Design and implementation of an IPv6 gateway allowing effective use of multihome network

Yoshiaki Hori*, Keisuke Onimaru**, Takeshi Ikenaga**, and Yuji Oie**

*Dept. of Art and Information Design, Kyushu Institute of Design, Fukuoka 815-8540 Japan
**Dept. of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka 820-8502 Japan

Abstract—We have designed and implemented an IPv6 gateway with a prefix translation mechanism and have evaluated its performance with the current IPv6 Internet. We have designed a backbone selection scheme based upon a minimum RTT (Round Trip Time), and have demonstrated its effectiveness.

I. INTRODUCTION

The Internet continues to further penetrate into our everyday life as well as impacting various social activities, and a diverse range of devices as well as common PCs will be more likely to have the capability of exchanging information on the Internet through utilization of various network services. Therefore, a new IP addressing scheme allowing end-to-end communication among a wide range of devices will be needed; IP version 6 (IPv6) can meet this requirement. IPv6 with its immense address space will further accelerate the use of multiple Internet backbones for each of user; i.e., it will lead to the development of a multihome network environment.

In this study, we focus on a multihome network environment using IPv6, and examine its performance in supporting efficient peer-to-peer communication in this context. Furthermore, we have designed and actually implemented an IPv6 gateway allowing effective use of multihome network, and evaluated its performance through experiments over the Internet.

II. MULTIPLE HOME ENVIRONMENT WITH IPv6 NETWORK

In this study, we deal with site-multihoming, which means that a site connects to two or more internet service providers (ISPs) instead of one. This is important for achieving load sharing and fault tolerance for sites.

This work was supported in part by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (A), 15200005, 2003, and in part by the Ministry of Public Management, Home Affairs, Posts and Telecommunications, Japan for Research and development for fostering younger excellent IT researchers of Strategic Information and Communications R&D Promotion Scheme.

Yoshiaki Hori is with Dept. of Art and Information Design, Kyushu Institute of Design, Fukuoka 815-8540, Japan (corresponding author to provider phone: +81 92 553 4443, fax: +81 92 553 4440; email: hori@kyushu-id.ac.jp).

Keisuke Onimaru is currently a graduate student of Graduate School of Information Science, Nara Institute of Science and Technology, Ibaraki, 630-0101 Japan (e-mail: keisu-o@is.aist-nara.ac.jp).

Takeshi Ikenaga and Yuji Oie are with Dept. of Computer Science and Electronics, Kyushu Institute of Technology, Iizuka 820-8502, Japan (e-mail: ike@cse.kyutech.ac.jp and oie@cse.kyutech.ac.jp, respectively).

In IPv6 networks, IPv6 global unicast address space is assigned to users as an address prefix, i.e., 3ffe:501:2c48::/48, by the ISP using a layered delegation scheme. This scheme can achieve an effective use of global unicast address space by means of enabling prefix aggregation within each ISP network. Nevertheless, the address prefix for a user depends on the ISP’s backbone topologies, so that each user has to use multiple address prefixes effectively by themselves if they are connected with two or more Internet backbones.

In an end-to-end oriented network such as the Internet, an ideal multihome architecture should also be based on end-to-end arguments. Each end system is assigned two or more address prefixes for unicast communications, and will select an appropriate IP address from among the available ones, in some sense by itself. It will thus have to obtain sufficient information to do that in an effective manner. The ideal scheme, however, is not currently realistic because supporting end-system multihoming to update all end systems and all routers is expensive. Therefore, we propose a simple means of supporting multihoming in an IPv6 network, by designing an IPv6 gateway with an address prefix translation mechanism which allows end-systems without any modification to effectively use a site-multihome environment.

Figure 1 shows the site-multihoming configuration as a simple scheme. Site X is connected to both ISP A and ISP B via a gateway. We assume that any external links for this site are connected with a single point at the gateway. The two ISPs assign IPv6 global unicast address prefixes to site X. For example, ISP A assigns aaaa:bbbb:cccc:dddd::/48 which is a part of aaaa:bbbb::/32, which is address space managed by ISP A. In the same way, ISP B assigns yyyy:zzzz:xxxx:ww00::/56 which is a part of yyyy:zzzz::/40. Consequently, site X acquires two IPv6 address spaces, given by ISP A and ISP B. We evaluate a network address translation scheme which enables all hosts to use a site-multihome environment.

Network address translation is a well known means of solving problems such as limited address space for global communications in current IPv4 networks. One typical mechanism is known as NAT, which allows hosts in a private network to transparently access an external network and thus enable access to selected local hosts from the outside [1]. NAT, however, makes end-to-end connectivity lost if the number of

0-7803-7978-0/03/$17.00 ©2003 IEEE 1000
local hosts is less than or equal to the number of addresses in the
global address set. In contrast, IPv6 provides a sufficient
amount of address space for end hosts because the IPv6 address
architecture divides into n-bit subnet prefixes and a 128-n bit
interface ID, and its interface ID is required to be 64bits long
and to be constructed in Modified EUI-64 format for a global
unicast address [2].

Thus, we obtain a subnet prefix translation instead of a 128bit
IPv6 address. Because of the prefix translation, the complicated
translation procedure is reduced at the gateway. We apply this
scheme as seen in the network of the figure 1. Site X, at first,
configures its local network by using the longest address prefix
yyyy:zzzz:xxxx:ww00::/56 among the prefixes which
is allowed at Site X. In this case, the longest prefix, whose
length is 56bit, is assigned by ISP B. The address of host A
becomes yyyy:zzzz:xxxx:ww00::1234/128. If host A
uses ISP B, gateway G performs no address translation for
packets being forwarded from/to host A. In contrast, if host A
uses ISP A, gateway G carries out the prefix translation
yyyy:zzzz:xxxx:ww00::/56 to
aaaa:bbbb:cccc::/56 for outbound packets, and the
reverse translation for inbound packets.

By carrying out such prefix translations at the gateway; all
hosts at site X are able to utilize a site-multihoming environment
without updating the end hosts or routers.

III. SELECTING A BACKBONE NETWORK FOR IPv6 GATEWAY

We have designed and implemented a prototype IPv6
gateway for relaying what between a user network and internet
backbones in order to provide a site-multihoming environment.
The gateway functions to gather information on user traffic and
to make a decision regarding the selection of an appropriate
internet backbone on behalf of the end systems in the user
network.

On the internet, a packet forwarding route is determined by
the packet’s destination address. A router examines each packet
header and decides upon the next hop router. In most
communications using TCP as the transport protocol, the source
address is a key to determining the route of return from a peer
host. Therefore, if a host would like to select one preferred
external link, the host should determine the appropriate one
from among a set of candidate source addresses.

IV. SELECTING AN ALGORITHM FOR MULTIPLE INTERNET
BACKBONE

We have designed and implemented the selection of one of the
available multiple internet backbones, in a minimum RTT
(round trip time)-based scheme. In order to select an appropriate
external link, the collection of knowledge and information is
required. We here attempt to carry out this procedure at the
gateway because this requires minimum implementation. Our
gateway carries out the steps below.

1) collection of destination addresses and their storage
2) measurement of round trip time (RTT) via each available
   external link for the destination
3) selection of appropriate external link according to
   historical data of minimum RTT
4) translation of appropriate address prefix

V. IMPLEMENTATION

We implement the IPv6 gateway for the supporting of a
site-multihome environment based on the OpenBSD3.2
operating system. We describe the detail of its implementation
in this section.

A. Prefix translation module in kernel

The OpenBSD packet filter (pf) is the firewall package that
has been a part of the OpenBSD kernel since OpenBSD 3.0.
Two principal features of pf are the packet filter and NAT.
When pf is activated, all packets are dealt with by pf. pf carries
out filtering or address translation for packets according to the
configuration rules. In order to implement our prefix translation
(PT) mechanism, we improve the pf kernel module. We have
added some features: the translation of the IPv6 prefix, the
collection of destination addresses, and the offering of
information to the management program (pfctl) in userland.

B. Control program for the PT kernel module

pfctl is a control program for the PT kernel module running in
userland. pfctl communicates with the PT kernel module
through a special device file named /dev/pf. At the start time, pfctl reads the rules of pf from the configuration file (/etc/pf.conf), and configure them. We improve pfctl to deal with new syntax for our implementations.

C. PT Manager

ptm is a prefix translation manager running as a userland process. We implement ptm from scratch. ptm has various features: the gathering of the states of all external links, the selection of an appropriate route for a destination, and the setting of a static route in the kernel. We denote the PT route information as the set of information for each destination network as follows:

1) destination network address: ptm chooses an appropriate external link for each destination network rather than for a destination host to avoid an increase in the number of PT routes.
2) typical destination address: ptm examines the round trip time for the typical destination address.
3) network interface: ptm holds information about the network interface which connects to the selected external link.
4) last forwarding time: ptm records the timestamp for the last packet transferred.
5) last changed time: ptm records the timestamp for the last evaluation of the route.
6) counter: ptm increments this counter when it finds that the appropriate external link is a link other than the currently selected link. While the value of this counter remains small, ptm does not change the external link, in order to avoid route flapping.

ptm is invoked by the cron daemon and carries out the procedure below:

1) obtains the destination addresses which are collected by the pf module.
2) reads the current PT routes.
3) measures RTT for every PT route, which is required for updating route information.
4) updates PT route information.
5) invokes pfctl and sets new rules to the pf module.
6) executes route add command and sets new static routes.

VI. EXPERIMENTS AND EVALUATION

We have executed experiments on the prototype IPv6 gateway over the current IPv6 internet in order to evaluate the performance of our proposed scheme.

A. Network testbed

We describe here the network testbed at the Kyushu Institute of Technology, Iizuka, used for these experiments. Our network are connected with three IPv6 research network backbones in Japan which are the WIDE Internet (operated by the WIDE Project), JGNv6 (the Japan Gigabit Network), and QGPOP (operated by the Kyushu Gigapop Project). The three internet backbones offer IPv6 network address prefixes for our network. They are 3ffe:501:2c36::/48, 3ffe:516:8180::/48, and 2001:248:185::/48 from WIDE, JGNv6, and QGPOP, respectively. We assign 3ffe:516:8180:387e::/64 to our local network. This address space is a part of the prefix from JGNv6. In Fig. 2, we illustrate our network testbed. The IPv6 gateway has connected a local network to the three IPv6 network backbones. The gateway can forward packets from a local network to one of the three backbones in two ways.

B. Measuring RTT for destination network

As mentioned above, we measure the minimum RTT in order to select the appropriate external link. Our gateway measures the RTT for the destination network at regular intervals and selects the optimum network backbone automatically.

In order to select the optimum internet backbone, the gateway measures RTT by ping utility which uses the ICMP protocol’s mandatory ECHO_REQUEST datagram to elicit an ICMP ECHO_RESPONSE from a destination host or gateway. Every five minutes, the gateway carries out four measurements in the interval of one second.

We show in Table 1 how many times each IPv6 backbone was selected for each of destination hosts depending upon the results of measurements on RTT. This experiment involved 270 measurements of RTT for each destination host in February 2003. Six hosts are in Japan those listed from the top of the table, and the three remaining hosts are in England and the U.S.A.

It is evident that, in fact, the minimum RTTs depend on the IPv6 backbone, and thus the selection of the internet backbone for the destination network is an important task.
Table 1 Number of times each IPv6 backbone was selected for each of destination hosts

<table>
<thead>
<tr>
<th>Destination host</th>
<th>via WIDE</th>
<th>via JGNv6</th>
<th>via QGPOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Kddlab.6to4.jp</td>
<td>0</td>
<td>268</td>
<td>2</td>
</tr>
<tr>
<td><a href="http://www.wide.ad.jp">www.wide.ad.jp</a></td>
<td>0</td>
<td>270</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.hitachi.co.jp">www.hitachi.co.jp</a></td>
<td>0</td>
<td>270</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.imnet.ad.jp">www.imnet.ad.jp</a></td>
<td>0</td>
<td>270</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.iij.ad.jp">www.iij.ad.jp</a></td>
<td>7</td>
<td>263</td>
<td>0</td>
</tr>
<tr>
<td><a href="http://www.kame.net">www.kame.net</a></td>
<td>2</td>
<td>235</td>
<td>15</td>
</tr>
<tr>
<td>6to4.ipv6.bt.com</td>
<td>10</td>
<td>162</td>
<td>98</td>
</tr>
<tr>
<td><a href="http://www.netbsd.org">www.netbsd.org</a></td>
<td>261</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>playground.sun.com</td>
<td>258</td>
<td>0</td>
<td>12</td>
</tr>
</tbody>
</table>

Table 2 Number of times each IPv6 backbone was selected for each of destination hosts

<table>
<thead>
<tr>
<th>Destination host</th>
<th>via WIDE</th>
<th>via JGNv6</th>
<th>via QGPOP</th>
<th>Fail</th>
</tr>
</thead>
<tbody>
<tr>
<td>cernet.ipv6.net.edu.cn</td>
<td>358</td>
<td>17</td>
<td>1353</td>
<td>3</td>
</tr>
<tr>
<td>cnet-coe-psu-th.tsps.1.</td>
<td>1125</td>
<td>230</td>
<td>351</td>
<td>26</td>
</tr>
<tr>
<td>freenet6.net (thailand)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We examine other destinations in China and in Thailand as shown in Table 2. In these experiments, we carried out 1731 measurements of RTT in February 2003. They look like unstable rather than in Table 1 because some measurements are failed.

C. Strategies to change backbone automatically

We hope to select the internet backbone automatically. There is a problem which should be considered, to avoid route flapping. When we utilize our IPv6 gateway in particular, the change of the route causes the change of the network address. Currently, as is well known, most communication via the internet is severed by conventional TCP when the network addresses change. In order to eliminate this problem, we introduce two parameters in our scheme. One parameter is a counter which increments whenever RTT is measured. A route is fixed until the counter reaches the threshold. The other parameter is \( rtt\_th \). When the measured value of RTT does not exceed \( rtt\_th \), it is regarded as meaningless. In other words, these parameters prevent route flapping for a short period. Figure 3 shows a number of changing routes and Fig. 4 shows the difference between the minimum RTT of the measured ones and RTT via a current backbone for cernet.ipv6.net.edu.cn. According to the results, in the case of \( rtt\_th \) exceeding three, the number of route changes is reduced and does not depend on \( rtt\_th \). On the other hand, the difference between minimum RTT and current RTT increases as \( rtt\_th \) increases. Therefore, there is an optimal value of \( rtt\_th \) for achieving the reduction of route flapping as well as the efficient use of this scheme.

Through our experiments, we have achieved a minimum RTT without any adverse effects due to route flapping caused by the change of network address prefixes.

VII. CONCLUDING REMARKS

In this research, we have designed and implemented an IPv6 gateway with a prefix translation mechanism and have evaluated its performance on the current IPv6 internet. We have designed a backbone selection scheme based upon measurements of the minimum RTT, and demonstrated its effectiveness. This work is a first step toward the practical and effective use of a site-multihome environment supporting peer-to-peer communications, and further extensive research will be required. By use of the data, the gateway can select an appropriate internet backbone to attain high performance as well as provide connectivity with fault tolerance for sites.

REFERENCES