects with each other based on the contracts. Through these inter-ISP connections, ISPs provide full IP reachability for their customers. The contracts between ISPs are determined mainly by the magnitude of ISPs, and ISPs build up a hierarchical structure that ensures scalability of the Internet showed in Figure 1.1.

The top-level ISPs of the hierarchical structure are referred as *Tier-1 ISPs*, which has the full-route information to whole part of the Internet. The other ISPs connect to more than one upper-level ISPs to achieve the connectivity to the Internet. Such inter-ISP link is called as a *transit link*. The lower-ISP pays monetary cost to the upper-ISP interconnected by the transit link according to the amount of traffic traversing the link, regardless of the traffic direction (we refer this cost just as *transit cost* in the remainder of this thesis). Such monetary cost accounts for a greater part of ISP's expense, so the lower-ISP desires to reduce traffic traversing the transit link. Therefore, to suppress

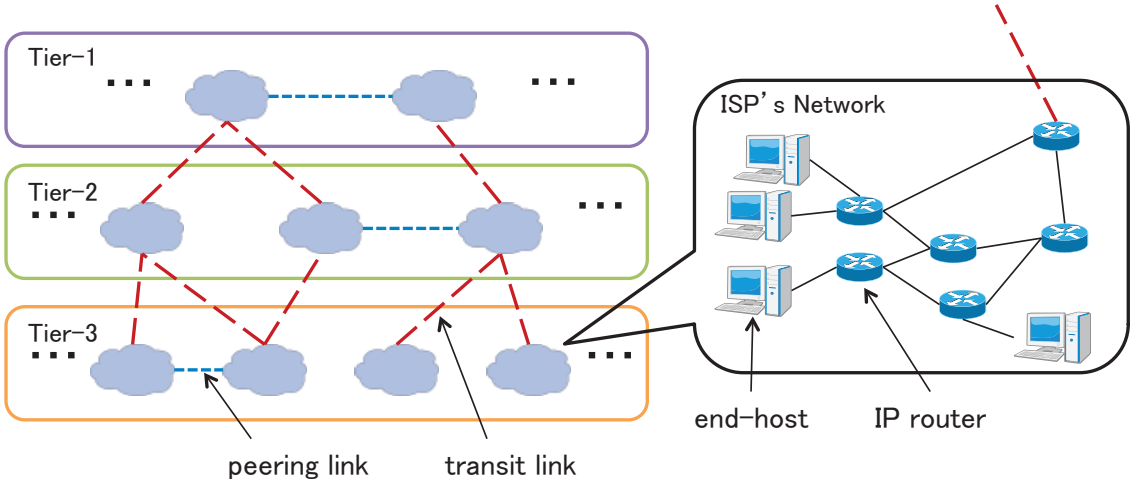
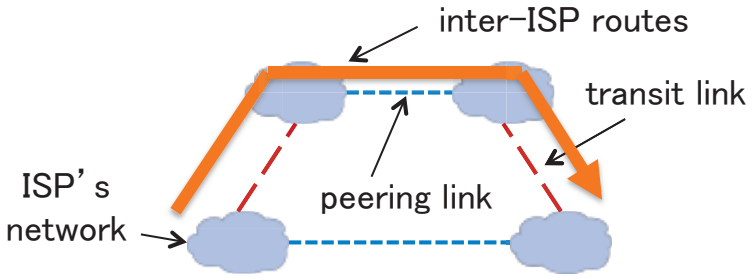


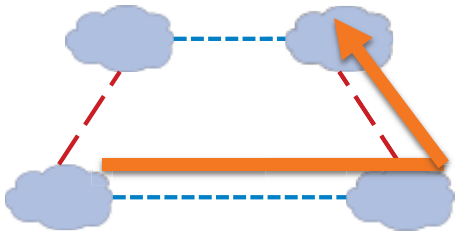
Figure 1.1: Hierarchical structure of the Internet

the monetary cost, ISPs may interconnect other ISPs whose magnitude is comparable to themselves, by making a different type contracts from the transit link. A link based on such contract is called as a *peering link*. It is usually used only for the traffic within the interconnected ISPs. The monetary cost is not incurred by traversing the peering link, except for the cost paid to carrier companies for the physical link facilities.

Each ISP configures routes for the network traffic according to those differences on monetary cost of inter-ISP links. The routes between ASes are advertised appropriate to policies on the ISPs' monetary cost structure. The ISPs uses the transit links connected to the upper-ISPs only for their own customer ISPs and end users, because the transit cost is incurred. As mentioned before, the peering links are used only for the traffic between the interconnected ISPs and their customers, that is to say the peering links are not used for the traffic to the upper-ISPs. Because of these policies come from ISPs' cost structure, the routes between ISPs have the rule termed *valley-free* [12, 13]. For example, Figure 1.2(a) is a valid route accordance with the valley-free rule. On the other hand, Figure 1.2(b) indicates an invalid route because the peering link used for the traffic to the upper-level ISP.



(a) valid route



(b) invalid route

Figure 1.2: Valley-free rule in inter-ISP routing

1.1.2 Application-level traffic routing

Application-level traffic routing is a technique for network application to provide application-level routes. It selects routes based on end-to-end network performance metrics such as end-to-end latency, available bandwidth, and TCP throughput. The route originating from the source endhost is determined for the network traffic, which can be either a route directly reaching the destination endhost or a route relaying other endhost(s) before reaching the destination as depicted in Figure 1.3. An early and typical example is the Resilient Overlay Network (RON) [14], in which each endhost measures the end-to-end latency and packet loss ratio of the network paths to other host and chooses a route according to the measurement results.

Application-level traffic routing can improve end-to-end network performance, which has revealed by extensive existing researches. The method proposed in [15] selects the application-level routes utilizing measurement results of capacity and available bandwidth to improve user-perceived performance of these metrics. In [16], the authors present the method to construct and maintain

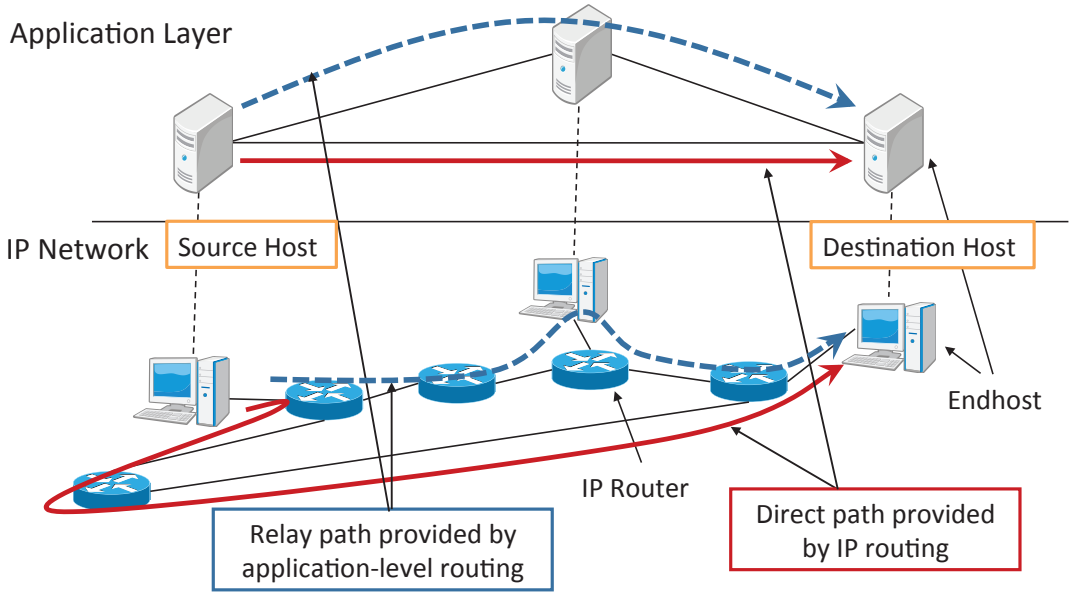


Figure 1.3: Application-level traffic routing

application-level networks for improving user-perceived performance by selfish neighbor node selection. In [17], the authors propose QoS-aware application-level routing by balancing application-level traffic among application-level nodes to ensure end-to-end QoS. All of them target to improve user-perceived performance such as end-to-end latency and available bandwidth for end users' traffic. Figure 1.3 shows a typical example of these advantage. We assume that IP routing uses the direct route and that application-level routing chooses the relay route. The length of the arrow in the IP network represents the value of the end-to-end latency. When we compare the IP routing and the application-level routing from the source host to the destination host in this figure, the direct route has smaller router-level hop counts, but longer end-to-end latency, as compared to the relay route. Therefore, the application-level routing provides better user-perceived performance (i.e., end-to-end latency) than the IP routing.

1.1.3 Content-centric networking

Content-centric networking (CCN) [11] is an architecture, which routes packets based on content name as depicted Figure 1.4, while the current Internet uses the identifier that indicates where the

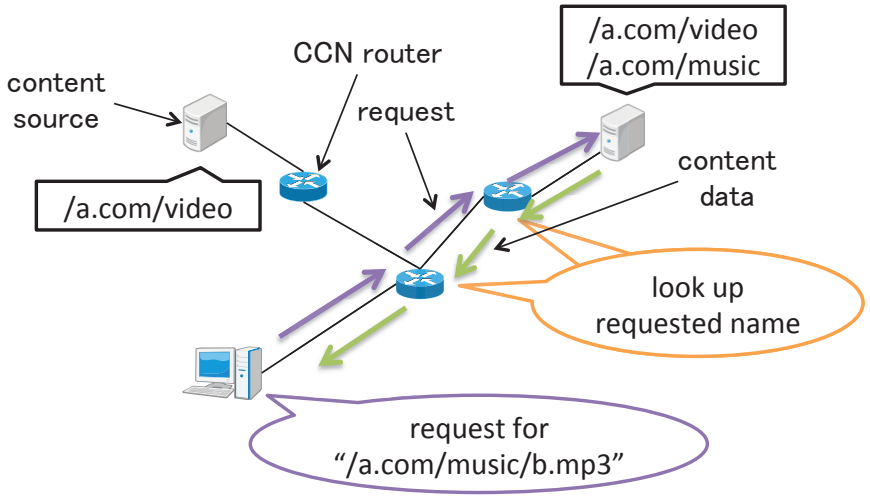


Figure 1.4: Content-centric networking

content holder is, i.e., IP address. This architecture is proposed to respond to the desire of fast and reliable content access, while not imposing excessive consumption of network bandwidth. The CCN can also release end users from maintaining where the requested contents exist, meaning that means the end users can request the content with the content name, without awareness of the location of content holder. The CCN is better suitable for content delivery and content distribution than current IP network layer, because it has acceptable multiple sources for a single content and caching mechanism. The CCN also supports a secure transfer of contents, where all contents transferred by CCN includes signatures of original content holders, and they are encrypted when it is transferred. These features are demonstrated by the authors in [11].

In-network caching is one of the important features of CCN. In CCN, the content traversing CCN routers are cached in the memory space of CCN routers. The CCN routers do not forward the requests for cached contents to the next hop router, alternatively return the cached contents to the end hosts who request the contents. Because of this caching mechanism, the CCN can reduce the traffic volume for repeatedly requested contents and also provide shorter response time for users. For efficient cache utilization in CCN, there are some researches on cache management in CCN. The method in [18] provided a way that the CCN routers on the route could cache without overlaps. In [19], the authors propose a method to distribute the content chunks along the route in probabilistic

manner. The method proposed in [20] considers the cache utilization including the outside of the route to the original content holder, which assigns the contents to be cached at each CCN router according to the request popularity of contents and the CCN routers collaborate on caching.

1.2 Issues for ISPs' cost structure

1.2.1 Impact of application-level traffic routing

Although application-level routing can improve user-perceived performance, it may also generate traffic that violates the ISPs' routing policy. The ISPs may incur additional monetary cost due to such traffic. If these cost increases accumulate, the transit cost over the entire network is increased. Figure 1.5 shows a simple example of this problem. In the figure, there are three endhosts, all of which work as application-level nodes, are connected by application-level links each other. Each application-level link is composed of multiple inter-AS links, each of which is either a transit or peering link. We assume that Node A generates traffic that is routed to Node C. When using the IP or application-level routing that chooses the direct path, the traffic traverses two transit links. Conversely, when the application-level routing utilizes the relay path via Node B, the traffic traverses four transit links between Nodes A and B, and those between Nodes B and C. Therefore, the sum of the transit links traversed by the relay path is increased by two compared with the direct path and as a consequence, the transit cost over the entire network increases. Naturally, there are possibilities that the relay path has lower transit cost than the direct path. However, we consider that the relay path usually has a higher transit cost because it is composed of a number of direct paths.

Even when a transit cost-aware mechanism for application-level routing is developed, an issue of policy mismatch still exists. That is, when end users control the overlay network based on their own objectives, it may degrade the satisfaction of ISPs, and vice versa. Therefore, a novel method is required that considers the objectives of both ISPs and end users.

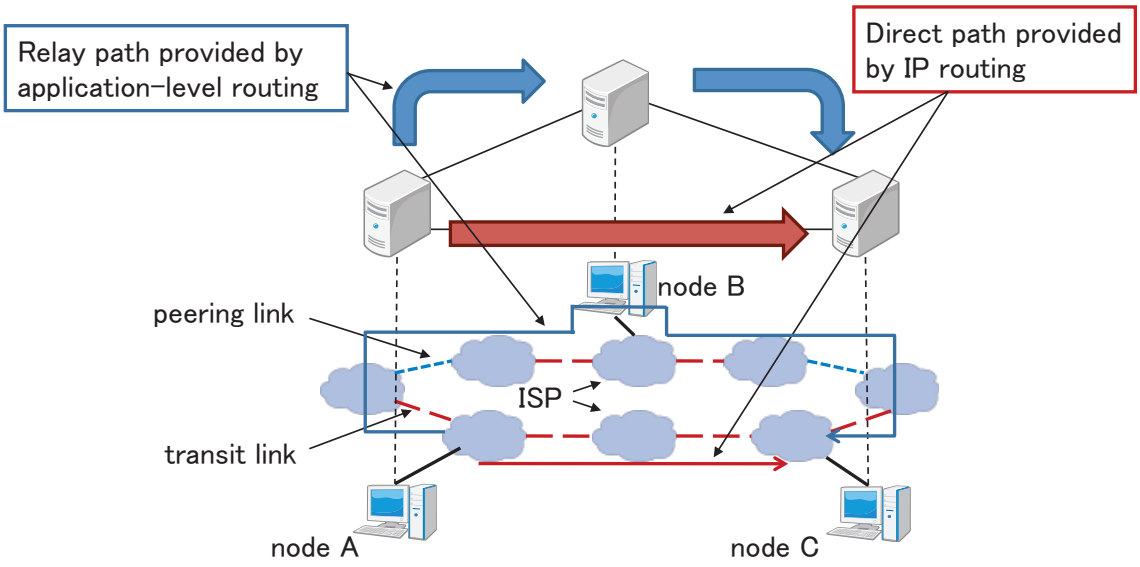


Figure 1.5: Transit cost increase by application-level routing

1.2.2 Potential of content-centric networking to reduce transit cost

As described in Subsection 1.1.3, CCN has the in-network caching mechanism. Reducing the traffic volume by the caching mechanism in CCN has a positive effect on the ISPs' monetary cost. In CCN, when the CCN router that has the requested content exists in the same ISP as the end user, the transit cost is not incurred. Therefore, the CCN can reduce the transit cost by its caching mechanism. The reduction of transit cost becomes large when the cache hit ratio is higher.

Generally, higher hit ratio can be realized by introducing larger storage. However, the memory space for content caching is relatively small compared to the amount of contents required by end users, because content cache is located at the router and it should has shorter access time compared by the endhost-based caching mechanism like Web proxy servers. According to [21], when we use the DRAM memory, it is expected that each CCN router may have only about 10 GB for content cache at a maximum. The methods proposed in [18-20] can be improve the efficiency of cache utilization, however, there are limitations on efficiency of cache memory utilization. We also believe that there is a potential benefit for ISPs connected by the peering link to decrease the transit cost by sharing the CCN router's cache and accessing the cached contents with each other. Although such

cooperative caching mechanism is proposed in [22], there is no concrete architecture to realize the idea.

1.3 Outline of thesis

This thesis begins by a proposal to reduce transit cost caused by application-level routing for individual end users in Chapter 2. In Chapter 3, a distributed application-level route selection method is presented, which is assumed to be conducted by individual operators in multiple ISPs. In Chapter 4, a cache sharing method among multiple ISPs is proposed. Finally, this thesis is concluded in Chapter 5.

1.3.1 Chapter 2: Reducing inter-ISP transit cost caused by application-level routing based on end-to-end network measurement [1-5]

In Chapter 2, we propose a novel method to decrease the transit cost of application-level routing while accounting for the standpoints of end users and ISPs, which we call a *limited application-level routing*. The proposed method chooses application-level routes using a transit cost metric of the routes. We propose two types of route selection methods for the limited application-level routing, which target end users and ISPs objectives, respectively.

The limited application-level routing needs the transit cost metric of application-level routes. For this purpose, we build up a method to estimate the transit cost of application-level routes from end-to-end network performance values that can be measured easily by application-level nodes, such as IP router-level hop count, end-to-end latency and available bandwidth. The estimation method is based on multiple regression analysis of network performance values.

We demonstrate the effectiveness of the proposed method by evaluating the performance of the application-level routing that is assumed to be operated on application-level networks on a PlanetLab [23] and a Japanese commercial network environments. To set a baseline for the discussion, we first evaluate the performance improvement of the application-level routing without a limitation on the transit cost metric. Next, we evaluate the limited application-level routing using precise information on the types of inter-AS links. Then, we show the regression equations used to estimate

value of the transit cost metric for both environments. After that, we evaluate the performance of the limited application-level routing by using the proposed estimation method and discuss parameter settings from the standpoints of ISPs and end users. From the extensive evaluations, we confirm the proposed method has the advantage that the method can achieve considerable reduction on the transit cost, while controlling the application-level routing according to the objectives of both ISPs and end users.

1.3.2 Chapter 3: An application-level routing method with transit cost reduction based on a distributed heuristic algorithm [6-9]

In Chapter 3, we focus on application-level routing based on a coordinated distributed manner performed by application-level nodes, with the aim of improving end-to-end network performance without increasing transit cost. First, we formulate the application-level routing and strictly define an optimization problem for selecting application-level routes with various route selection metrics and a limitation on the transit cost. In general, there are two candidates of coordinated algorithms to achieve near-optimal solutions for the optimization problem: centralized and distributed algorithms. We assume that the operator of each application-level node wants to decide the application-level route on its own. For example, we can easily imagine a use case where each application-level 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 application-level routing method based on a distributed heuristic algorithm that produces near-optimal solutions to the optimization problem. We also design the proposed method to perform route selection not only for a fixed application-level traffic demand, but also in reaction to dynamic application-level traffic demand changes.

For the evaluation of proposed method, we assume that PlanetLab nodes utilize application-level 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 application-level node pair. Next, we evaluate the effectiveness of the proposed method in a situation where the amount of application-level 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 experiment results show that the proposed method achieves considerable improvement of network performance without increasing transit cost. In particular, in the case of using end-to-end latency as routing metric, the number of overloaded end-to-end paths can be reduced by around 65%, as compared that with non-coordinated methods.

1.3.3 Chapter 4: Cooperative cache sharing among ISPs for additional reduction in inter-ISP transit cost in content-centric networking [10]

In Chapter 4, we propose a cooperative cache sharing method among multiple ISPs to improve cache hit ratio for reducing the transit cost effectively. In the proposed method, the cached contents are shared among the CCN routers in ISPs under cooperation. The CCN routers share their content cache without overlapping of the cached contents. A request packet for the cached contents is forwarded to the CCN router who has the content, even when it is not located on the route to the original content holder. This enables to improve cache hit ratio. We introduce a mechanism to keep the consistency among ISPs' cache since cache miss causes the extra traffic on the transit links of cooperating ISPs. We also design mechanisms to balance the network traffic to cached contents between cooperating ISPs to ensure the fairness between ISPs by controlling the CS size for cache sharing and by the content duplication in the shared cache.

We evaluate the performance of the proposed method by simulation experiments using the actual ISPs' IP-level network topologies. From the evaluation results, we show the proposed method can reduce the transit cost effectively compared with the normal CCN caching mechanism by up to 45%, while ensuring the fairness between ISPs' under cooperation.

Chapter 2

Reducing inter-ISP transit cost caused by application-level routing based on end-to-end network measurement

2.1 Introduction

Application-level traffic routing is a routing mechanism on application layer that provides application-level routes for network application traffic. One advantage of application-level routing is that user-perceived network performance, such as end-to-end latency and available bandwidth, can be improved without modifying the current IP network [24-26]. Although this policy mismatch improves end-to-end network performance, it generates a problem for the ISPs' cost structure. Specifically, the inter-ISP transit cost is increased over the entire network [27, 28, 43]. To reduce transit cost, the locality-aware method has been proposed in [29] that controls network traffic based on the locality inferred from the IP address prefix or domain name. However, those types of information are not always suitable for estimating the locality of the Internet topology. Application-layer Traffic Optimization (ALTO) [30], which is based on the concept of P4P [31] is another approach that attempts to reduce transit cost by controlling outgoing traffic from an ISP while considering the utilization of its connected transit and peering links. However, such a mechanism can only optimize outgoing

traffic from a single ISP, and it cannot control incoming traffic. Moreover, that mechanism cannot optimize the end-to-end network traffic governed by multiple interconnected ISPs. To reduce transit cost across the entire network, a routing mechanism based on transit cost information between ISPs on end-to-end paths is required. However, the contract information between ISPs is not available in general and a simple end-to-end measurement or estimation method to obtain this information has yet to be developed.

In this chapter, we propose a novel method to decrease the transit cost of application-level routing while accounting for the standpoints of end users and ISPs, which we call a *limited application-level routing* (we describe application-level as AL for short in this chapter). The proposed method chooses AL paths using a transit cost metric of the paths. We propose two types of path selection methods for the limited AL routing, which target end users and ISPs objectives, respectively.

The limited AL routing needs the transit cost metric of AL paths. For this purpose, we build up a method to estimate the transit cost of AL paths from end-to-end network performance values that can be measured easily by AL nodes, such as router-level hop count, end-to-end latency and available bandwidth. The estimation method is based on multiple regression analysis of network performance values.

We demonstrate the effectiveness of the proposed method by evaluating the performance of the AL routing that is assumed to be operated on AL networks on a PlanetLab [23] and a Japanese commercial network environments. To set a baseline for the discussion, we first evaluate the performance improvement of the AL routing without a limitation on the transit cost metric. Next, we evaluate the limited AL routing using precise information on the types of inter-AS links. Then, we show the regression equations used to estimate value of the transit cost metric for both environments. After that, we evaluate the performance of the limited AL routing by using the proposed estimation method and discuss parameter settings from the standpoints of ISPs and end users.

The remainder of this chapter is organized as follows. In Section 2.2, research background on AL routing is given and the problem of increased transit cost and incentives for reducing it are described. In Section 2.3, we propose a method to reduce transit cost. In Section 2.4, we explain the dataset used for evaluation of the proposed method, and then we present the results of the evaluation in Section 2.5. Finally, in Section 2.6, we summarize our conclusions.

2.2 Background on application-level routing

2.2.1 Effectiveness of application-level routing

Application-level routing can improve end-to-end network performance by choosing the paths based on application-level network performance metrics, such as end-to-end latency, packet loss ratio, available bandwidth, and TCP throughput. This advantage of AL routing is mainly a result of the policy mismatch between IP routing and AL routing. AL routing typically makes their routing decisions that improve user-perceived performance using these metrics. Conversely, 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, and IP-level routing configurations are affected considerably by these cost structures. Two types of links are common between ASes[1]: transit links that connect the upper-level and the lower-level ISPs, and peering links that are used for peering relationship. The monetary cost of the transit link is usually determined by the amount of traffic traversing the link, and transit links can be used by an ISP's customers. In contrast, there is almost no monetary charge for peering links, except for the cost paid to carrier companies for the physical link facilities. Therefore, peering links can be used only by traffic between interconnected ISPs.

A simple example of the advantage of AL routing are described at Section 1.1.2.

2.2.2 Impact on the cost structure of ISPs

Although AL routing can improve user-perceived performance, it may also generate traffic that does not follow to the ISPs' cost structure (i.e., the IP routing policy provided by the ISPs), and so the ISPs may incur additional monetary cost due to such traffic. If these cost increases accumulate, the transit cost over the entire network is increased. A simple example of this problem are described at Section 1.2.1.

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

2.2.3 Related works

The method proposed in [15] selects the AL paths utilizing measurement results of capacity and available bandwidth. In [16], the authors present the method to construct and maintain AL networks for improving user-perceived performance by selfish neighbor node selection. In [17], the authors propose QoS-aware AL routing by balancing AL traffic among AL nodes. All of them target to improve user-perceived performance such as end-to-end latency and available bandwidth for end users' traffic, which is not specified for particular kinds of application. This feature is the same as that of our method proposed in this chapter. However, the methods proposed in [15-17] are not treat the inter-ISP transit cost that incurs a considerable impact from the AL routing as described in Subsection 2.2.2.

The method proposed in [33] uses a *cost* for AL path creation and AL traffic routing in an abstract way and optimizes the cost. In [34], the authors focus on the resource allocation on AL networks and try to deal it as optimization problem. Although these methods can treat various kinds of cost by including it in their optimization problems, they have not considered the inter-ISP transit cost.

2.3 Proposed method

We first explain the network model utilized in this chapter. Next, we propose a *limited AL routing* with two path selection methods. One of those methods appropriate to the standpoint of end users and the other is for that of ISPs. Then, we present some use cases from both standpoints. Finally, we propose a method of estimating a transit cost metric from network performance values that can be obtained easily.

2.3.1 Network model

We assume the network model depicted in Figure 2.1. The underlay network is constructed from a number of ASes, and each AS is constructed from a number of IP routers. Each AS is connected to its neighbors by transit or peering links. A transit cost is incurred whenever traffic traverses a transit

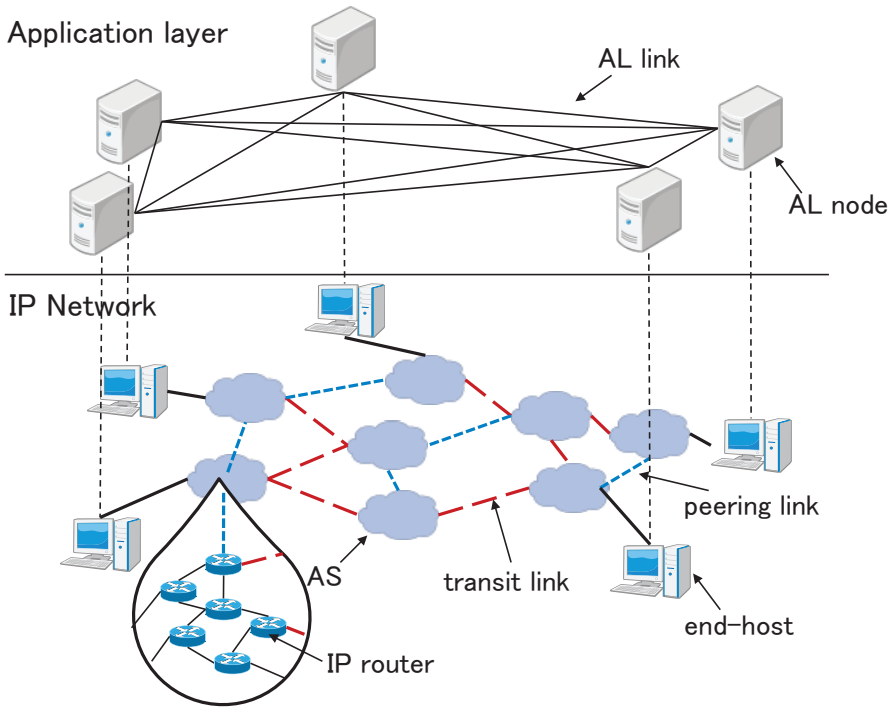


Figure 2.1: Network model

link. Note that we ignore which ISPs connected by transit links are upper-level or lower-level ISPs, since we consider reduction in the transit cost over the entire network.

An AL network is constructed over the underlay network and end hosts located at ASes perform as AL nodes. We term AL nodes just as *nodes* in the remainder of this chapter. We assume that the AL routing can utilize the AL links between all node pairs to evaluate the potential performance to reduce the transit cost due to the AL routing.

An AL routing is operated on the AL network and can provide a route from the source node to the destination node. We consider the following two types of AL paths.

direct path It is a path from the source node to the destination node that is routed directly. A direct path consists of only a single AL link between the source and destination nodes, and so the direct path is equal to that provided by IP-level routing alone.

relay path It is a path from the source node to the destination node via another node. Here, we

consider only two-hop paths, because paths with greater hop counts do not contribute to improve user-perceived performance [32]. Thus, a relay path consists of two AL links.

2.3.2 Limited AL routing

The limited AL routing can be implemented using any of the metrics associated with transit links. As a generalization, we describe the limited AL routing using only a transit cost metric.

In what follows, m_{ij} is the value of the transit cost metric for the AL link between nodes i and j. Hence, the value of transit cost metric for the direct path between nodes i and j and that of the relay path via node k are given, respectively, as follows.

$$M_{ij} = m_{ij} \tag{2.1}$$

$$M_{ikj} = m_{ik} + m_{kj} \tag{2.2}$$

Improving user-perceived performance under inter-ISP transit cost constraint

One path selection method focuses on the upper limit of increases in the value of transit cost metric. This selection method considers the end users' objectives. The constraint on the value of transit cost metric when choosing a relay path instead of a direct path is defined as follows.

$$M_{ikj} \leq M_{ij} + \alpha \tag{2.3}$$

where α is the upper limit of the increase in the value of transit cost metric through using the relay path. The AL routing thus selects the relay path with the best performance from all possible candidates under this constraint. Here, the performance of direct path between nodes i and j is denoted P_{ij}, and the performance of relay path via node k is denoted P_{ikj}. Then, we define the *improvement ratio* of user-perceived performance, which is denoted $\hat{I}_{ij}$, as follows.

$$\hat{I}_{ij} = P_{ij} / \min_{k \neq i,j} \left(P_{ikj} \right) \tag{2.4a}$$

$$\hat{I}_{ij} = \max_{k \neq i,j} \left(P_{ikj} \right) / P_{ij} \tag{2.4b}$$

Here, Equation (2.4a) is used in the case that a low performance metric value represents better performance, such as end-to-end latency. Conversely, Equation (2.4b) is used when a high value represents better performance, such as available bandwidth. Note that when no relay path has better performance than the direct path, the improvement ratio becomes smaller than one. In other words, the AL routing with this path selection method provides the performance improvement for the data transmission between nodes under the limitation on the increase degree of the value of transit cost metric.

Reducing inter-ISP transit cost under user-perceived performance constraint

The other path selection method focuses on the decrease in the AL routing performance. This method considers the ISPs' objectives. When the best performance by the AL routing between nodes i and j without considering the value of transit cost metric is provided by the relay path via node l, we define the constraints on the degree of decrease in AL routing performance as follows.

$$P_{ikj} \leq P_{ilj} \times (1 + \beta) \tag{2.5a}$$

$$P_{ikj} \geq P_{ilj} \times (1 - \beta) \tag{2.5b}$$

where β determines the lower limit of the decrease degree of the performance of the AL routing compared with the best performance. Note that when a low value represents better performance, Equation (2.5a) should be satisfied, and when a high value represents better performance, Equation (2.5b) should be satisfied. Then, the AL routing selects a path with the lowest value of transit cost metric while satisfying Equations (2.5a) or (2.5b). The *reduction in the value of transit cost metric*, which is denoted as $\hat{M}_{ij}$, can be defined as follows.

$$\hat{M}_{ij} = M_{ilj} - \min_{k \neq i,j} \left(M_{ikj} \right) \tag{2.6}$$

In other words, this path selection method can reduce the value of transit cost metric under a given decrease in the user-perceived performance compared with that of the best path.

2.3.3 Use cases

In this subsection, we present some use cases of proposed limited AL routing.

For end users

For the standpoint of end users, we assume the situation where the end users construct an AL network by their end hosts as AL nodes. Each AL node measures the network performance of AL links by end-to-end manner and exchanges the measurement results of network performance with other AL nodes. After that, each end user selects an AL path independently. We can also presume another case where a content provider sets up AL nodes on a number of ISPs and operates AL networks to provide their contents to end users with high network performance. As a similar use case, companies providing cloud network service operate AL nodes and conduct the AL routing to improve network performance among the cloud networks.

In those cases, the end users can achieve the benefit, which is the improvement of network performance, provided by the AL routing. For the case of content provider, they can increase their revenue from end users in return for better quality of content delivery. Hence, the end users and the content provider have incentive to operate it. On the other hand, the AL routing focusing only to improve user-perceived performance may be harmful to ISPs because the AL traffic may generate additional transit cost. The considerable increase of transit cost causes ISPs to control or shut out the AL traffic. The proposed limited AL routing in Subsection 2.3.2 can resolve such situation by setting the upper limit of increase in transit cost generated by AL traffic.

For ISPs

For the standpoint of ISPs, we suppose the case that a number of ISPs set up AL nodes at own IP network. Comparing to the case for end users, when ISPs operates an AL routing, they can monitor the under-lay networks directly. Utilising these measurement results, they can perform the AL routing more effectively than end users. They share the measurement results of network performance among the ISPs and each ISP selects AL paths for the end users belonging to the ISP. Alternatively, we can assume that the ISPs organize an alliance for AL routing and the alliance

operates the AL routing in centralized manner. In both cases, after selecting the AL paths, the ISPs or the alliance provide these paths to end users by an architecture such as ALTO.

In those cases, by using the cost-aware AL routing, the ISPs mainly achieve the benefit, because that the ISPs can reduce the transit cost compared with the case that the end users select the AL paths selfishly. However, if ISPs selects the AL paths only focusing on the reduction of transit cost, the end users lost the incentive to use the AL paths provided by the ISPs since such AL paths do not improve user-perceived performance. The proposed limited AL routing in Subsection 2.3.2 can select the AL paths maintaining the improvement of user-perceived performance while decreasing the transit cost.

2.3.4 Transit cost estimation of an AL path

Although the limited AL routing described above uses a transit cost metric of an AL path, such as the number of transit links, the exact value of the metric cannot be explicitly known by nodes because the contract information between ISPs is not disclosed in general. Furthermore, an effective method to measure the value of transit cost metric in an end-to-end manner has yet to be proposed. Indeed, in [13], the relationships among ISPs are inferred by collecting Border Gateway Protocol (BGP) messages from numerous backbone routers, which are also difficult to obtain by end users. In addition, the relationships between IP address prefix and AS numbers to obtain the AS-level paths are based on BGP messages. Although these information can be obtained at CAIDA [35] and Route Views Project [36] and we utilized them, it is unrealistic in the actual situations that the all AL nodes obtain such information at such as CAIDA and Route Views Project. Furthermore, these information should be obtained periodically, because they change in time. Therefore, we propose a method of estimating the value of transit cost metric of an AL link using network performance values that can be measured easily by nodes.

We first investigate the correlation between the *true* values of transit cost metric of paths between AL nodes obtained by a method such as [13] and network performance values that are obtained easily by end-to-end measurement, such as router-level hop count, end-to-end latency, and available bandwidth. We find linear relationships between the number of transit links that have

strong correlation with the true metric value of transit cost and each network performance value from the graph of the number of transit links vs. each network performance value using the PlanetLab dataset described in Section 2.4. Therefore, we utilize *Pearson's correlation coefficient C* in Equation (2.7).

$$C = \frac{\sum (m^t_{ij} - \bar{m^t})(x_{ij} - \bar{x})}{\sqrt{\sum (m^t_{ij} - \bar{m^t})^2} \sqrt{\sum (x_{ij} - \bar{x})^2}} \tag{2.7}$$

where m^t_{ij} is the true value of transit cost metric of the AL link between nodes i and j, and x_{ij} is each performance value (i.e., router-level hop count, end-to-end latency, and available bandwidth). $\bar{m^t}$ and $\bar{x}$ then represent the average values of each variable, respectively.

Then, to perform the estimation, we select some parameters that are highly correlated to the value of transit cost metric. We conduct a multiple regression analysis on the selected parameters and thus derive the regression equation from the analysis to estimate the value of transit cost metric.

We employ a linear least squares method to derive the regression equation. If x^q_{ij} is the q-th parameter value of the AL link between nodes i and j, then the regression equation to estimate the value of transit cost metric of the AL link, m^e_{ij}, is described as follows.

$$m^e_{ij} = b_0 + b_1 x^1_{ij} + b_2 x^2_{ij} + \ldots + b_n x^n_{ij} \tag{2.8}$$

where b_0 is the intercept of the equation, b_q is the partial coefficient value of the q-th parameter calculated by the multiple regression analysis, and n is the number of parameters.

Once the regression equation is derived, the all AL nodes can estimate the transit cost of AL path by the network performance values that are easily obtained by themselves. In addition, the regression equation can be reused in other network environments if the property of the network environment is similar to where the regression equation is derived.

2.4 Dataset

To evaluate the proposed method, we utilize data obtained from two kind of actual network environments. One network environment is constructed from PlanetLab nodes, and the other from nodes located at Japanese commercial ISPs. To evaluate the AL routing and the proposed method in both environments, we must know the following properties of the end-to-end path between AL nodes: end-to-end latency, available bandwidth, router-level path and hop count, and AS-level path and hop count. We also require the information on the transit/peering relationships between ASes to evaluate a value of transit cost metric of the AL routing. In the remainder of this section, we describe both environments and explain how to obtain their property values.

2.4.1 PlanetLab environment

For the PlanetLab environment, we obtained a dataset among the 459 PlanetLab nodes that were active when we obtained the measurement data. Actually, because a number of end-to-end paths were found for which we could not obtain the measurement data, we used the measurement data of 64 077 end-to-end paths between nodes.

End-to-end latencies We obtained latencies of end-to-end paths between PlanetLab nodes from Scalable Sensing Service (S^3) [37]. In S^3, the measurement results are available for all network paths between PlanetLab nodes, and are summarized every four hours. S^3 uses two types of end-to-end latencies, one is *measured_latency* that is actual measured values and the other is *nv_estimated_latency* that is estimated by the method proposed by Sharma et al. in [38]. Since *measured_latency* was not available for large part of node pairs, we utilized *nv_estimated_latency* in this chapter.

Available bandwidths They were obtained in the same way as end-to-end latencies. We utilized the results of available bandwidth measurements with Spruce [39] in S^3.

IP-level paths and router-level hop counts We conducted `traceroute` commands between all node pairs in PlanetLab. Here, we utilized the `traceroute` results obtained on November 12, 2008.

AS-level paths and AS-level hop counts We converted the IP-level paths into AS-level paths by using the relationships between IP address prefixes and AS numbers, which are available at the Route Views Project [36].

Transit/peering information To obtain a value of transit cost metric for each path, we used the information on transit/peering relationship between ASes that is available at CAIDA [35]. This information is obtained with the method in [13]. However, CAIDA does not provide the relationship information for all links between ASes. Furthermore, there are many IP addresses for routers whose corresponding AS numbers cannot be obtained by the method described above. Therefore, we applied two additional methods to infer the relationship information. The first method is based on the degree of each AS (the number of outgoing links to other ASes). We first obtained the degree of each AS from CAIDA database and then derived the ratio at which the relationship was peering for each pair of degrees of ASes. Figure 2.2 depicts the distribution of the ratio for various pairs of ASes' degrees, where z-axis is the percentage of ASes pairs that have peering links in each cell. Then, the unknown relationship information was stochastically determined according the ratio distribution. The second method is based on the BGP property. When the AS number of the router on an IP-level path cannot be obtained by above-described method, this indicates that the BGP does not advertise the AS number of the router. This may mean that there is no need to advertise the number since the router belongs to the same AS at which the previous-hop router is located. For this reason, as depicted in Figure 2.3, when there exists an IP-level path which is constructed of the router of AS X, the router whose AS number is not advertised, and the router of AS Y, the relationships between each router were estimated as peering and as the relationship between AS X and AS Y, respectively. Consequently, we could infer the unknown relationship between a non-advertised router and AS Y's router once we had already obtained the relationship between AS X and AS Y.

To exhibit the characteristics of the environment, we show the average and variance values of end-to-end latencies and available bandwidth in Table 2.1. We also present the average and variance values of degrees of the ASes where the PlanetLab nodes are located in Table 2.2, which

Table 2.1: Average and variance values of network performance

	PlanetLab		Japanese commercial
	end-to-end latency	available bandwidth	end-to-end latency
average	152 ms	48,214 kbps	31 ms
variance	2.5×10^4	9.5×10^9	3.3×10^2

Table 2.2: Average and variance values of AS-level degree

	PlanetLab	Japanese commercial
average	27	21
variance	12,061	1,028

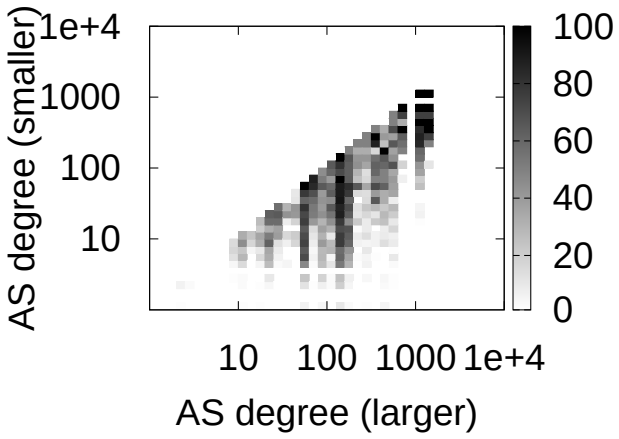


Figure 2.2: Peering ratio from the degree of each AS pair

are calculated by the AS-level links observed in the `traceroute` results.

In the PlanetLab environment, we assumed two types of AL networks. One network was the constructed from all nodes in the PlanetLab environment, which we call the *full PlanetLab network*. The other network, which we call the *generalized PlanetLab network*, was built such that the effect of geographical distribution of AL nodes could be evaluated. The node distribution of the generalized PlanetLab network was constructed to conform to the Internet host distribution. To this end,

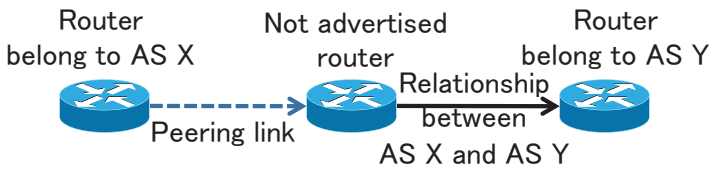


Figure 2.3: Relationships inferred from BGP property

Table 2.3: Number of ASes in each RIR and number of nodes for evaluation

RIR (region name)	number of ASes	number of nodes
ARIN (North America)	24422	50
RIPE NCC (Europe)	21065	43
APNIC (Asia)	5782	12
LACNIC (South America)	2815	6

we referred to the number of ASes in Regional Internet Registries (RIRs) in the current Internet obtained from [40], and to the number of PlanetLab nodes used in each region, which is proportional to the number of ASes (Table 2.3). We randomly selected PlanetLab nodes from each region according to Table 2.3. Comparing these AL networks, we evaluate the effect of geographical node distribution on the proposed method in Section 2.5.4.

2.4.2 Japanese commercial network environment

The dataset for the Japanese commercial network environment was obtained from a colleague. The environment is composed of 18 nodes of 13 Japanese commercial ISPs. The data of 289 end-to-end paths between nodes were used. This dataset included the full-mesh `traceroute` results and end-to-end latencies measured using `ping` commands. Thus, end-to-end latencies, IP-level paths, and router-level hop counts could be determined from the dataset. The dataset utilized in this chapter was obtained on March 22, 2009. Note that we cannot obtain the data on available bandwidth, because that the measurement puts an extra load on the Japanese network environment. Then, the evaluation on the available bandwidth-based AL routing is excluded from Section 2.5.

The average and variance values of end-to-end latencies and degrees of the ASes where the

nodes are located are shown in Tables 2.1 and 2.2, respectively.

The AS-level paths and hop counts and transit/peering information were obtained in the same manner for the PlanetLab environment.

In the Japanese commercial network environment, we assume the AL network constructed from all nodes in the Japanese commercial network environment, and we call this the *Japanese network*.

Because the dataset of Japanese commercial network environment is measured under non-disclosed conditions, we cannot describe the details of the geographical locations of the nodes in the environment. However, we ensure that the Japanese commercial network environment covers the wide area of Japan including large ISPs' network.

2.5 Numerical evaluation

In this section, we first evaluate the performance improvement of AL routing without a limitation on the transit cost metric in order to set a baseline for the discussion. Next, we evaluate the limited AL routing using the precise information of relationships between ASes in order to confirm the potential performance improvement. After that, we present the regression equations, as explained in Subsection 2.3.4, for the two networks on the PlanetLab and one networks on the Japanese commercial environment. Then, we show the evaluation results of the limited AL routing by using the estimated transit cost value calculated through the regression equations. We also confirm the effect of the geographical node distribution on the proposed method.

We utilize end-to-end latencies and available bandwidths between nodes as path selection metrics for the AL routing. We denote the end-to-end latency of the AL link between nodes i and j as δ_{ij}. Then the end-to-end latency of the direct path between the nodes denoted as D_{ij} and that of the relay path via node k denoted as D_{ikj}, are defined as follows.

$$D_{ij} = \delta_{ij} \tag{2.9}$$

$$D_{ikj} = \delta_{ik} + \delta_{kj} \tag{2.10}$$

We do not explicitly include the processing cost of relaying traffic in Equation (2.10) because this

processing cost may be negligibly-small compared with propagation and congestion delays. To cite a case, the end-to-end latency of a relay path is approximately equal to the sum of the latencies of the direct paths that form the relay path in [41]. We denote the available bandwidth of the AL link between nodes i and j as ω_{ij}. Then the available bandwidth of the direct path between the nodes denoted as B_{ij}, and that of the relay path via node k denoted as B_{ikj}, are defined as follows.

$$B_{ij} = \omega_{ij} \tag{2.11}$$

$$B_{ikj} = \min\left(\omega_{ik}, \omega_{kj}\right) \tag{2.12}$$

We utilize Equations (2.4a) and (2.5a) as the improvement ratio and constraint on the performance of AL routing, respectively, when end-to-end latency is employed as the routing metric, and Equations (2.4b) and (2.5b) when the available bandwidth is employed.

Since transit cost is generated by the traffic traversing the transit links, the cost is highly correlated to the number of transit links and the amount of traffic. We assume that the same billing mechanism is used for all transit links and that the traffic volumes between all AL node pairs are equal. Based on these assumptions, when an AL path traverses transit links, the AL path costs one per transit link in the evaluation. We utilize the value calculated by this definition as the transit cost metric for the limited AL routing. Of course, we can assume a specific billing mechanism for each transit link. However, because information is unavailable on how ISPs configure their billing mechanisms in practice, we use the simplest transit cost metric (i.e., one per transit link) in the evaluation.

We further assume that the value of transit cost metric based on the transit/peering information obtained by the method described in Section 2.4 is the *true* value of transit cost metric, because we consider that the transit/peering information reflects the actual network condition, since this information is acquired from numerous BGP routing tables and `traceroute` results. Moreover, the information has high accuracy compared with the estimated value of transit cost metric calculated by Equation (2.8), which is derived from only the network performance values easily obtainable by the nodes. The distribution of the true value of transit cost metric between PlanetLab nodes is shown in Figure 2.4. We use this distribution as the baseline for the discussion in the evaluation.

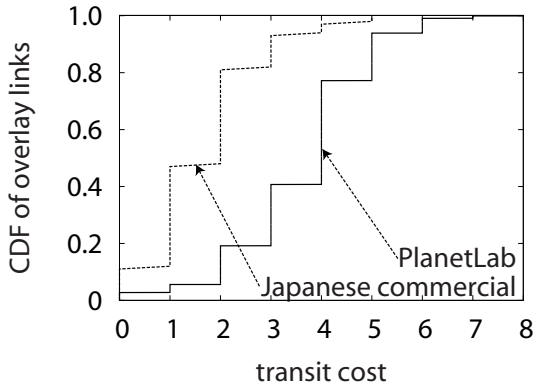


Figure 2.4: Distribution of the true value of transit cost metric of AL links in PlanetLab network environment and Japanese commercial network environment

2.5.1 Unlimited AL routing

The "no limit" lines in Figure 2.5 plot the cumulative distribution of the improvement ratios as defined in Equations (2.4) for the full PlanetLab network. Here, end-to-end latency is employed as the performance metric in Figure 2.5(a) and available bandwidth is employed in Figure 2.5(b). The results in Figure 2.5 are based on the median value of dataset recorded over a two-week period from November 12, 2008 to November 25, 2008. The ratio of node pairs that have at least one relay path that has smaller end-to-end latency than the direct path is 22%. In the case of available bandwidth, the percentage is 97%. These results agree with the results in [32], implying that available bandwidth-based AL routing improves user-perceived performance significantly.

The "no limit" line in Figure 2.7 shows the result in the same manner as the line in Figure 2.5 for the Japanese network. Since we do not have the data about available bandwidth for this environment, the evaluation is only on the end-to-end latency-based AL routing. The ratio of node pairs that have at least one relay path that is better than the direct path is 15%.

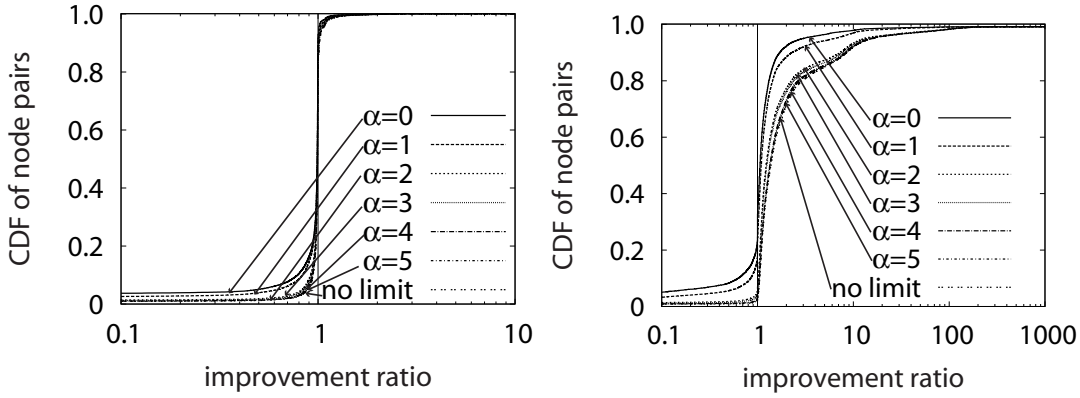
2.5.2 Limited AL routing with precise information on transit links

Next, we show the results for the case when a limitation is placed on the true value of transit cost metric. This value is based on the precise relationship information among ASes obtained by the

method explained in Section 2.4. The detailed algorithm of limited AL routing can be found in Section 2.3.2.

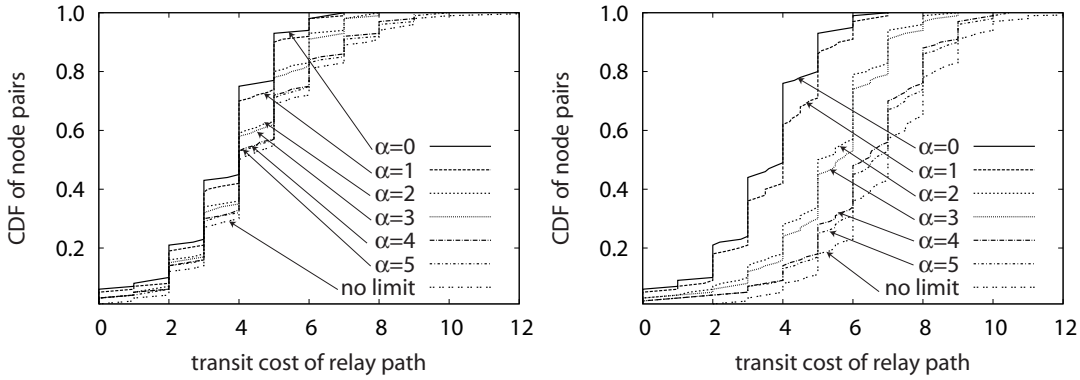
Figure 2.5 exhibits the cumulative distribution of the improvement ratio for the full PlanetLab network when limiting the increase in the true value of transit cost metric. Path selection method here focuses on the upper limit of the increase in the value of transit cost, α, by using Equation (2.3). Note that when α is too small, relay paths satisfying the limitation cannot be found for some node pairs. These node pairs are accounted for at the origin of the x-axis. Figure 2.5 indicates that, no matter which routing metric is used, as α increases, the performance improvement of the AL routing approaches that for the case without the limitation. Furthermore, when α is greater than or equal to three or four, the performance improvement of the AL routing becomes approximately equal to the case without the limitation. From these results, we conclude that the AL routing with the upper limit of the increase in the true value of transit cost metric can provide the performance improvement of the AL routing similar to the case without the limitation, when the upper limit of the increase is greater than or equal to three or four. Figure 2.6 shows the cumulative distribution of the true value of transit cost metric for AL paths between all node pairs. These paths are the same as those chosen in Figure 2.5 for the full PlanetLab network. This figure tells that when α is two in the AL routing with the proposed method, the 80-th percentiles are 5.0 for the case of latency and 5.0 for the case of available bandwidth, respectively, whereas the values without the limitation are 6.0 and 6.4. When α is three, the 80-th percentiles are 5.5 for the case of latency and 7.0 for the case of available bandwidth. For the total transit cost of the all AL paths, when $\alpha = 2$, the limited AL routing reduces the transit cost by 11% for the end-to-end latency and by 25% for the available bandwidth comparing the case without the limitation. When $\alpha = 3$, these values are 9% for the end-to-end latency and 22% for the available bandwidth. Then, the limited AL routing with the precise information can reduce transit cost to a certain degree, although this degree is small for the case of latency.

Figure 2.7 shows the results in the same manner as Figure 2.5 for the Japanese network. The trend of the results is the same as that in the full PlanetLab network, which is that the performance improvement approaches that for the case without limitation as α increases, except one difference. That is, when α is greater than or equal to one (three or four for the full PlanetLab network), the



(a) end-to-end latency (b) available bandwidth

Figure 2.5: Improvement ratio distribution with the limitation on the true value of transit cost metric (full PlanetLab network)



(a) end-to-end latency (b) available bandwidth

Figure 2.6: Transit cost distribution with the limitation on the true value of transit cost metric (full PlanetLab network)

performance improvement is approximately equal to that without the limitation. This lower value of α is due to the difference in the network property between both environments, that is, the true transit cost of paths in the Japanese commercial network environment is smaller than that in the PlanetLab environment. Figure 2.8 then shows the results in the same manner as Figure 2.6 for the Japanese network. When α is one in the AL routing with the proposed method, the 80-th percentile is 3.0 whereas the value without the limitation is 4.0. For the total transit cost of the all AL paths,

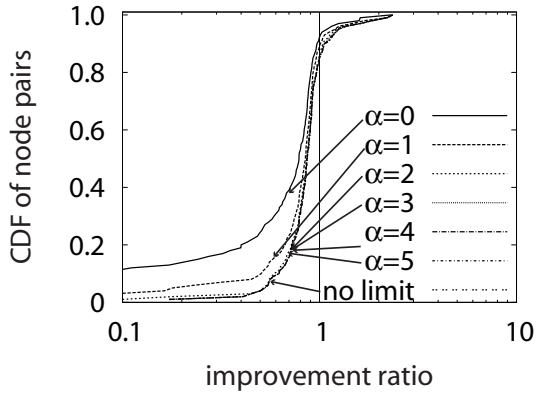


Figure 2.7: Improvement ratio distribution with the limitation on the true value of transit cost metric (Japanese network)

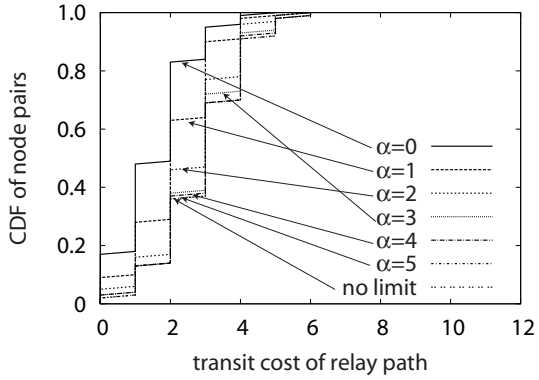


Figure 2.8: True metric value of transit cost distribution with the limitation on the true value of transit cost metric (Japanese network)

when $\alpha = 1$, the limited AL routing reduces the transit cost by 27%. The same advantage as for the full PlanetLab network is thus revealed.

2.5.3 Limited AL routing with estimated value of transit cost value

Regression equations and estimation accuracy

To evaluate the limited AL routing with the proposed estimation method in Subsection 2.3.4, we first derived regression equations (Equation (2.8)) for the three AL networks described in Section 2.4.

Table 2.4: Correlation coefficients (full PlanetLab network)

Router-level hop count	0.420
End-to-end latency	0.300
Available bandwidth	-0.027

Table 2.5: Partial coefficients of the regression equation

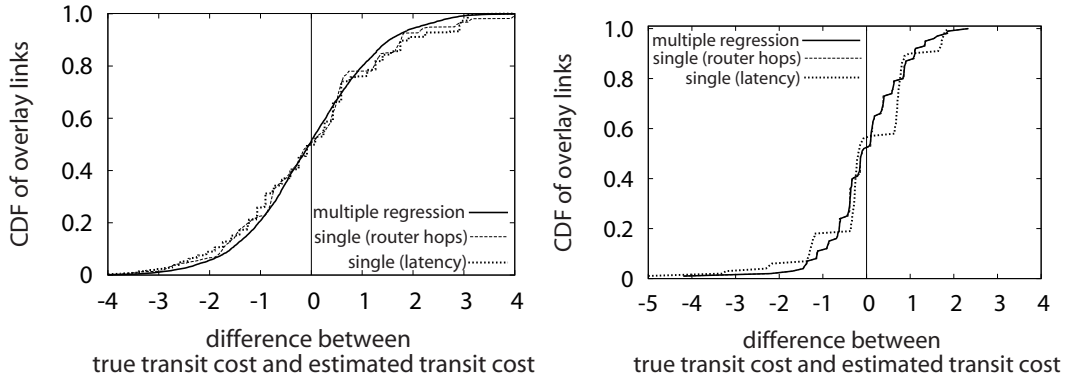
	b_y	b_r	b_d
Full PlanetLab network	1.22	0.135	0.00263
Japanese network	-1.48	0.240	-0.000889
Generalized PlanetLab network	0.846 (0.20)	0.145 (7.56×10^{-4})	0.00120 (1.08×10^{-6})

We calculated the correlation coefficients in Equation (2.7) between the true value of transit cost metric and the following three metrics: router-level hop count, end-to-end latency, and available bandwidth for all nodes in the PlanetLab environment. The calculation results are listed in Table 2.4, and based on these results, we omitted the available bandwidth from the regression equation because its correlation was quite weak compared with the other two metrics. Since we do not have available bandwidth data for the Japanese network, the same parameters (i.e., router-level hop count and end-to-end latency) were also selected in the regression equation. When the router-level hop count and the end-to-end latency of the AL link between nodes i and j are denoted as h_{ij} and δ_{ij}, respectively, then Equation (2.8) can be rewritten as follows.

$$\mu^e_{ij} = b_y + b_r h_{ij} + b_d \delta_{ij} \tag{2.13}$$

where b_y is the intercept of the equation, and b_r and b_d are the partial coefficients of the router-level hop count and the end-to-end latency, respectively. The partial coefficients (b_y, b_r, and b_d), which are the results of the multiple regression analysis, for the three networks are listed in Table 2.5.

We also show the evaluation results of the estimation accuracy of the regression equation to verify the effectiveness of the analysis. The estimation error between the true and the estimated value of transit cost metric of an AL link is calculated for each AL node pair. The true value of transit cost metric of the AL link between nodes i and j is denoted as μ^t_{ij}, and the estimated value by



(a) PlanetLab (b) Japanese commercial network

Figure 2.9: Estimation error distribution of the regression equation for all AL links in each network environment

the regression equation is demoted as μ^e_{ij}. Then the estimation error of the AL link, d_{ij}, is obtained as follows.

$$d_{ij} = \mu^e_{ij} - \mu^t_{ij} \tag{2.14}$$

Note that we consider both positive and negative values of the estimation errors. When the value is positive, it indicates that the transit cost is overestimated and the available relay paths are excessively restricted. Conversely, when the value is negative, it indicates that the transit cost is underestimated and the transit cost can be relaxed actually.

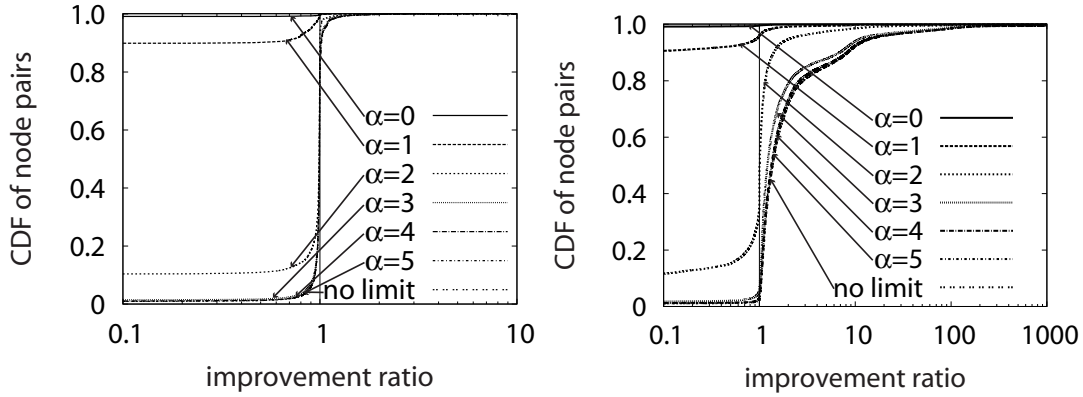
Figures 2.9 plot the cumulative distribution of d_{ij} for all node pairs in each network environment. For comparison, the results of the single regression analyses on the router-level hop count and end-to-end latency are also plotted, respectively. The figures indicate that the maximum absolute estimation errors resulting from Equation (2.8) are smaller than four for the PlanetLab environment, three for the Japanese commercial network environment. The absolute estimation errors are smaller than one for almost 60% of the AL links in both network environments. Furthermore, compared with the results obtained by the single regression analyses, the multiple regression equation can give the highest estimation accuracy.

The differences between the regression equations for both environments can be observed in

Table 2.5. These differences of network properties may be caused by the differences between the PlanetLab and the Japanese commercial network environments. PlanetLab is a global research network and is constructed from the nodes that are in universities and enterprises, whereas the Japanese commercial environment is constructed from the nodes located at Japanese commercial ISPs. In addition, PlanetLab nodes are spread more geographically than Japanese commercial environment. Hence, the proposed method can obtain the regression equations appropriate to each network's properties.

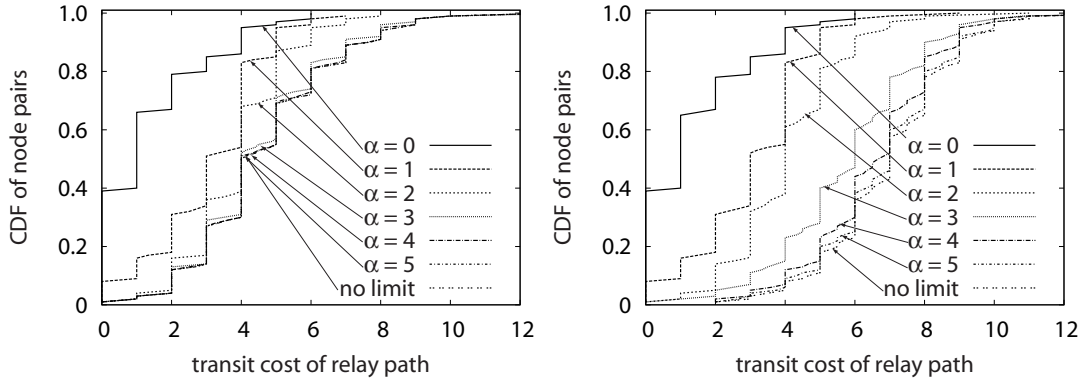
Improvement in user-perceived performance under limitation on inter-ISP transit cost

Figure 2.10 plots the results in the same manner as Figure 2.5 for the full PlanetLab network using the estimated value of transit cost metric instead of the true value. This figure tells that when α is smaller than three, many node pairs who do not have any relay path and the portion increases significantly compared with the results in Figure 2.5, since a significant portion of the node pairs cannot find any relay paths satisfying the limitation. This is because of the estimation error described in Subsection 2.5.3. In contrast, when α is greater than or equal to three, the improvement is approximately the same as in the case without the limitation and that with the limitation on the true value of transit cost metric (Figure 2.5). From these results, we conclude that the AL routing with the proposed method, which has no precise information on transit links, can achieve the same performance as the case with the precise information. Figure 2.11 shows the results in the same manner as Figure 2.6 for the full PlanetLab network. This figure tells that when α is two in the AL routing with the proposed method, the 80-th percentiles are 5.0 for the case of latency and 5.0 for the case of available bandwidth, respectively. When α is three, the 80-th percentiles are 6.0 for the case of latency and 7.5 for the case of available bandwidth. For the total transit cost of the all AL paths, when $\alpha = 2$, the limited AL routing reduces the transit cost by 25% for the end-to-end latency and by 47% for the available bandwidth comparing the case without the limitation. When $\alpha = 3$, these values are 3% for the end-to-end latency and 17% for the available bandwidth. Comparing Figures 2.6 and 2.11, when α equals to zero or one, we can see that the limited AL routing with the estimated transit cost excessively limits the transit cost, which is because of the estimation error of



(a) end-to-end latency (b) available bandwidth

Figure 2.10: Improvement ratio distribution with the limitation on the estimated value of transit cost metric (full PlanetLab network)



(a) end-to-end latency (b) available bandwidth

Figure 2.11: Transit cost distribution with the limitation on the estimated value of transit cost metric (full PlanetLab network)

regression equation. However, we consider that these value of α is too tight limitation, which can be observed in Figure 2.10. In the appropriate range of α (i.e., α is more than or equals to two) for the network performance, the limited AL routing with the estimated transit cost reduce a certain transit cost while maintaining the network performance, though there are some estimation error of transit cost.

Figure 2.12 shows the results in the same manner as Figure 2.10 for the Japanese network. The

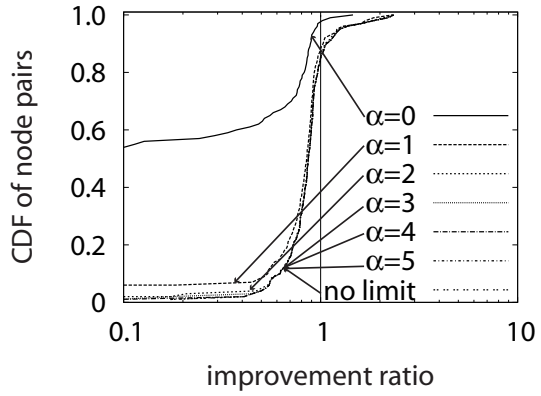


Figure 2.12: Improvement ratio distribution with the limitation on the estimated value of transit cost metric (Japanese network)

trend of the results is the same as that for the full PlanetLab environment except the exact value of α. When α is greater than or equal to one, the AL routing performance is approximately the same as the case with the true value of transit cost metric and the case without the limitation (Figure 2.7). Figure 2.13 then shows the results in the same manner as Figure 2.11 for the Japanese network. When α is one in the AL routing with the proposed method, the 80-th percentile is 3.0 whereas the value without the limitation is 4.0. For the total transit cost of the all AL paths, when $\alpha = 1$, the limited AL routing reduces the transit cost by 18%. The same advantage as for the full PlanetLab network is thus revealed.

From the viewpoint of the trade-off relationship between the performance improvement of AL routing and the transit cost, Figure 2.11 indicates that the greatest reduction in the true value of transit cost metric is when α is equal to zero. However, the improvement ratio becomes less than one for a number of node pairs, implying that these node pairs cannot achieve any improvement in user-perceived performance by the AL routing. Conversely, when α is greater than or equal to three for the case of latency and four for the case of available bandwidth, almost no reduction is found in the true value of transit cost metric. We thus conclude that $\alpha = 2$ for the case of latency is the best value from the viewpoint of the trade-off relationship between the performance of AL routing and the reduction in the transit cost for the full PlanetLab network. For the case of available bandwidth, we conclude $\alpha = 3$ is the best value.

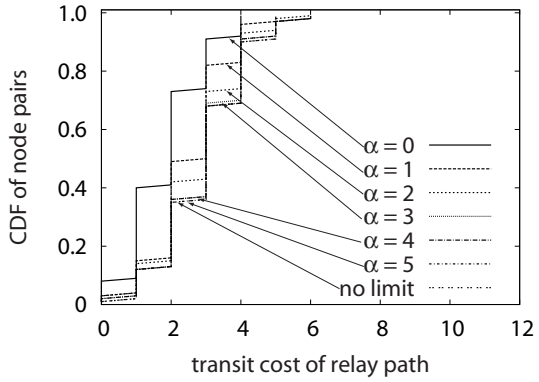
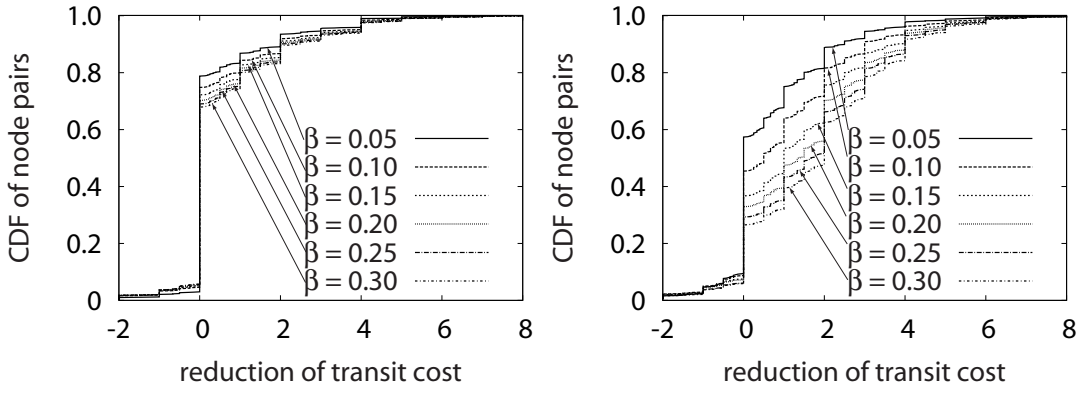


Figure 2.13: True metric value of transit cost distribution with the limitation on the estimated value of transit cost metric (Japanese network)

Based on the results for both the PlanetLab network and Japanese commercial network environments, the parameter α may be affected by the scale of the target network environment. Specifically, for a large network, the more largest value we should choose for α. The scale of network can be known through the IP-level or AS-level hop counts of end-to-end paths. In general, the parameter α can be set by the scale of target network and the degree of constraint required by the operator (i.e., end users).

Reduction in inter-ISP transit cost under limitation on user-perceived performance

Figure 2.14 shows the results of limited AL routing with the path selection method that focuses on the degree of decrease in the performance of the AL routing, where β is the lower limit of the degree in Equations (2.5). Figures 2.14(a) and 2.14(b) plot the distribution of true value of transit cost metric reduction for the full PlanetLab network when the end-to-end latency and the available bandwidth employed as the routing metric, respectively. This figure indicates that the proposed method can reduce the true value of transit cost metric by at least one in 16% of node pairs when the end-to-end latency is used as the routing metric and by at least one in 33% of node pairs when the case for available bandwidth, allowing only a 5% decrease in the AL routing performance. We can achieve a greater reduction in the true value of transit cost metric by allowing a greater decrease in user-perceived performance. For example, if we can allow a 30% decrease in the AL routing



(a) End-to-end latency

(b) Available bandwidth

Figure 2.14: Distribution of reduction in the true value of transit cost metric with the limitation on the decrease in the AL routing performance (full PlanetLab network)

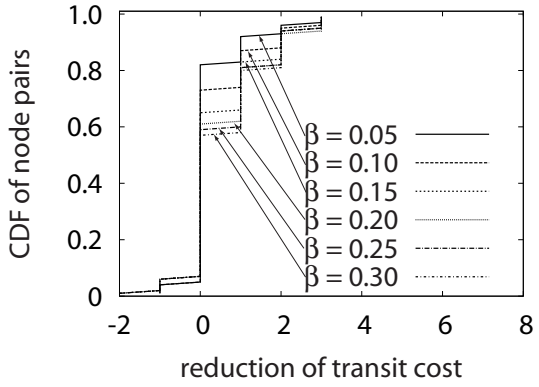


Figure 2.15: Distribution of reduction in the true value of transit cost metric with the limitation on the decrease in the AL routing performance (Japanese network)

performance, the true value of transit cost metric can be reduced by at least one in 25% and 68% node pairs when the end-to-end latency and the available bandwidth is employed as the routing metric, respectively.

Figure 2.15 shows the results in the same manner as Figure 2.14 for the Japanese network. The trend of the results is the same as that for the full PlanetLab environment except the exact value of β. When allowing a 5% decrease and a 30% decrease in the AL routing performance, we can reduce the true value of transit cost metric by at least one in 17% and 42%, respectively.

In practice, the parameter β can be determined by the degree of constraint required by the operator (i.e., ISPs). For example, ISPs measure the performance of direct and relay paths and determine β under the constraint such that the relay paths have sufficiently better than that of direct path.

2.5.4 Effect of geographical distribution of AL nodes

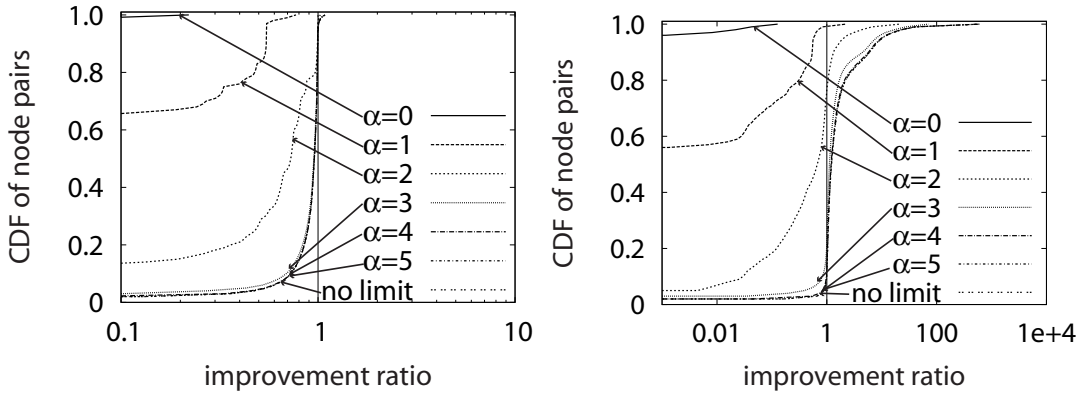
We conducted node selections twenty times for the generalized PlanetLab network according to the method described in Subsection 2.4.1 and calculated the partial coefficients of the regression equation. The average and variance (in parentheses) values of the partial coefficients are listed in Table 2.5. Since the variances are significantly smaller than the average values, we consider that the node selection in the generalized PlanetLab network does not affect the performance of the proposed method.

Figure 2.16 plots the improvement ratio distribution of the AL paths for the generalized PlanetLab network. Comparing Figures 2.10 and 2.16, the trend of the results is the same as that for the full PlanetLab network, especially when α is larger than or equal to three. From these results, we verify that the proposed method is effective for not only the AL network constructed from all nodes in the PlanetLab environment, which are mainly located in North America and Europe, but also in a more general AL network.

Since the trend of the results corresponding to Figure 2.14 for the generalized PlanetLab environment is the same as that for the full PlanetLab environment, we do not show the results here.

2.6 Conclusion

In this chapter, we proposed a method to reduce transit costs caused by AL routing while accounting for the objectives of both ISPs and end users. The proposed method utilizes a transit cost metric of an AL paths and chooses an AL path that can satisfy the objectives and constraints of both parties. Through the extensive evaluation using measurement results taken from the actual network environments, which were the PlanetLab and the Japanese commercial network environments, we confirmed the effectiveness of the proposed method. The results revealed that the advantage of the



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

Figure 2.16: Improvement ratio distribution with the limitation on the estimated value of transit cost metric (generalized PlanetLab network)

proposed method whereby we could estimate the transit cost of AL paths using measurable network performance values. Furthermore, the method could control the AL routing according to the standpoints of end users and ISPs while reducing the transit cost over the entire network. Through the evaluation results, we confirmed the suitable parameter values for both network environments.

In general, the operator of the AL routing with the proposed method can decide the degree of limitations based on their objectives and constraints. Using suitable parameters for a target network environment, the AL routing with the proposed method can satisfy both of a reduction of transit cost and improvement of end-to-end network performance.

Chapter 3

An application-level routing method with transit cost reduction based on a distributed heuristic algorithm

3.1 Introduction

Application-level route selection can inflate the monetary cost incurred by 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. Such a situation can be expected because the application-level path relaying an application-level node includes more than one IP-level path. Furthermore, selfish application-level route selection performed by multiple application users can lead to a decrease in path performance due to overload by route overlaps. For example, in [44] a number of non-coordinated overlay networks cause oscillations in route selection. Even if each user of application-level routing can exactly measure the performance of application-level links, route overlaps and oscillations occur due to scheduling conflict of path selection.

In this chapter, we focus on application-level traffic routing based on a coordinated manner performed by application-level nodes, with the aim of improving end-to-end network performance without increasing transit cost (we describe application-level as AL for short in this chapter), which

can utilize for the same use cases in Chapter 2. 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 limitation on the transit cost. In general, there are two candidates of coordinated algorithms to achieve near-optimal 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 distributed heuristic algorithm that produces near-optimal 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 chapter is organized as follows: In Section 3.2, we describe the background of the present research. In Section 3.3, we define the optimization problem for AL route selection. In Section 3.4, we propose a novel AL traffic routing method. In Section 3.5, we show the results of evaluating the proposed method. Finally, in Section 3.6, we present our conclusions and describe avenues of future research.

3.2 Route overlaps and impact on ISP cost structure in application-level routing

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. That situations lead to the degradation on the

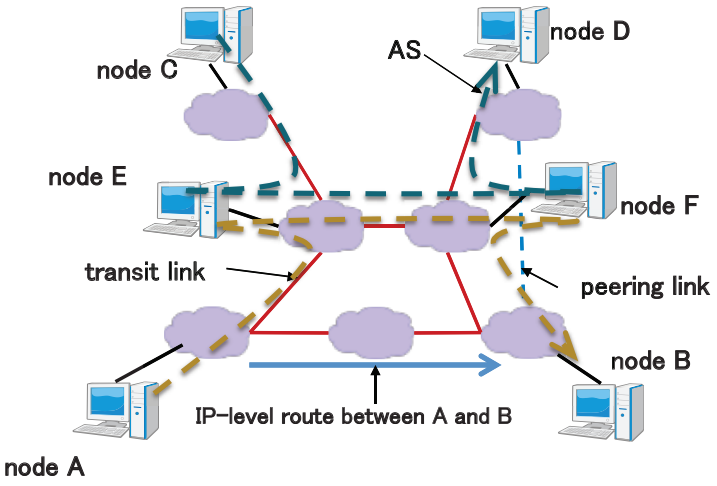


Figure 3.1: Problems on application-level routing

benefits of AL routing. Figure 3.1 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. 3.1 each AL link includes more than one inter-AS link, 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 the native IP routing or AL routing that chooses the direct path, the traffic traverses two transit links. Conversely, when the 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.

3.3 Application-level route optimization problem

We begin this section by explaining the network model assumed in this chapter. We then formulate the AL routing and define the optimization problem for selecting AL routes.

3.3.1 Network model

We assume a network model as depicted in Fig. 3.2. 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. 3.2, the AL route drawn with dotted line consists of two AL links, each of which is the native IP-level paths between the end hosts.

3.3.2 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

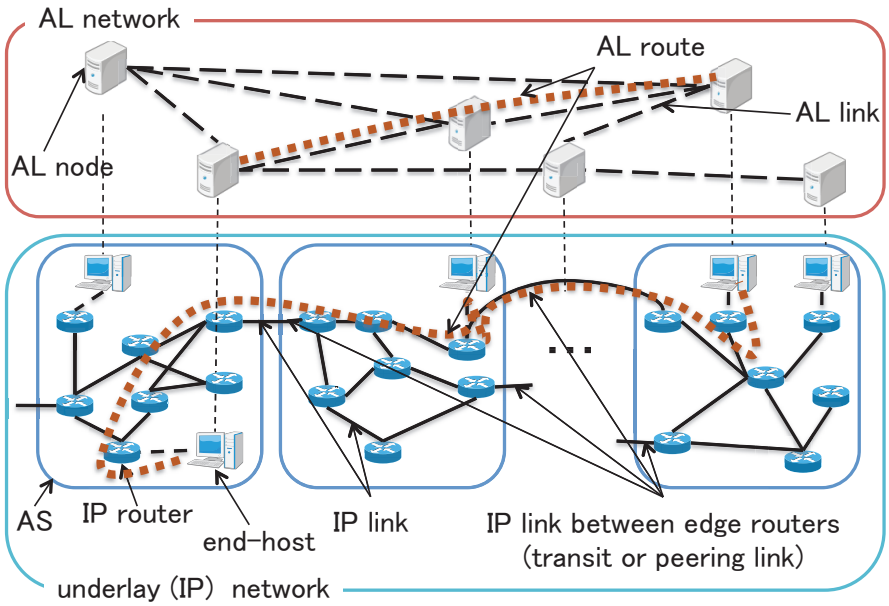


Figure 3.2: Network model

$1, 2, \ldots, (N-1)N$ and $1, 2, \ldots, M$.

$$R^{\mathrm{IP}} = \begin{pmatrix} IP_1^1 & \cdots & IP_1^{(N-1)N} \\ \vdots & \ddots & \vdots \\ IP_M^1 & \cdots & IP_M^{(N-1)N} \end{pmatrix} \tag{3.1}$$

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:

$$\mathcal{E} = \{e_1^{\mathrm{AL}}, e_2^{\mathrm{AL}}, \ldots, e_{(N-1)N}^{\mathrm{AL}}\} \tag{3.2}$$

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, …, 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:

$$\begin{aligned}\Gamma_j^{\mathrm{AL}} \quad = \quad & \{(p_1, p_2, \ldots, p_h) | h \geq 1,\ s_{p_1} = s_j, t_{p_h} = t_j, \\ & t_k = s_{k+1}\ (2 \leq h,\ 1 \leq k \leq h-1), \\ & e_{p_k}^{\mathrm{AL}} = 1\ (1 \leq k \leq h)\}\end{aligned} \tag{3.3}$$

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:

$$R^{\mathrm{AL}} = \begin{pmatrix} AL_1^1 & \cdots & AL_1^{(N-1)N} \\ \vdots & \ddots & \vdots \\ AL_{(N-1)N}^1 & \cdots & AL_{(N-1)N}^{(N-1)N} \end{pmatrix} \tag{3.4}$$

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, as follows:

$$\mathcal{Y} = R^{\mathrm{IP}}\mathcal{X}^{\mathrm{IP}} + R^{\mathrm{IP}}R^{\mathrm{AL}}\mathcal{X}^{\mathrm{AL}} \tag{3.5}$$

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):

$$\mathcal{D}^{\mathrm{AL}} \quad = \quad f_D(\mathcal{Y})R^{\mathrm{AL}} \tag{3.6}$$

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:

$$\mathcal{B}^{\mathrm{AL}} \quad = \quad f_B(\mathcal{Y}, R^{\mathrm{AL}}) \tag{3.7}$$

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.

$$\mathcal{C}^{\mathrm{AL}} \quad = \quad f_C(\mathcal{Y}, R^{\mathrm{AL}}) \tag{3.8}$$

We now define the limitation on transit cost in AL route selection. We regard the transit cost of the direct paths as a baseline, and set the limitation on 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:

$$R^{\mathrm{DR}} = \begin{pmatrix} 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} \tag{3.9}$$

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

$$C^{\mathrm{DR}} \quad = \quad f_C(\mathcal{Y}, R^{\mathrm{DR}}) \tag{3.10}$$

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

$$c_j^{AL} \quad \leq \quad \alpha c_j^{DR}\ (\forall j | j \in \Theta) \tag{3.11}$$

where Θ is the set of identifiers of AL node pairs that have traffic demand. Equation (3.11) 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 of the latencies 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:

$$\begin{array}{ll} \text{minimize :} & (\sum_{j \in \Theta} d_j^{\mathrm{AL}})/|\Theta| \\ \text{subject to :} & r_j \in \Gamma_j^{\mathrm{AL}} \\ & c_j^{AL} \leq \alpha c_j^{DR}\ (\forall j | j \in \Theta) \end{array} \tag{3.12}$$

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

$$\begin{aligned} \text{maximize :} \quad & (\sum\nolimits_{j\in\Theta} b_j^{\mathrm{AL}})/|\Theta| \\ \text{subject to :} \quad & r_j \in \Gamma_j^{\mathrm{AL}} \\ & c_j^{AL} \leq \alpha c_j^{DR} \ (\forall j | j \in \Theta) \end{aligned} \tag{3.13}$$

3.4 Proposed method

In this section, we propose an AL routing method, based on obtaining near-optimal solutions to the problem described in Section 3.3. For this purpose, we take advantage of a popular heuristic algorithm known as *simulated annealing* (SA). As described in Section 3.1, because the distributed algorithm is desirable for application scenarios of AL routing, we utilize the distributed simulated annealing (DSA) proposed in [46]. 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.

3.4.1 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, *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. The process of DSA is slightly different from that of SA so that SA is conducted in a distributed manner. That is, a number of agents determine the parts of the state and exchange them with each other to share the whole state. Each agent then calculates the cost of the gathered state and determines whether or not to change the part of state corresponding to itself.

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, it 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.

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.

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. The candidates of AL routes are restricted by the constraint condition in Eqs. (3.12) and (3.13).

Cost function Cost()

This function estimates the network performance obtained by the given state as the argument and returns the average value of end-to-end latencies or available bandwidths of all AL routes. These costs correspond to the optimization problems (Eqs. (3.12) and (3.13)). In addition, we normalize the state cost by the initial state cost to avoid the transition probability affected by the absolute value of the cost.

Transition probability function Probability()

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

$$\text{Probability}(T, S, S_{tmp}) = e^{-\frac{\text{Cost}(S_{tmp})-\text{Cost}(S)}{T}} \tag{3.14}$$

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 [47]. We use the following typical cooling schedule function in SA:

$$\text{Cooling}(T, I) = \gamma T \ (0 < \gamma < 1) \tag{3.15}$$

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.

3.4.2 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 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()

Algorithm 1 Algorithm for static route selection on AL node i

$I_i \leftarrow 0, \quad T_i \leftarrow T_{i_{init}}, \quad S_i \leftarrow S_{i_{init}}$
while $T_i > T_{low}$ **do**
  Update(S_i)
  $S_{i_{tmp}} \leftarrow$ Neighbor(S_i)
  if Cost(S_i) $\geq$ Cost($S_{i_{tmp}}$) **then**
    $S_i \leftarrow S_{i_{tmp}}$
  else
    $r_i \leftarrow$ Random(1)
    if $r_i <$ Probability(T_i, Cost(S_i), Cost($S_{i_{tmp}}$)) **then**
      $S_i \leftarrow S_{i_{tmp}}$
    end if
  end if
  $I_i \leftarrow I_i + 1$
  $T_i \leftarrow$ Cooling(T_i, I_i)
  if $I_i \bmod U_i = 0$ **then**
    SendNeighbor(S_i)
  end if
end while

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 originating itself occur, 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.

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
```

3.5 Evaluation

In this section, we show the evaluation results of the proposed algorithms described in Section 3.4, assuming that the PlanetLab nodes constitute an AL network and conduct AL routing.

3.5.1 Dataset and settings

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) [37]. S^3 provides the measurement results among PlanetLab nodes every 4 hours. In this chapter, we use the measurement results obtained on October 18–19, 2010.

AS-level routes

We converted the IP-level routes into the AS-level routes using the relationships between IP address prefixes and AS numbers, available at the Route Views Project [36]. We use the data obtained on April 16, 2009. Although the AS number data is older than the other data, we believe this does not affect the evaluation results because attached AS numbers are not changed so frequently.

The relationships between ASes

We utilize the relationships between ASes as provided by CAIDA [35] on January 20, 2010, to calculate the transit cost of AL routes, as described below.

Cost functions

In what follows, we explain the functions f_D, f_B, and f_C in Eqs. (3.6)–(3.8) 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.
- The queuing delay at an AL link occurs only at the tight IP link.
- The queuing delay included by the measurement delay is negligibly small compared to that caused by the AL traffic.

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:

$$d_j^q = \frac{\frac{g_j - a_j + x_j}{c_j}}{1 - \frac{g_j - a_j + x_j}{c_j}} \cdot \frac{P}{g_j} \tag{3.16}$$

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:

$$d_j = d_j^q + d_j^p \tag{3.17}$$

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 [48]. 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:

$$f_{\mathcal{B}}(i) = (a_i - \sum\nolimits_{j \in \Theta^i} b_j^{\mathrm{AL}})/|\{k|b_k^{\mathrm{AL}} = 0, k \in \Theta^i\}| \tag{3.18}$$

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. (3.18) 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:

$$b_j^{\mathrm{AL}} = f_{\mathcal{B}}(i) \;\; (j|b_j^{\mathrm{AL}} = 0, j \in \Theta^i) \tag{3.19}$$

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:

$$c_j^{\mathrm{AL}} \quad = \quad \beta_j x_j \tag{3.20}$$

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:

$$v_i^j = \begin{cases} 1 & (IP_i^j = 1 \text{ and } i \text{ is a transit link}) \\ 0.05 & (IP_i^j = 1 \text{ and } i \text{ is a peering link}) \\ 0 & (IP_i^j = 0 \text{ or } i \text{ is not AS-level link}) \end{cases} \tag{3.21}$$

The value of β_j was calculated from Eq. (3.22) as follows:

$$\beta_j = \sum_{i=1}^{M} v_i^j \tag{3.22}$$

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.

Evaluation scenarios and evaluation 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 value of end-to-end latencies, the number of AL routes that use the overloaded AL links (referred to as the 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 [49]. That is, traffic flows that each of them requires 100 kbps are generated in accordance with the 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 [49] 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 two-second 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 value of available bandwidth.

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 whole time in the 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 [32]. Other parameters for the proposed method are shown in Table 3.1.

For comparison purposes, we show the evaluation results of 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-cooperation* method below.

Table 3.1: Parameters for the evaluation

Number of AL nodes	30
T_{init} and T_{re}	0.15
γ in Eq. (3.15)	0.995
U	20
T_{low}	10^{-6}
C_{th}	10

Table 3.2: Average value of end-to-end latencies classified by AL traffic demand and number of overloaded AL routes

traffic demand	proposed method ($\alpha = \infty$)	proposed method ($\alpha = 1$)	non-cooperation 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

3.5.2 Evaluation results

Static algorithm

We first show the evaluation results using end-to-end latency as a routing metric. Table 3.2 shows the average values of end-to-end latencies of the AL routes selected by the proposed and non-cooperation 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 limitation on transit cost ($\alpha = \infty$ in Eq. (3.12)) and those with the limitation ($\alpha = 1$). From Table 3.2, we can observe that the proposed method provided slightly larger end-to-end latencies than did the non-cooperation method. However, the non-cooperation method generated the overloaded AL routes twice as much as the proposed method without the limitation on transit cost. Note that the average value of end-to-end latencies was calculated except for the overloaded AL routes. Therefore, the average value by the non-cooperation method was

smaller end-to-end latencies than that by the proposed method.

The reason for the difference in the number of overloaded AL routes can be explained by Table 3.3, which exhibits the samples of the AL route selection results. The values in parenthesis lateral to each AL node pair, (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 3.3, 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-cooperation methods selected the direct route. However, the number of overlaps in the proposed method was smaller than that in the non-cooperation method. This is because the proposed method shares the 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 the detour AL route to avoid using the direct route that is 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-cooperation method. These samples represent that the proposed method can select AL routes considering the bandwidth utilization on AL links including AL traffic demand of other AL node pairs in the coordinated manner, that results in avoiding overlaps and overload on AL links.

Table 3.4 shows the average value of the transit cost of the selected AL routes by the proposed method with and without the limitation on the transit cost. Although the transit cost could be reduced by 20% for the case with the limitation, we can observe that the overloaded AL routes increased compared with the case without the limitation in Table 3.2. This is because that the number of AL links available for the proposed method with the limitation is smaller than those without the limitation.

We next present the evaluation results using the available bandwidth as a routing metric. Fig. 3.3 shows the distribution of the available bandwidth of paths between AL node pairs. The average value of available bandwidth in the proposed method was $54,738$ kbps, while that in the non-cooperation method was $26,570$ kbps. We can observe from this figure that almost all AL node

Table 3.3: 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-cooperation 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)

Table 3.4: Average value of transit cost of the AL routes

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

pairs could achieve the considerable improvements by the proposed method compared with that by the non-cooperation method, which is brought by the advantage of the proposed method in avoiding AL route overlaps like the case of end-to-end latency shown in Table 3.3.

From the above results, we conclude that the proposed method, which works in the coordinated manner between AL nodes, can reduce congestion and avoid overlaps on AL links. The limitation on transit cost can efficiently decrease that cost 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 limitation.

Dynamic algorithm

Figures 3.4 and 3.5 show the average value of end-to-end latencies and the number of overloaded AL routes as a function of simulated time, when end-to-end latency as a routing metric. The tendency of the results at each second in Fig. 3.4 is similar to that in Table 3.2, where the end-to-end latency achieved by the proposed method was slightly larger than that by the non-cooperation method. On

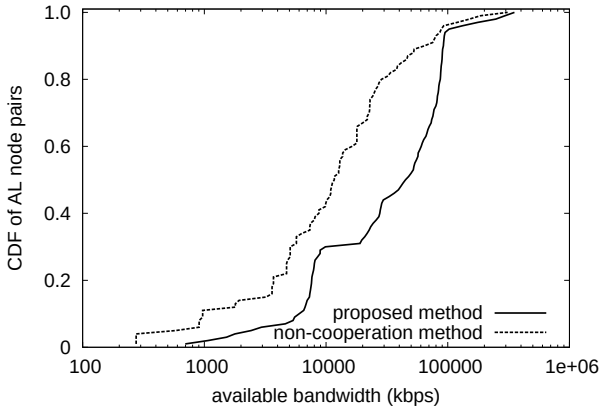


Figure 3.3: Distribution of available bandwidth between the AL node pairs

the other hand, from Fig. 3.5, the number of overloaded AL routes was significantly affected by the AL traffic demand changes. At between 2 sec and 3 sec, and at between 4 sec and 5 sec, the number of overloaded AL routes increased in both methods. However, when comparing two methods, the proposed method significantly reduced the increase in overloaded AL routes. Specifically, the proposed method reduced the overloaded AL routes by roughly 65% from that by the non-cooperation method at all times.

Table 3.5 shows the samples of changes in selected AL routes at between 2 sec and 3 sec by the same manner as Table 3.3. In the case of 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-cooperation 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 3.6 shows the changes in the available bandwidth, presented in the same manner as Fig. 3.4, when available bandwidth is utilized as a routing metric. The tendency of 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-cooperation method, regardless of the changes in the

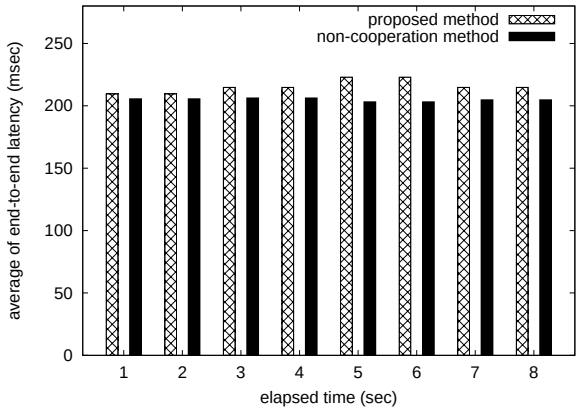


Figure 3.4: Average value of end-to-end latencies over time

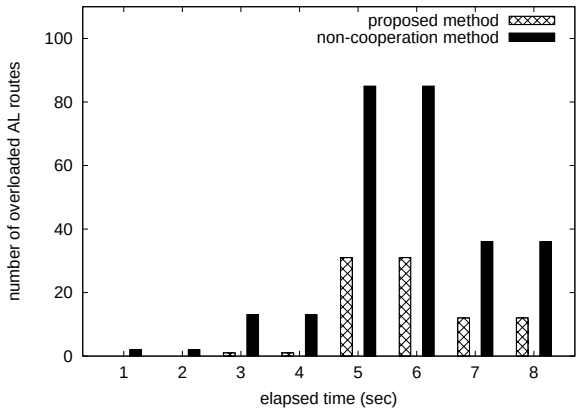


Figure 3.5: Number of overloaded AL routes over time

number of AL node pairs that required the bandwidth. This is because that the proposed method shares knowledge of 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.

Table 3.5: Samples of changes in selected AL routes

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

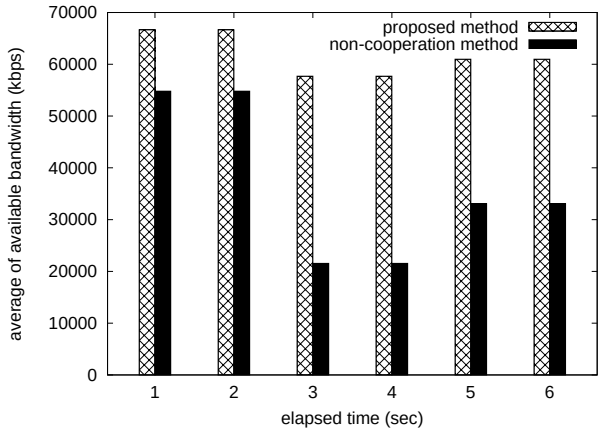


Figure 3.6: Average value of available bandwidth over time

3.6 Conclusion

In this chapter, we proposed the application-level routing method that works in a coordinated manner based on distributed simulated annealing. First, we formulated the application-level routing and defined an optimization problem for selecting AL routes. After that, we proposed the application-level routing method based on distributed simulated annealing with two algorithms, one for static AL traffic demand and the other that dynamically reacts to 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 reducing congestion or performance degradation of 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 becomes more efficient.

Chapter 4

Cooperative cache sharing among ISPs for additional reduction in inter-ISP transit cost in content-centric networking

4.1 Introduction

Content-centric networking (CCN) [11] is a new architecture which routes packets based on content name, while the current Internet uses the identifier that indicates where the content holder is, i.e., IP address. The end users can request the content with the content name, without the awareness of the location of content holder.

In-network caching is one of the important features of CCN. In CCN, the content traversing CCN routers are cached in the memory space of CCN routers called as Content Store (CS). The CCN routers do not forward the requests for cached contents to the next hop router, alternatively return the cached contents to the end hosts who request the contents. Because of this caching mechanism, the CCN can reduce the traffic volume for repeatedly requested contents and also provide shorter response time for users.

Reducing the traffic volume by the caching mechanism in CCN has a positive effect on the ISPs' monetary cost. In general, ISPs have the transit links for ensuring connectivity to the whole Internet. In CCN, when the CCN router that has the requested content exists in the same ISP as the end user, the transit cost is not incurred. Therefore, the CCN can reduce the transit cost by its caching mechanism. The reduction of transit cost becomes large when the cache hit ratio is higher. Generally, higher hit ratio can be realized by introducing larger storage. However, the memory space of CS is relatively small compared to the amount of contents required by end users.

Peering link is the other kind of inter-ISP links, which is used for traffic between inter-connected ISPs by the link to suppress the transit cost. It requires no monetary cost for traversed traffic, except for that of the physical link facilities. We believe that there is a potential benefit for ISPs connected by the peering link to decrease the transit cost by sharing the CCN router's cache and accessing the cached contents with each other. Although such cooperative caching mechanism is proposed in [22], there is no concrete method to realize the idea.

In this chapter, we propose a cooperative cache sharing method among multiple ISPs to improve cache hit ratio for reducing the transit cost effectively. In the proposed method, the cached contents are shared among the CCN routers in ISPs under cooperation. The CCN routers share their CSes without overlapping of the cached contents. A request packet for the cached contents is forwarded to the CCN router who has the content, even when it is not located on the route to the original content holder. This enables to improve cache hit ratio. We introduce a mechanism to keep the consistency among ISPs' cache since cache miss causes the extra traffic on the transit links of cooperating ISPs. We also design to balance the network traffic to cached contents between cooperating ISPs to ensure the fairness between ISPs by controlling the CS size for cache sharing and by the content duplication in the shared cache. We evaluate the performance of the proposed method by simulation experiments using the actual ISPs' IP-level network topologies. From the evaluation results, we show the proposed method can reduce the transit cost effectively compared with the normal CCN caching mechanism, while ensuring the fairness between ISPs' under cooperation.

The remainder of this chapter is organized as follows: In Section 4.2, we describe the background of the research in this chapter. In Section 4.3, we list the challenges for the cache sharing among multiple ISPs. In Section 4.4, we propose a novel cache sharing method. In Section 4.5,

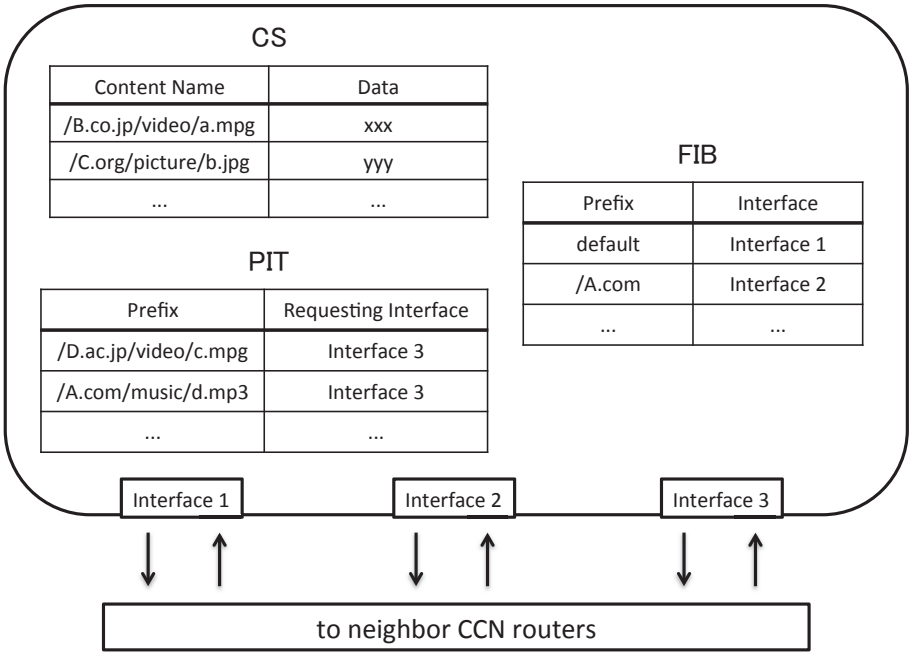


Figure 4.1: Overview of CCN router

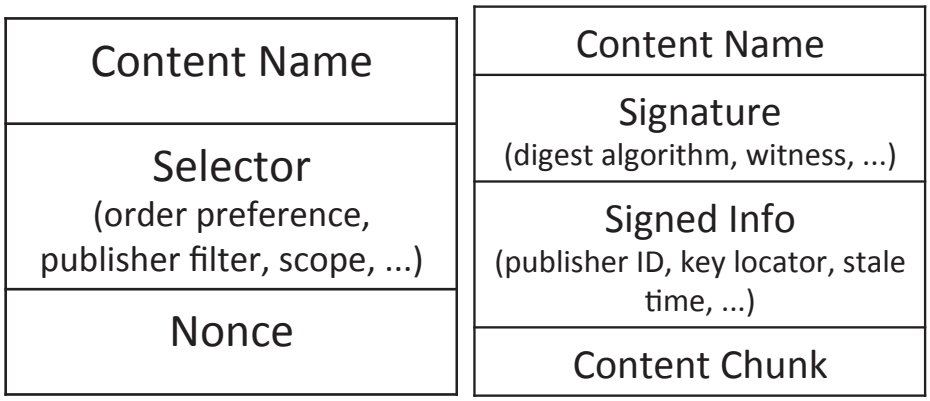
we show the results of evaluating the proposed method. Finally, in Section 4.6, we present our conclusions and describe avenues of future research.

4.2 Background

4.2.1 Content-centric networking [11]

A CCN router is constructed by three main components, that are Pending Interest Table (PIT), Forwarding Information Base (FIB), and Content Store (CS) as shown in 4.1. PIT maintains a list of request packets that are waiting for the content. FIB is the routing table for forwarding request packet to sources of requested content. CS is the memory space that caches the contents traversing the router itself. The packets in CCN are of two types, that are Interest and Data packets as depicted in Figure 4.2. Interest packet is request packet for content, and Data packet is the data chunk of content.

The brief overview of packet forwarding in CCN is as follows. First, the end host generates an



(a) Interest packet (b) Data packet

Figure 4.2: Kinds of packets in CCN

Interest packet for a content and sends it to the neighbor CCN router. The CCN router that received the Interest packet refers the own FIB, then forwards the packet to the appropriate neighbor CCN router. Repeating this process on the CCN routers, the Interest packet reaches to the host who has the requested content. The host receiving the Interest packet divides the requested content into a number of Data packets and returns them to the end host along the reverse path that the Interest packet traverses. Also, the CCN routers on the path cache the Data packets as a content chunk to their CSes. The CCN router returns the cached chunks to any Interest packets that request the cached content chunks. Due to such in-network caching mechanism, CCN suppress the traffic volume for repeatedly requested contents, as well as provides the shorter response for users.

4.2.2 Related works

There were a number of researches about endhost-based caching mechanism. [50] evaluated the benefits when the ISPs manage the cache for the P2P traffic in cooperated manner. [51] proposed the caching method for P2P traffic by the proxy servers on a single ISP or on a number of ISPs in cooperated manner. Both of [50, 51] targeted the caching on the application-layer for P2P environment. Therefore, these methods cannot be applied directly to the CCN's in-network caching.

In [52-54], a cache sharing method for the Web proxy servers was proposed. [52] is a famous

protocol and implementation that construct a hierarchical structure of proxy servers to achieve scalability and manageability. The method in [53] forwards the request to other proxy servers, when the cache miss is occurred, according to the advertised information. [54] proposed a method that utilizes a consistent hashing mechanism to assign and discover cached contents. However, the authors in [52-54] did not consider the network traffic problem among multiple ISPs when they utilize the proposed method in cooperative manner. Also, the methods are operated at the application-layer and cannot apply to the CCN directly because required performance and restriction are different between differnent layers.

[18, 19] proposed the methods to improve the efficiency of in-network caching in CCN. The method in [18] provided a way that the CCN routers on the route could cache without overlaps. The method in [19] distributes the content chunks along the route in probabilistic manner. Both of [18, 19] intended to utilize the cache on the route efficiently, then they cannot utilize the cache outside the route to the original content holder. Therefore, the ISPs cannot adopt these methods directly in cooperative manner.

The method proposed in [20] considers the cache utilization including the outside of the route to the original content holder, which assigns the contents to be cached at each CCN router according to the request popularity of contents and the CCN routers collaborate on caching. However, when we use the method in [20] among multiple ISPs in cooperative manner, the balancing of network traffic becomes a problem, which was not considered in [20]. Additionally, the method has a possibility of cache miss that leads the policy violation of transit links as mentioned in Section 4.3.

To the best of our knowledge, there is no efficient method for the cache sharing among multiple ISPs. Therefore, in this chapter, we propose the method to realize such cache sharing.

4.3 Challenges of cache sharing

In this section, we summarize challenges to be solved in order to realize cache sharing among multiple ISPs in CCN.

4.3.1 Challenges on cache sharing

Advertisement of cached contents among CCN routers

For effective cache sharing among CCN routers, the information on what contents are cached at which CCN router should be maintained. In [11], the authors assume to utilize OSPF or IS-IS as routing protocols. However, they do not propose the advertisement mechanism of cached content names among CCN routers since cache sharing is outside of the scope of [11]. OSPFN [55] is a routing protocol to be developed for CCN, which is based on OSPF. It has the advertisement mechanism of content names by content holder. CCN routers update their FIB according to the advertised content name. However, in [55], it is not considered to advertise cached content names among CCN routers.

One possible way to advertise cached content names is to exploit OSPFN. However, the cached contents are replaced frequently because that the memory space for CS is limited. In addition, the method for advertisement in OSPFN utilizes a flooding mechanism. Therefore, it can be expected that when we use OSPFN for advertising cached content names among CCN routers, numerous messages are generated with frequent replacement of cached contents. To limit the message volume to feasible area, we need to tune the frequency of advertisement carefully.

Forwarding of Interest packets

When the advertised information is inconsistent among CCN routers, there is possibility to occur cache miss. The cache miss brings the forwarding loop of Interest packet as follows:

- The CCN router where a cache miss is occurred forwards the Interest packet to the source of content.
- A CCN router on the route makes a misjudgment that the CCN router where a cache miss was occurred has the cached content, and forwards the Interest packet to it.
- The cache miss is occurred again.

Figure 4.3 depicts these behaviors between CCN routers. We need the mechanism to avoid such forwarding loop.

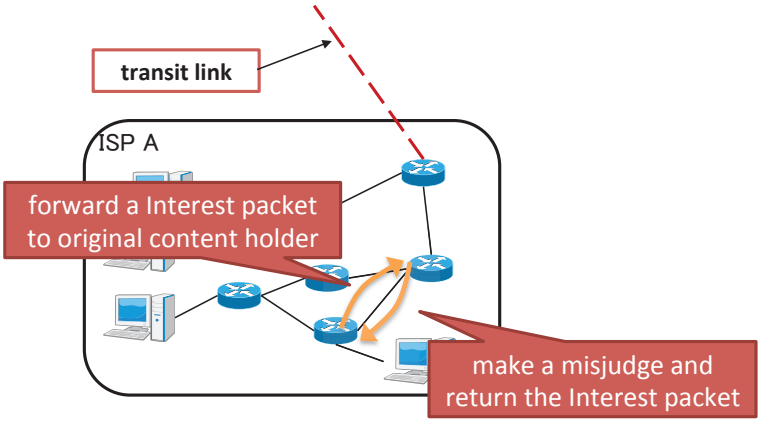


Figure 4.3: Forwarding loop due to cache miss

4.3.2 Challenges on inter-ISP traffic

Assuming that the problems in Subsection 4.3.1 are overcome, we consider that two ISPs interconnected by a peering link share the cached contents to reduce their transit cost. Based on [11], the straightforward way for packet forwarding by the CCN router receiving an Interest packet is as follows:

- If the requested content exists in own CS, the CCN router returns the cached content.
- Otherwise, the CCN router looks up the advertised content names by other CCN routers, and forwards the Interest packet to the appropriate CCN router when the content name exists.
- If there is no cached content in own and other CCN routers' CSes, the CCN router forwards the Interest packet to the source of requested content.

When we assume these behaviors, the following problems are emerged.

Traffic unbalance between ISPs

When the requested content is located at a CCN router in a cooperating ISP, the Interest packet and the corresponding Data packets traverses the peering link. Therefore, the traffic unbalance may happen due to difference in cache hit ratio and requested frequency between the ISPs. For example,

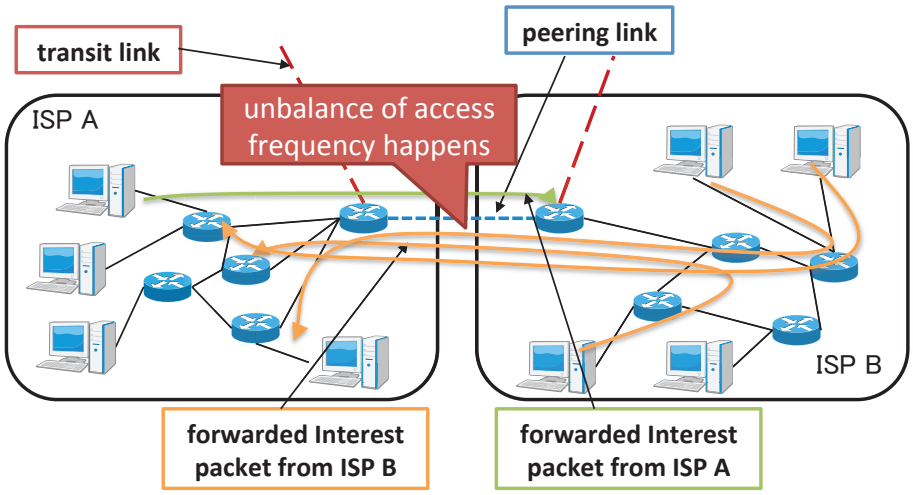


Figure 4.4: Traffic unbalance between ISPs

depicted in Figure 4.4, the access to the cached contents from the ISP B to ISP A drastically exceeds that from ISP A. The excess unbalance of the traffic on peering links may break the peering relationship between ISPs. Then, a mechanism to ensure the fairness of traffic volume between ISPs is required in the cache sharing among multiple ISPs.

Free-riding on transit links

If there is a possibility of cache miss due to inconsistency of advertised cached content names among CCN routers, the policy violation of transit link usage occurs depicted as Figure 4.5. In general, the transit link should be used only for the traffic generated by own customers. However, in the cache sharing, when a CCN router in the ISP forwards a Interest packet to the cached content in the other ISP's CCN router and a cache miss is occurred due to the inconsistency, the Interest packet is forwarded to the original content holder from the CCN router where the cache miss is occurred. For example, in Figure 4.5, the Interest packet from ISP A is forwarded to the source of content by ISP B's CCN router. Then, the transit link is used by the Interest packet that is not generated by the customer of the owner of the link. Furthermore, since the Data packets traverse the reverse route of the Interest packet, the Data packets also use the same transit link. This means that although the purpose of cache sharing is to reduce the transit cost, the ISP may incur the additional transit cost

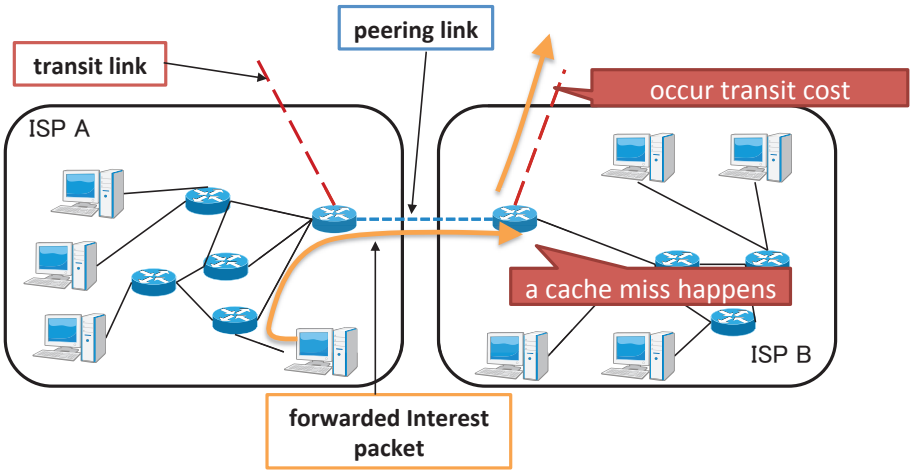


Figure 4.5: Free-riding problem due to cache miss

due to the traffic generated by the other ISP's customer, which we call the *free-riding* problem. We need to avoid the problem or to balance the frequencies of using the transit links by accessing the shared cache.

4.4 Proposed method

In this section, we propose a novel cache sharing method among CCN routers as well as the extension to accommodate multiple ISPs for further transit cost reduction while overcoming the problems described in Section 4.3.

The proposed method consists three main components and two additional components. The main components are as follows:

- Advertisement of cached contents among CCN routers
- Cache management and decision which contents to be advertised
- Forwarding of Interest packet according to the advertised information

The additional components are especially for cache sharing for multiple ISPs:

- Duplicate cached contents for maintaining ISP's cost and user-perceived performance

- Balancing traffic to ensure the fairness between ISPs

Note that, in CCN, contents are divided into a number of chunks and each chunk has a unique name. In what follows, however, we assume that the content is stored in CCN router's cache space without being divided into chunks.

In the remainder of this section, we first explain a network model assumed in the present research. After that, we show the details of five components of the proposed method. Finally, we show an idea to suppressing volume of content advertisement messages.

4.4.1 Network model

We assume a network model as depicted in Figure 4.6. The network consists a number of ISPs' networks, each of which is constructed from a number of CCN routers. Each CCN router has a unique name to be identified by other CCN routers. The behavior of CCN routers follows [11], where OSPFN is used as a routing protocol. ISPs are interconnected by transit or peering links. A transit cost is incurred when traffic traverses transit links. We refer routers interconnected by inter-ISP links as *edge routers*. Endhosts are connected to each CCN router and they request contents by sending Interest packets to the CCN router.

4.4.2 Advertisement of cached contents

We divide the advertisement of cached contents into two parts for *intra-ISP sharing* and *inter-ISP sharing*. This partitioning enables to decrease and balance the network traffic between ISPs as described later. An advertisement message has two fields, which are a content name and a CCN router name holding the content. Note that when a CCN router removes a content from its shared cache, the corresponding advertisement message includes the removed content name. For intra-ISP advertisement, all CCN routers including the edge router advertise the cached contents to all other CCN routers in the ISP, utilizing OSPFN. When a CCN router receives an advertisement message, the router replies an acknowledgement to the source router of the message. For inter-ISP advertisement of cooperating ISPs, two edge routers interconnected by a peering link chooses the cached contents to be shared respectively, and advertise the contents with each other. Then,

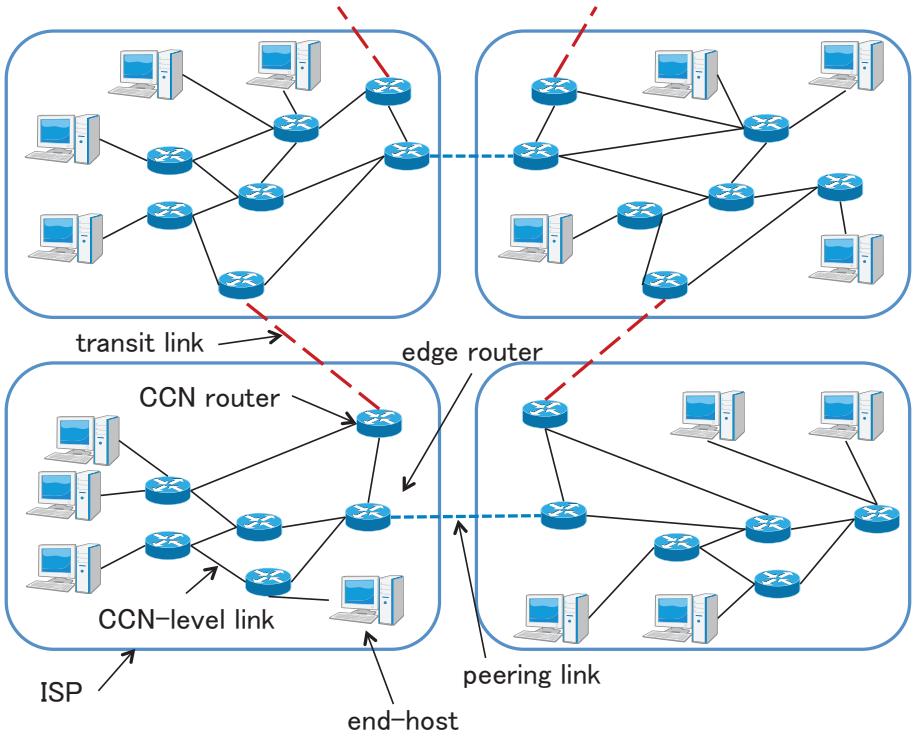


Figure 4.6: Network model

each edge router advertises the content names from an ISP under cooperation (we refer such ISP as *partner ISP* in the remainder) to all other CCN routers in own ISP in the same manner as for the intra-ISP advertisement. When balancing the network traffic between the cooperating ISPs, the ISPs make negotiation based on access frequencies to and from each other ISP. The detail of negotiation is described in Subsection 4.4.6.

OSPFN has a mechanism to advertise locations of contents. Because the advertisement mechanism in OSPFN utilizes a simple flooding mechanism, it is not reasonable to generate advertisement messages on each change in the cached contents to be shared at a CCN router. Therefore, in the proposed method, each CCN router advertise cached contents for intra-ISP sharing at regular intervals of T_{intra}. The inter-ISP advertisement by edge routers is activated at regular intervals of T_{inter}.

4.4.3 Cache management

There are a lot of methods of cache management for Web contents in past literature. Since most of them utilize Least Frequently Used (LFU) or Least Recently Used (LRU) [52-54, 56], we also use LFU or LRU as the base of cache management of CS in the proposed method. The detail of cache management mechanism is as follows:

- Each CCN router manages its own CS according to LFU or LRU algorithm. The contents in the CS are always sorted by access frequency or last access time, respectively.

- Each CCN router chooses the contents in CS in the order of LFU/LRU rank so that the total size is within K. Note that K is a parameter for determining the amount of cached contents to be shared. Then the router advertises the changes of sharing contents. Once the CCN router advertises the contents, it does not remove the advertised contents from its CS until the next advertisement is completed.

- When receiving an advertisement from other CCN routers, the CCN router removes the advertised contents if the contents exist in own CS. When the CCN router also has advertised the same content, it keeps or removes the content, according to the hash values of the combination of the content name and the router name. This hush-based decision maintains the uniqueness of cached content holder among all corresponding CCN routers. The CCN router whose hash value of the name larger than the other's keeps the content, and the other router removes it.

- When CCN Data packets traverse a CCN router, it caches the Data packet in the normal CCN behavior. In addition to the such basic behavior, in the proposed method, the CCN router checks cache sharing status and does not cache the Data packet when it is already cached in other cooperating CCN routers.

By the above mechanisms, we can avoid the overlap of the cached content among cooperating routers, that results in the efficient usage of cache memory and the improvement of cache hit ratio. Also, we can maintain the consistency of cached contents among cooperative CCN routers. It means

content A	Router B1
content B	Router B2
content C	Router A1
⋮	⋮

Figure 4.7: Sharing Content Table (SCT) in the proposed method

that the proposed method enables to avoid cache miss completely, then the problems described in Subsections 4.3.1 and 4.3.2 are overcome.

4.4.4 Packet forwarding according to advertised information

Each CCN router keeps a list of advertised contents with names of CCN routers that are the sources of corresponding advertisements. For that purpose, we introduce a *Sharing Content Table* (SCT) as depicted in Figure 4.7.

Each CCN router handles an incoming Interest packet as follows:

1. According to the normal behavior in CCN, the CCN router looks up the content requested by the Interest packet in own CS. If CS has the requested content, the router replies it.

2. When the requested content does not exist in own CS, the router looks up own SCT for the requested content. If the corresponding entry is found, the router transfers the Interest packet to the router described in the SCT entry.

3. Otherwise, the router transfers the Interest packet by the normal forwarding behavior in CCN.

4.4.5 Duplication of cached contents

As described in the above subsection, the proposed method avoids the overlap of the cached contents in CCN routers in the cooperating ISPs. However, for a certain ISP, when there are many requests from its customer endhosts for a content cached at CCN router in a partner ISP, a CCN router in the ISP should cache the content for avoiding such requests from traversing the peering link between

cooperating ISPs. We call this behavior as cached content duplication. The pros and cons of the cached content duplication are summarized as follows:

Pros - Because the duplicated contents can be replied to the end users from the inside of ISP, it enables the ISP to provide the shorter response to the end users. Furthermore, it can reduce inter-ISP network traffic for the duplicated contents.

Cons - The total number of unique contents in shared cache decreases, which leads the decrease in the cache hit ratio.

The balance of pros and cons is determined by the content request frequency from end users. Therefore, the ISP decides contents to be duplicated as follows. Note that we assume that request frequencies of cached contents from end users in the ISP are given. Denoting the request frequency for a content c from end users in the ISP as $p(c)$, the ISP duplicates the content c when the following condition is satisfied:

$$p(c) > P_{dup} \tag{4.1}$$

where P_{dup} represents a threshold for determining whether or not duplicate the content. After deciding the content to be duplicated, the edge router of the ISP informs the content to the partner ISP. This is required for ISPs under cooperation recognize the duplication of cached contents.

On the other hand, the ISP cancels the duplication of the content c when the following condition is satisfied:

$$p(c) \leq P_{dup} \tag{4.2}$$

Then, the edge router informs the removal of the content to the edge router of partner ISP.

4.4.6 Balancing traffic between ISPs

One of possible situations when using the above mentioned mechanisms is that the traffic between two cooperating ISPs becomes unbalanced due to the differences in request frequencies for contents cached in each ISP. The unbalance of network traffic is serious problem for ISPs even when they are interconnected by a peering link. Therefore, in the proposed method, we maintain the traffic

balance between ISPs by regulating the number of contents to be advertised to the partner ISP for cache sharing. The pros and cons for the ISP when increasing the amount of advertised contents are as follows:

Pros - Cache hit ratio becomes increased since the increase in the number of advertised contents means the increase of the shared cache size. It will decrease the transit cost of both ISPs.

Cons - The traffic between the ISPs increases because the number of contents handled by the ISPs becomes large. It affects the cost of a link interconnecting the ISPs.

The amount of network traffic for cache sharing depends on the access frequency to each shared content cached at both ISPs. Therefore, in the proposed method, when balancing the network traffic, each ISP selects the contents to be shared according to request frequencies both from the ISP and partner ISP. The detailed algorithm is explained as follows.

We assume that the access frequencies of contents from an ISP and a partner ISP are separately monitored by both ISPs. For increasing the number of shared contents, the following process is conducted at the edge routers of cooperating ISPs. In what follows, we assume ISPs A and B are under cooperation and ISP A initializes the process. Note that the following process is activated at regular intervals of T_{inter}. Three parameters are utilized: P_{sum} is the amount of contents to add to the sharing at once, Δ_P is a value for tuning P_{sum}, and α is a parameter to decide the acceptable difference of access frequency between the ISPs.

1. ISP A chooses candidate contents from the cached, but not shared contents so that their total access frequency falls with the range $P_{sum} \pm \alpha$ and inform the content names to ISP B.

2. When ISP B receives the content names from ISP A, ISP B also select candidate contents from the cached, but not shared contents so that the total access frequency becomes $P_{sum} \pm \alpha$, and inform the content names to ISP A. When ISP B cannot prepare such contents because there are no content set to meet the condition, ISP B sends the message to ISP A to deny the negotiation.

3. When the exchange of content names is successfully completed, both ISPs A and B advertise

the increased content names to be shared to CCN routers in each ISP and they finish the process.

4. When ISP A receives the denial message, ISP A decrease the value of P_{sum} by Δ_P and restart the negotiation (return to step 1).

On the other hand, when the difference in access frequencies of both directions, denoted by P_{diff}, becomes larger than P_{th}, the ISP having larger access frequency from the other ISP initializes the following process. Note that in what follows we assume that ISP A initializes the process to ISP B.

1. ISP A chooses candidate contents from the cached and shared contents between the ISPs so that their total access frequency falls with the range $P_{diff} \pm \alpha$, and advertises the withdraw of the selected contents to ISP B.

2. ISP B forwards the withdrawn advertisement messages to CCN routers in ISP B's network.

4.4.7 Suppression of advertisement message

To suppress advertisement messages, we define the area in which the proposed method is applied, that is H hop from each edge router. That is, CCN routers within H hops from the edge router follow the behavior of proposed method and other routers outside the area behave as normal CCN routers. $H = \infty$ means that we apply the proposed method to all CCN routers. The area limitation with H hops is for suppressing volume of content advertisement messages, as well as reducing traffic inside each ISP for cache sharing. When we set H to larger value, the memory space for sharing cached contents becomes larger since larger number of CCN routers is involved in cache sharing. At the same time, a larger amount of advertisement messages is generated.

4.5 Evaluation

In this section, we show the evaluation results of the proposed method described in Section 4.4, assuming that two ISPs interconnected by a peering link adopt the proposed method.

4.5.1 Evaluation environment

Network topology

To construct ISP topologies for the evaluation, we utilized the backbone network topologies of commercial ISPs from CAIDA's database [57]. For the evaluation, we extract AT&T backbone network topology from the dataset. The number of nodes and links in AT&T topology are 84 and 124, respectively. Although the dataset includes the link capacity values between nodes, the date of the data is 06/07/2000, which is too old to evaluate the performance of CCN and its modification. To regulate the link capacity, we referred the up-to-date capacity information from IIJ Web site [58], which is a Japanese commercial ISP. In detail, we set 20 Gbps to the link which has the maximum capacity in AT&T data, and set the bandwidth of other links proportionally to their original capacities. We also assume that the propagation delays of all links are 10 ms.

Other settings

We regard the network topology determined in Subsection 4.5.1 as a single ISP's topology. For the evaluation, we assume that two ISPs with the same topology interconnect with each other by a peering link. They also have transit links for ensuring connectivity to whole part of the Internet.

We adopt the shortest path routing between all CCN router pairs. We assume that each CCN router has full routing entries to forward the Interest packet for all contents in the network. We also assume the processing delays at CCN routers are negligibly small compared to the link propagation delay.

The distribution of request frequencies to contents follows the Zipf distribution [59]. The content size follows a uniform distribution in the range of [100 KB, 100 MB]. In each ISP, a request for a content is generated every 80 ms from an end user connected at randomly selected CCN router, referring to the survey results on end-to-end traffic flow in [49]. In other words, in each ISP, one Interest packet arrives at one of CCN routers every 80 ms. The memory space for CS at each CCN router is 10 GB and LFU is utilized for cache replacement algorithm.

We use the bit rate of the traffic on the transit link as evaluation metric since the request for the content that is not cache-hit and corresponding content data traverse the transit link. It means that

Table 4.1: Parameters for the evaluation

T_{intra}	1,000 ms
T_{inter}	1,000 ms
K	8 GB
H	∞
P_{dup}	0.01
P_{sum}	0.1
P_{th}	0.1
α	0.05
Δ_P	0.01

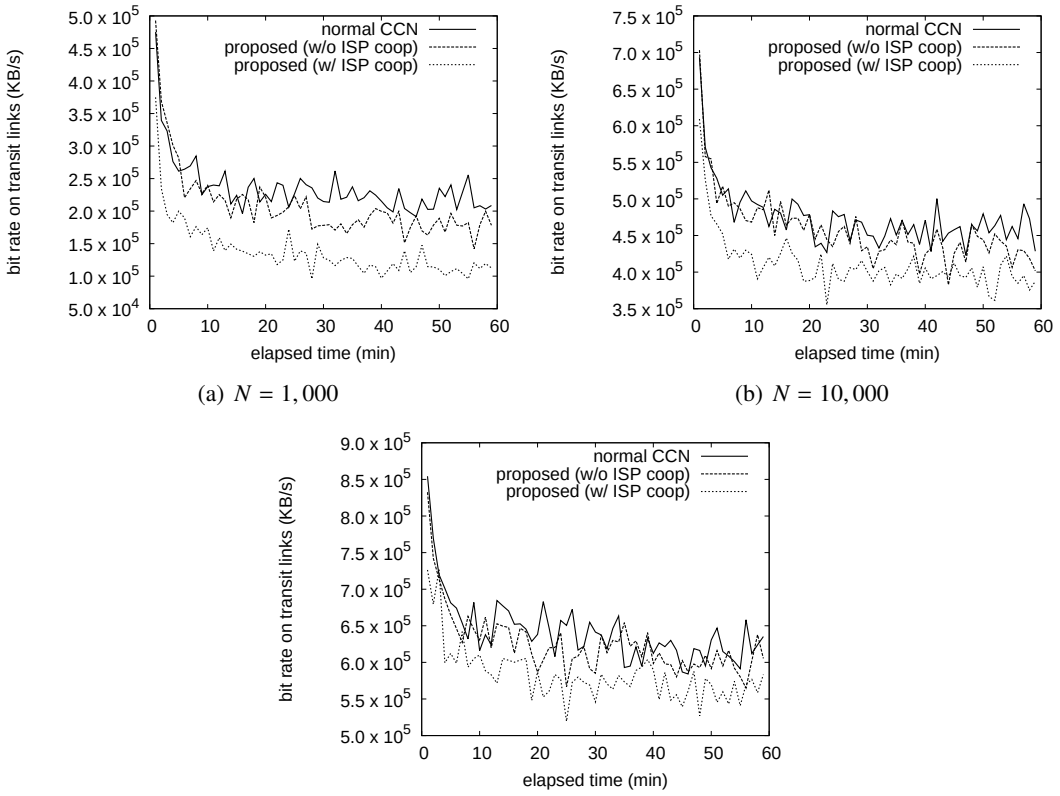
when the bit rate is smaller, the performance of content cache mechanism is better. We initialize all CSs in CCN routers as empty when starting simulation experiment. Simulation time for experiment is 60 minutes. By comparing the performance with the proposed method and with the normal CCN, we assess the effectiveness of the proposed method. We also compare the performance of the proposed method with and without inter-ISP cache sharing to confirm the advantage of inter-ISP cache sharing.

The parameter settings for proposed method are summarized in Table 4.1. Note that the performance of the proposed method is not so sensitive to parameter values.

4.5.2 Evaluation results

Figures 4.8 shows the total bit rate of traffic on transit links of two ISPs, varying the number of kinds of contents N to 1,000, 10,000, and 100,000. The x-axis of these graphs is elapsed simulation time. For all results, we can observe that at the beginning of the simulation, the transit bit rate is high, and it decreases as the simulation progressed. This is because the simulation experiment starts with empty cache and the number of cached contents increases as the simulation progresses.

In the case $N = 1,000$ (Figure 4.8(a)), we can confirm that the transit bit rate decreases largely by introducing the proposed method, even when we do not utilize the inter-ISP cache sharing. This means that only intra-ISP cache sharing has a significant contribution on decreasing the transit cost for an ISP. When we use the inter-ISP cache sharing, the degree of reduction is further advanced. Comparing the average bit rate during the latter half of the simulation experiment, the proposed



(a) $N = 1,000$ (b) $N = 10,000$

(c) $N = 100,000$

Figure 4.8: Throughput on the transit links

method can decrease the transit traffic by 45% and 19% with and without inter-ISP cache sharing, respectively. When $N = 10,000$, the reduction ratios are 13% and 5%, and that of the case when $N = 100,000$ are 9% and 3%. This means that the performance gain by the proposed method becomes small when the number of contents increases. This means that the cache hit ratio cannot be increased largely by cache sharing since end users' requests are distributed to larger number of unique contents. In such situations, increasing the size of CS, fixed to 10 GB for the evaluation, is possible way to decrease the transit traffic. Note that when increasing the size of CS, we can expect the size for cache sharing can be also increased, which leads to the enhancement of the performance of the proposed method. The investigation of the relationships between the CS size and the size for cache sharing is one of our important future works.

4.6 Conclusion

In this chapter, we proposed a method of cooperative cache sharing among CCN routers in multiple ISPs. The main idea of the proposed method is that we relax the limitation of the current CCN architecture: the cache only on the route to the original content holder is utilized for content access. We extended the cache sharing mechanisms to inter-ISP sharing with additional mechanisms to balance the network traffic between cooperating ISPs. By the evaluation assuming actual commercial ISPs adopt the proposed method, we confirmed the additional reduction by up to 45%, compared with that of normal CCN.

Chapter 5

Conclusions

The current Internet has the cost structure constructed by a numerous ISPs, each of which routes network traffic according to its own policy. On the other hand, there are new traffic routing mechanisms called as application-level routing and content-centric networking, which are not consider the ISPs' routing policy directly. Consequently, when the new mechanisms are spread widely, it is no doubt that the ISPs' cost structure will be significantly impacted. In this thesis, we presented the researches on these new traffic routing mechanisms from the aspect of the ISPs cost structure and proposed the methods comfortable to ISPs.

In Chapter 2, we proposed a method to reduce transit cost generated by application-level routing conducted by individual end user. We first constructed the estimation method of transit cost, which are calculated by end-to-end network performance easily measured by end users. Utilizing the estimation results, the proposed method chooses application-level routes under the consideration of the objectives both of ISPs and end users. By the evaluation assuming the PlanetLab and Japanese commercial network environment using the measurement results of network performance, we confirmed the proposed method achieves the considerable reduction on transit cost while maintaining the improvement brought by the application-level routing. At the same time, we also showed the ability to control the application-level routing from the both standpoints of ISPs and end users. By the proposed method, individual end users can control their application-level route selections to be comfortable to ISPs only with the simple calculation for estimating the transit cost, while improving

the own network performances.

While we assumed that the method in Chapter 2 is conducted by individual users independently, in Chapter 3, we proposed the application-level routing method conducted by the operators in multiple ISPs in the coordinated manner based on distributed simulated annealing. For the first phase to construct the proposed method, we formulated the application-level routing and defined the optimization problem for application-level route selection. After that, we proposed a method to select application-level routes based on the distributed simulated annealing, which obtains a near-optimal solution to the problem and utilize the solution to route selection. We evaluated the proposed method assuming the PlanetLab nodes perform application-level routing. The evaluation results showed that the proposed method can avoid overlaps and overload on application-level links. Although the method in Chapter 3 needs some processes that are exchanging information among application-level nodes and estimating performance to choose application-level routes, the method can achieve higher network performance, compared to the method in Chapter 2. Therefore, when the operators of application-level routing have enough computing resource and can exchange information with other operators, the proposed method in Chapter 3 is relatively appropriate.

In Chapter 4, we focused on content-centric networking that largely affects the ISPs' network operation cost. Although in-network caching mechanism of CCN has a positive effect to reduce transit cost, there is a potential to reduce further amount of transit cost. Then, we proposed a cache sharing method for the in-network caching of CCN to achieve further reduction of transit cost by exploiting peering links between ISPs. The method avoids overlaps of cached contents at CCN routers in multiple ISPs, as well as includes additional mechanisms to balance the network traffic between cooperating ISPs. We confirmed the additional reduction in transit cost by the proposed method, compared to the normal CCN behavior.

In the future, for the application-level routing, there is a challenge on how to spread the proposed methods to operators of application-level routing. In particular, as described above, the method in Chapter 3 needs the processes to exchange information and calculate a near-optimal solutions. Therefore, if end users do not want to follow these process, the proposed method cannot be activated. Then, the following situation has more reality: ISPs conducts the proposed method and provide the calculated routes to end users. In such situation, the ISPs may tend not to exchange

whole network information each other to avoid security risk by exposing own network information. Also when cloud network service providers adopt the proposed method in cooperation, they may refuse to provide the detailed information of cloud networks. Therefore, a methodology may be required to maintain the performance under limited information on measurement results of network performance, network topology, and other nodes' route selections.

The next step of the proposed cache sharing method in Chapter 4 is to ensure the scalability and manageability. The advertisement of content names by the flooding mechanism is not scalable to the number of CCN routes. Therefore, we need to investigate a scalable mechanism such as hierarchical clustering mechanism. We will also examine performance bottleneck of CCN with the proposed method and develop a method to avoid it.

As long as the Internet is built up by multiple competing and cooperating ISPs, any traffic routing mechanism ignoring ISPs' cost is not acceptable. The proposed methods in this thesis can provide a reasonable scenario to utilize application-level traffic routing and CCN continuously. We believe that the researches in this thesis, which are the considerations and evaluations on the new types of traffic routing from the aspect of ISPs' cost, will contribute the further development of the Internet.

Bibliography

[1] K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "A method to reduce inter-isp transit cost caused by overlay routing based on end-to-end network measurement," to appear in *IEICE Transactions on Information and Systems*, vol. E96-D, no. 2, Feb. 2013.

[2] K. Matsuda, G. Hasegawa, and M. Murata, "Decreasing ISP transit cost in overlay routing based on multiple regression analysis," in *Proceedings of International Conference on Information Networking (ICOIN) 2010*, Jan. 2010.

[3] 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 *14th International Telecommunications Network Strategy and Planning Symposium (NETWORKS 2010)*, pp. 250–255, Sept. 2010.

[4] Kazuhito Matsuda and Go Hasegawa and Masayuki Murata, "Decreasing ISP transit cost in overlay routing based on multiple regression analysis," *Technical Report of IEICE* (ICM2009-25), vol. 109, no. 120 pp. 67–72, July 2009. (in Japanese)

[5] Kazuhito Matsuda and Go Hasegawa and Satoshi Kamei and Masayuki Murata, "Reducing inter-ISP transit cost caused by overlay routing," *Technical Report of IEICE* (NS2010-17), vol. 110, no. 39, pp. 7–12, May 2010. (in Japanese)

[6] K. Matsuda, G. Hasegawa, and M. Murata, "An application-level routing method with transit cost reduction based on a distributed heuristic algorithm," submitted to *IEICE Transactions on Communications* [conditionally accepted], July 2012.

[7] K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "Centralized and distributed heuristic algorithms for application-level traffic routing," in *Proceedings of International Conference on Information Networking (ICOIN) 2012*, Feb. 2012.

[8] K. Matsuda, G. Hasegawa, S. Kamei, and M. Murata, "An application-level routing method for improving end-to-end network performance based on heuristic algorithm," *Technical Report of IEICE* (NS2011-65), vol. 111, no. 196, pp. 23–28, Sept. 2011. (in Japanese)

[9] K. Matsuda, G. Hasegawa, and M. Murata, "A dynamic application-level routing method reacting traffic changes based on distributed heuristic algorithm," *Technical Report of IEICE* (NS2012-36), vol. 112, no. 134, pp. 19–24, July 2012. (in Japanese)

[10] K. Matsuda, G. Hasegawa, and M. Murata, "Cooperative cache sharing among ISPs for additional reduction in inter-ISP transit cost in content-centric networking," submitted to *IFIP Networking 2013*, May 2013.

[11] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking named content," in *Proceedings of CoNEXT2009*, pp. 1–12, Dec. 2009.

[12] L. Gao, "On inferring autonomous system relationships in the Internet," *IEEE/ACM Transscations on Networking*, vol. 9, no. 6, pp. 733–745, 2001.

[13] X. Dimitropoulos, D. Krioukov, M. Fomenkov, B. Huffaker, Y. Hyun, K. Claffy, and G. Riley, "AS relationships: Inference and validation," *SIGCOMM Computer Communication Review*, vol. 37, pp. 29–40, Jan. 2007.

[14] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proceedings of SOSP 2001*, Oct. 2001.

[15] S.-J. Lee, S. Banerjee, P. Sharma, P. Yalagandula, and S. Basu, "Bandwidth-aware routing in overlay networks," in *Proceedings of INFOCOM 2008*, pp. 1732–1740, Apr. 2008.

[16] G. Smaragdakis, V. Lekakis, N. Laoutaris, A. Bestavros, J. W. Byers, and M. Roussopoulos, "EGOIST: overlay routing using selfish neighbor selection," in *Proceedings of CoNEXT 2008*, no. 6, pp. 1–12, Dec. 2008.

[17] Z. Li and P. Mohapatra, "QRON: QoS-aware routing in overlay networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, pp. 29–40, Jan. 2004.

[18] L. Zhe and S. Gwendal, "Time-shifted TV in content centric networks: the case for cooperative in-network caching," in *Proceedings of ICC 2011*, pp. 1–6, June 2011.

[19] I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in *Proceedings of the second edition of the ICN workshop on Information-centric networking*, pp. 55–60, Feb. 2012.

[20] S. Guo, H. Xie, and G. Shi, "Collaborative forwarding and caching in content centric networks," in *Proceedings of the 11th international IFIP TC 6 conference on Networking - Volume Part I*, pp. 41–55, May 2012.

[21] D. Perino and M. Varvello, "A reality check for content centric networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pp. 44–49, Aug. 2011.

[22] S. DiBenedetto, C. Papadopoulos, and D. Massey, "Routing policies in named data networking," in *Proceedings of the ACM SIGCOMM workshop on Information-centric networking*, pp. 38–43, Aug. 2011.

[23] PlanetLab web site. available at `http://www.planet-lab.org/`.

[24] 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.

[25] D. G. Andersen, A. C. Snoeren, and H. Balakrishnan, "Best-path vs. multi-path overlay routing," in *Proceedings of IMC 2003*, Oct. 2003.

[26] 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.

[27] P. Rodriguez, S.-M. Tan, and C. Gkantsidis, "On the feasibility of commercial, legal P2P content distribution," *SIGCOMM Computer Communication Review*, vol. 36, pp. 75–78, Jan. 2006.

[28] S. Seetharaman and M. Ammar, "Characterizing and mitigating inter-domain policy violations in overlay routes," in *Proceedings of ICNP 2006*, pp. 259–268, Nov. 2006.

[29] T. Karagiannis, P. Rodriguez, and K. Papagiannaki, "Should Internet service providers fear peer-assisted content distribution?," in *Proceedings of IMC 2005*, pp. 6–6, Oct. 2005.

[30] IETF ALTO Working Group web site. available at `http://datatracker.ietf.org/wg/alto/`.

[31] H. Xie, Y. R. Yang, A. Krishnamurthy, Y. G. Liu, and A. Silberschatz, "P4P: Provider portal for applications," *SIGCOMM Computer Communication Review*, vol. 38, pp. 351–362, Oct. 2008.

[32] 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.

[33] M. Kamel, C. Scoglio, and T. Easton, "Optimal topology design for overlay networks," in *Proceedings of NETWORKING 2007*, pp. 714–725, May 2007.

[34] R. Cohen and D. Raz, "Cost effective resource allocation of overlay routing relay nodes," in *Proceedings of INFOCOM 2011*, 2011.

[35] University of California CAIDA. available at `http://www.caida.org/home/`.

[36] University of Oregon Route Views Project. available at `http://www.routeviews.org/`.

[37] Hewlett-Packard Laboratories Scalable Sensing Service. available at `http://networking.hpl.hp.com/s-cube/`.

[38] P. Sharma, Z. Xu, S. Banerjee, and S.-J. Lee, "Estimating network proximity and latency," *SIGCOMM Computer Communication Review*, vol. 36, pp. 39–50, July 2006.

[39] J. Strauss, D. Katabi, and F. Kaashoek, "A measurement study of available bandwidth estimation tools," in *Proceedings of IMC 2003*, pp. 39–44, Oct. 2003.

[40] IANA, "IANA AS Numbers assignment data page." available at `http://www.iana.org/assignments/as-numbers/`.

[41] S. Ryuta, H. Go, and M. Masayuki, "Performance evaluation of spatial composition method of measurement results in overlay networks," *IEICE technical report. Information networks*, vol. 110, pp. 217–222, Feb. 2011.

[42] 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.

[43] 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.

[44] 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.

[45] 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 *Proceedings of NETWORKS 2010*, pp. 250–255, Sept. 2010.

[46] M. Arshad and M. C. Silaghi, "Distributed simulated annealing and comparison to DSA," in *Proceedings of the Fourth Workshop on DCR*, Aug. 2003.

[47] J. Hromkovic, *Algorighmics for Hard Problems*. Springer, 2005.

[48] X. R. Wu and A. A. Chien, "A distributed algorithm for max-min bandwidth sharing," tech. rep., University of California, San Diego, 2006.

[49] M. Pustisek, I. Humar, and J. Bester, "Empirical analysis and modeling of peer-to-peer traffic flows," in *Proceedings of MELECON2008*, pp. 169–175, May 2008.

[50] G. Dan, "Cache-to-Cache: Could ISPs cooperate to decrease peer-to-peer content distribution costs?," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, pp. 1469 –1482, Sept. 2011.

[51] M. Hefeeda and B. Noorizadeh, "On the benefits of cooperative proxy caching for peer-to-peer traffic," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, pp. 998 –1010, July 2010.

[52] D. Wessels and K. Claffy, "ICP and the Squid Web cache," *IEEE Journal on Selected Areas in Communications*, vol. 16, pp. 345–357, Apr. 1998.

[53] L. Fan, P. Cao, J. Almeida, and A. Broder, "Summary cache: a scalable wide-area Web cache sharing protocol," *IEEE/ACM Transactions on Networking*, vol. 8, pp. 281–293, June 2000.

[54] D. Karger, A. Sherman, A. Berkheimer, B. Bogstad, R. Dhanidina, K. Iwamoto, B. Kim, L. Matkins, and Y. Yerushalmi, "Web caching with consistent hashing," *Computer Networks*, vol. 31, pp. 1203–1213, May 1999.

[55] L. Wang, A. K. M. M. Hoque, C. Yi, A. Alyyan, and B. Zhang, "OSPFN: An OSPF based routing protocol for Named Data Networking," Tech. Rep. NDN-0003, NDN Technical Report, July 2012.

[56] S. Romano and H. ElAarag, "A quantitative study of recency and frequency based web cache replacement strategies," in *Proceedings of CNS 2008*, pp. 70–78, 2008.

[57] CAIDA, "Backbone data." available at `http://www.caida.org/tools/visualization/mapnet/Data/`.

[58] Internet Initiative Japan, "IIJ backbone network." available at `http://www.iij.ad.jp/company/network/backbone/`.

[59] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: evidence and implications," in *Proceedings of INFOCOM 1999*, vol. 1, pp. 126–134, mar 1999.