PIA-SM: multicast based IPv6 anycast routing protocol

Satoshi Matsunaga

Graduate School of Information Science and Technology,
Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail: s-matung@anarg.jp

Shingo Ata*

Graduate School of Engineering, Osaka City University,
3-3-138 Sugimoto, Sumiyoshi-ku, Osaka 558-8585, Japan
E-mail: ata@info.eng.osaka-cu.ac.jp
*Corresponding author

Hiroshi Kitamura

Solution Development Laboratories, NEC Corporation,
2-11-5 Shibaura, Minato-Ku, Tokyo 108-8557, Japan
E-mail: kitamura@da.jp.nec.com

Masayuki Murata

Graduate School of Information Science and Technology,
Osaka University, 1-5 Yamadaoka, Suita, Osaka 565-0871, Japan
E-mail: murata@ist.osaka-u.ac.jp

Abstract: Today, the use of anycast address is quite limited. One of the reasons is that there is no routing protocol providing a global anycasting service. In this paper we design and implement a new anycast routing protocol called PIA-SM (Protocol Independent Anycast – Sparse Mode). We focus on PIM-SM (Protocol Independent Multicast – Sparse Mode), which is one of multicast routing protocols available now, to develop an anycast routing protocol, because anycast and multicast have many similar properties. We modified PIM-SM based on differences between multicast and anycast. We next describe the technical issues to be solved in the implementation of PIA-SM. We also show some experimental results to demonstrate PIA-SM, and verify that PIA-SM enables routers to forward an anycast packet to an appropriate node of multiple candidate nodes.

Keywords: IPv6; anycast; routing protocol; PIA-SM.

Reference to this paper should be made as follows: Matsunaga, S., Ata, S., Kitamura, H. and Murata, M. (2006) 'PIA-SM: multicast based IPv6 anycast routing protocol', *Int. J. Internet Protocol Technology*, Vol. 1, No. 3, pp.189–197.

Biographical notes: Satoshi Matsunaga received his BE degree in Information Science and Technology from Osaka University in 2003. He is currently a Graduate Student at the Graduate School of Information Science and Technology, Osaka University.

Shingo Ata is a Lecturer in the Graduate School of Engineering at Osaka City University, Japan. He received his ME and PhD Degrees in Informatics and Mathematical Science from Osaka University in 1998 and 2000, respectively. His research includes design of communication protocols and performance modelling of communication networks.

Hiroshi Kitamura works at NEC Corporation since 1990. He has also been working as a Visiting Associate Professor at the University of Electro-Communication since 2004. He received his BS and MS Degrees from Nagoya University in 1988 and 1990, respectively. He also received a PhD degree in Informatics and Mathematical Science from Osaka University in 2002. He has been engaged in research and development of internet protocols. He currently focuses on research and development of IPv6, Mobile IPv6, Plug and Play, and Security.

Masayuki Murata is a Professor in the Graduate School of Information Science and Technology at Osaka University, Japan. He received his ME and DE Degrees in Information and Computer Sciences from Osaka University, Japan, in 1984 and 1988, respectively. In April 1984 he joined the Tokyo Research Laboratory, IBM Japan, as a Researcher. In 1987 he moved to Osaka University. He has more than 300 papers in international and domestic journals and conferences. His research interests include computer communication network architecture and performance modelling and evaluation.

1 Introduction

Anycast is a new networking paradigm supporting service oriented addresses. In anycast, an identical address can be assigned to multiple nodes providing a specific service. An anycast packet (i.e., one that has an anycast address as its destination) is delivered to one of nodes, corresponding to the packet's destination address. Anycast was first defined in RFC1546 (Patridge et al., 1993) stating that the motivation for anycast is to considerably simplify the task of finding an appropriate server within the internet. The basic idea behind anycast communication is to separate the logical service identifier from the physical host equipment. That is, the anycast address is assigned on a type of service basis so that the network service acts as a logical host.

In the Internet Protocol version 6 Specification (Deering and Hinden, 1998), the anycast address is defined. The addressing architecture (Hinden and Deering, 2003) of IPv6 has two other types of IP addresses; unicast and multicast. Table 1 summarises the communication forms for these addresses. A unicast address is a unique identifier for each network interface, and multiple interfaces must not be assigned the same unicast address. Packets with the same destination address are sent to the same node. A multicast address, on the other hand, is assigned to a group of nodes, i.e., all group members have the same multicast address. Packets for this address are sent to all members simultaneously. Like a multicast address, a single anycast address can be assigned to multiple nodes (called *anycast membership*), but unlike multicasting, only one member of the assigned anycast address communicates with the originator at a time.

Table 1 IPv6 address types

	Unicast	*Multicast*	*Anycast*
Number of membership	Single	Multiple	Multiple
Communication form	Point to point	Point to multipoint	Point to point
Address space	Except multicast address	Special address space	Shared with unicast

However, the use of IPv6 anycast is currently limited. It is because there are many technical issues to be resolved in the current IPv6 anycast specification (Hagino and Ettikan, 2003). One of the important problems is that there is no routing protocol for anycasting. The router should have an active role in deciding the destination network/node so that anycast packets can be appropriately forwarded. We need to design and implement a routing protocol suited to anycast applications for realising a global anycasting service.

Doi et al. (2003, 2004) show the way of designing routing protocols for IPv6 anycast by some modifications of existing multicast routing protocols. These are based on several similarities between anycast and multicast communications. However they do not discuss issues the in implementation of these protocols. Therefore we need to verify if we can realise the anycast routing protocol by the proposed method.

In addition, we need to evaluate three anycast routing protocols and to analyse where each anycast routing protocol is suitable to be used when we consider standardisation of anycast routing protocols in IPv6. Therefore, we should implement three anycast routing protocols to verify that they can be used for anycast communication on the real IPv6 network. In this paper, we design and implement one of the anycast routing protocols proposed in Doi et al. (2003, 2004).

In this paper we give a specific design of PIA-SM (Protocol Independent Anycast – Sparse Mode), which is first defined in Doi et al. (2003, 2004). Three types of multicast routing protocols are available today, and each multicast protocol has both advantages and disadvantages. PIM-SM (Protocol Independent Multicast – Sparse Mode), which is one of the multicast routing protocols, has an advantage in global multicasting. In other words, PIM-SM is designed to be used in the network where multicast listeners are sparsely distributed. This model is very similar to the case where anycast receivers for a single anycast address are widely spread in the internet. By focusing on this property, we define a new anycast routing protocol called PIA-SM based on the behaviour of PIM-SM. Through the implementation of PIA-SM, we describe the technical issues to be solved in PIA-SM which are unclear in the previous literatures (Doi et al., 2003). We also show some experimental results to demonstrate PIA-SM, and verify that PIA-SM enables routers to forward an anycast packet to an appropriate node of multiple candidate nodes.

This paper comprises five sections. In Section 2, we show a specific design of PIA-SM and describe the behaviour of PIA-SM. In Section 3, we describe implementation matters of PIA-SM and also give solutions for these issues. We show results to demonstrate PIA-SM, and verify the behaviour of PIA-SM in Section 4. Section 5 gives a brief conclusion.

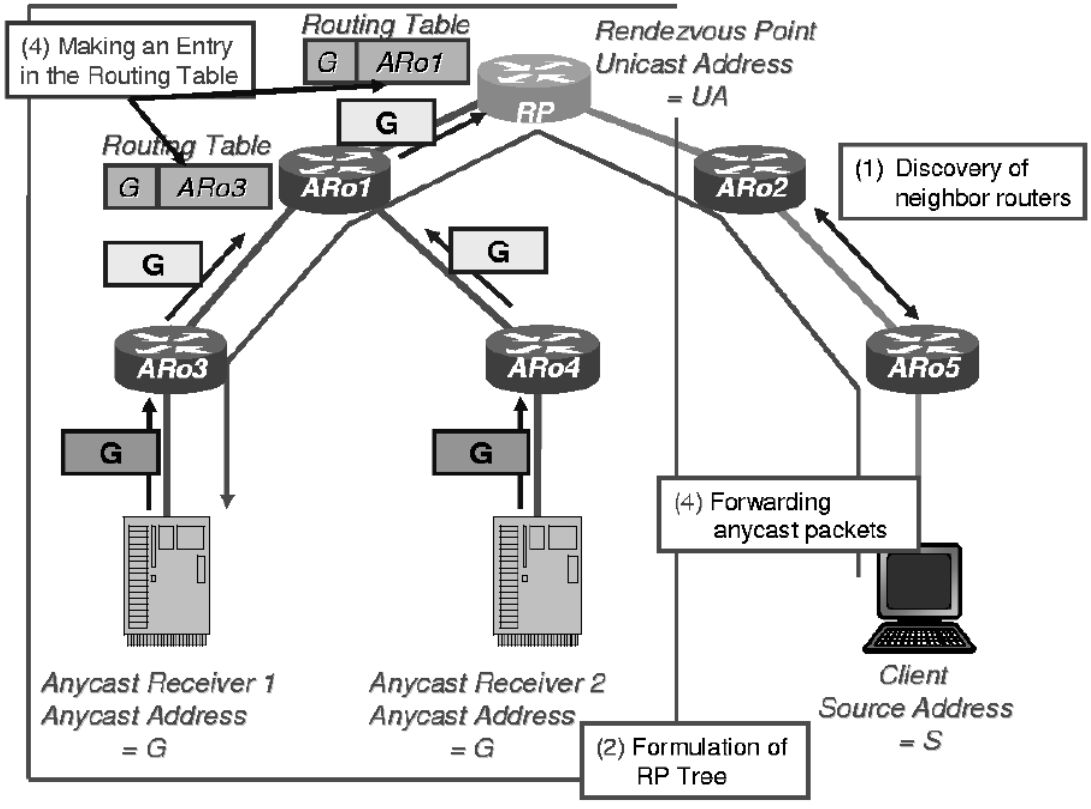
2 Design of PIA-SM

In anycast communication, an anycast packet is forwarded to the most appropriate node among memberships which have the same anycast address. As shown in Table 1 there are multiple nodes which are assigned a single address in both multicast and anycast. This similarity implies that we can use the same method to manage multiple nodes having the same address in both multicast and anycast. That is, we can use the mechanism in the role of managing multicast listeners in PIM-SM to manage nodes which have anycast addresses in PIA-SM. Here, a node assigned an anycast address is called an anycast receiver. On the other hand, anycast and multicast have some differences. First, in anycast communication only one (appropriate) anycast receiver may receive the packet addressed to the anycast address, while the multicast packet is forwarded to all of multicast listeners simultaneously. Moreover, because of the specification of the addressing architecture (Hinden and Deering, 2003) anycast and unicast addresses are syntactically indistinguishable. For these reasons, it is desirable to use the same mechanism in unicast communication for forwarding anycast packets. To conclude, we design a new anycast routing protocol which combines the mechanism of multicast routing for managing anycast receivers and the mechanism of unicast routing for packet forwarding.

Like PIM-SM, PIA-SM manages anycast receivers by composing a distribution tree for each anycast address rooted at the Rendezvous Point (RP). The RP is a router configured to be used as the root of the tree (called RPT) for the anycast address. But unlike PIM-SM, PIA-SM forwards anycast packets to only one anycast receiver. PIA routers select one anycast receiver, based on the appropriateness notified by each receiver. Such appropriateness is called *metric* in PIA-SM.

Figure 1 shows the overview of PIA-SM. In this figure we assume all of routers support PIA-SM (we refer them as PIA routers). Also, *upstream* means the direction towards the root (i.e., RP) of the RPT, and *downstream* is the direction to receivers. RPT is constructed for each anycast address. In the RPT, RP is the root node and anycast receivers are its leaves. First, each anycast receiver notifies its own metric to the neighbour PIA router which is directly connected to the anycast receiver. By receiving metrics from anycast receivers or downstream neighbour PIA routers, the PIA router compares these metrics and selects the most appropriate anycast receiver or PIA router. Note here, we define the anycast receiver which has the smallest value of the metric to be the most appropriate receiver. After selecting the appropriate node, the PIA router notifies the metric to the upstream neighbour PIA router. By notifying metrics at all of PIA routers with hop by hop basis, the RP finally recognises the most appropriate receiver for the anycast address. Following are functions needed to realise above behaviour of PIA-SM.

Figure 1 The overview of PIA-SM



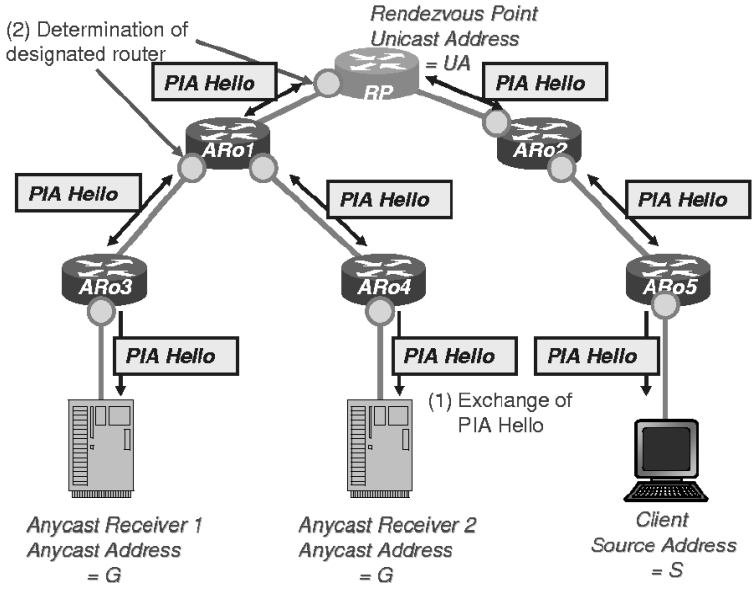
- *Discovery of neighbour PIA routers*. Each PIA router exchanges messages between neighbour routers. When a PIA router receives a message from other PIA router, the PIA router knows that there is another PIA router on the segment from where the message of the other router came.
- *Composition of RPT*. Anycast receiver notifies own assigned anycast addresses to neighbour PIA routers. When a PIA router receives a message from the anycast receiver, the PIA router knows that there is an anycast receiver on the segment from where the message came. Next, the neighbour router notifies the anycast address of the received message from the anycast receiver to the upstream router. When a PIA router receives the message from downstream routers, the PIA router also notifies the anycast address of the received message to the next upstream router. This notification is repeated by each router until the notification reaches the RP. Finally PIA routers and anycast receivers compose the RPT of each anycast address. In Figure 1, RP, ARo1, ARo3, ARo4, Anycast Receiver 1 and 2 compose the RPT of the anycast address *Any*.
- *Adding the entry for the anycast address into the unicast routing table*. Each PIA router selects one route based on received metrics. It then makes an entry of the anycast address by setting the selected route as the outgoing interface in its unicast routing table. After that, all routers can forward anycast packets as unicast packets because the anycast address is existing as the entry of unicast address in the router's unicast routing table.
- *Forwarding anycast packets*. When a client sends an anycast packet to the anycast address, the neighbour PIA router which is directly connected to the client receives the packet. This router is called the *Sender Router* (SR). The SR encapsulates the received anycast packet with the unicast address of the RP. After that, the SR sends the encapsulated packet directly to the RP by unicast routing. When the RP receives the encapsulated packet from the SR, the RP decapsulates the encapsulated packet and forwards the anycast packet to the RPT. As described above, the anycast packet is forwarded to the selected anycast receiver by the unicast routing because each PIA router has an entry for the anycast address in its unicast routing table. Finally the anycast packet is forwarded to the selected anycast receiver by unicast routing, based on PIA router's unicast routing table.

We describe the detail of each process in the following subsections.

2.1 Discovery of neighbour PIA routers

Each PIA router exchanges PIA Hello messages in order to discover other PIA routers in the same segment. The router sends a PIA Hello message by setting the broadcast address of the segment. If any PIA routers exist on the same segment, they may receive the PIA Hello message. As a result, other routers on the segment can find the router by receiving the PIA Hello. Additionally a Designated Router (DR) is defined on each segment in the same way as PIM-SM. The DR is elected based on the PIA routers' IP address such as PIM-SM. In Figure 2, circle marks show the DRs selected on each segment.

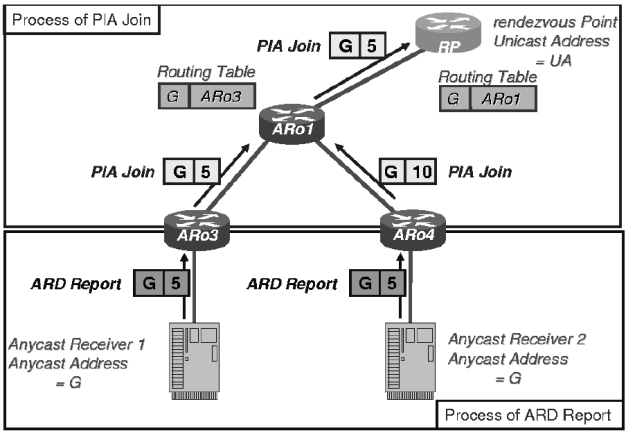
Figure 2 Discovery of neighbour routers



2.2 Composition of RPT

Figure 3 shows how PIA routers and anycast receivers compose RPT. Anycast receivers and PIA routers use ARD Report and PIA Join in order to compose the RP tree, respectively. ARD (Anycast Receiver Discovery) is a similar technique proposed in Haberman and Thaler (2002), except that we add a metric field into the message format so that PIA routers need a criterion to select the most appropriate receiver from multiple ones. ARD Report is a message that notifies the anycast receiver's address and the receiver's metric to PIA routers. On the other hand, PIA Join messages are used to notify the existence of the PIA router to other neighbour PIA routers. Exchanging ARD reports (resp. PIA Join messages) establishes a branch between an anycast receiver and PIA router (resp. two PIA routers) of the RPT. As a result PIA routers joined by two messages compose the RP tree. We describe the process of ARD Report and PIA Join.

Figure 3 Formulation of RP tree



2.2.1 Process of ARD report

ARD Report is exchanged between PIA routers and anycast receivers. If an anycast receiver is assigned the anycast address G and if the receiver wants to receive anycast packets of the address G, the receiver sends an ARD Report, including the address G to a neighbour (i.e., directly connected) PIA router. The PIA router knows that the receiver exists on the segment which the ARD Report came from. As described in Haberman and Thaler (2002), the source address of the ARD Report is the receiver's link-local address. If there are multiple anycast receivers on the same segment, we cannot identify each receiver/router by using the anycast address. The PIA router thus should not use the anycast address but link-local address of receivers or routers to identify them.

2.2.2 Process of PIA join

A PIA router which hears anycast receivers sends PIA Join to upstream PIA router including the address G of the received ARD Report. PIA Join also includes the metric of the router. The metric is the best (i.e., minimum) value among anycast receivers' metrics from downstream links. The upstream router which receives the PIA Join knows the existence of the PIA router on the segment. If the upstream router is not the RP, the router also makes a PIA Join including the address G and the router's metric, and sends it to the next upstream router. At last, all routers between anycast receivers and RP receive PIA Join of the address G, and know the receivers on the downstream link. At this time, the RPT of the anycast address G is formed by PIA routers and anycast receivers.

2.3 Adding the entry for the anycast address into the unicast routing table

Each PIA router on the RPT selects one of the downstream routes according to metrics received by ARD reports and PIA Join messages. The router selects the router/receiver whose metric is smallest among other routers and receivers on the downstream links as the next hop of the address G. The PIA router then adds an entry for the address G in its own unicast routing table. In Figure 3, metrics are shown as numbers in ARD Reports and PIA Joins. The router *ARo*1 received two PIA Joins (from *ARo*3 and *ARo*4). *ARo*1 selects the router *ARo*3 *as* the next hop because the *ARo*3's metric is smaller (5) than the one of *ARo*4 (10). If the receiver changes its metric, the receiver sends another ARD Report including the new metric to the PIA router. If the new metric is the smallest among all anycast receivers' metrics for the address G, all PIA routers between the receiver and the RP update the entry of the address G in their unicast routing tables. In the worst case, PIA routers need to update their unicast routing table whenever the router receives PIA Join or ARD Report from downstream link.

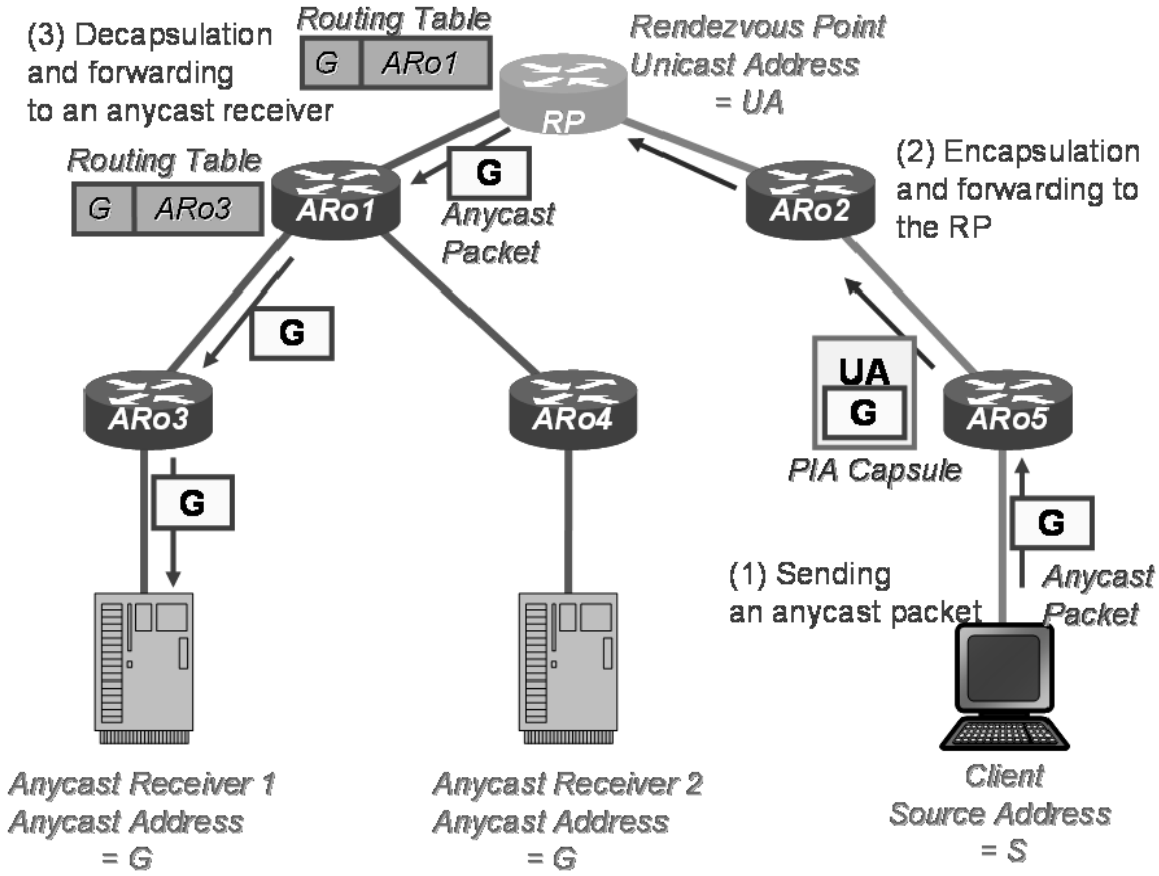
2.4 Forwarding anycast packets

The steps of forwarding anycast packets are as follows (Figure 4).

- Sending an anycast packet from a client. A client sends an anycast packet to the anycast address G.
- Capsulation of the packet and forwarding to RP. When the SR (*ARo*5 in Figure 3) receives the anycast packet, the SR searches the unicast address of the RP corresponding to the address G. If the SR finds the RP corresponding to the anycast address, the SR encapsulates it with the unicast address of the RP and sends it to upstream. The encapsulated packet is forwarded to the RP, based on unicast routing. In Figure 4, PIA router *ARo*5 encapsulates the anycast packet from the client with the RP's unicast address UA. The encapsulated anycast packet is called as PIA Capsule.

- Forwarding the packet to the selected anycast receiver. When the RP receives a PIA Capsule, the RP decapsulates it, and obtains the anycast packet. Then the RP sends out the original anycast packet. The anycast packet is forwarded from the RP to the selected anycast receiver by unicast routing because each router already has the entry of the anycast address in its own unicast routing table (it is done in Section 2.3). In Figure 4, the anycast packet passes the routers *ARo*1 and *ARo*3 and arrives at the selected Anycast Receiver 1 by the unicast routing.

Figure 4 Forwarding anycast packets



Note that PIM-SM has the cutthrough capability for long-lived flows. In PIM-SM, when the sender sends more packets than the threshold configured by the RP, the PIM routers on the listener's segment notify a threshold exceeded message to the sender. After that the PIM routers change to forward multicast packets onto the paths from where the notification messages came. These paths compose the Shortest Path Tree (SPT) which is composed by paths not dependent on the RPT. Unlike multicast, however, it is rare that the same client continuously sends many anycast packets with the same anycast address, because it is not guaranteed that multiple packets addressed to the same anycast address are forwarded to the same anycast receiver. In other words, the selected anycast receiver may change during the communication. Therefore the client would use the unicast address of the selected anycast address to establish the stateful communication instead of the anycast address. The anycast address is used only for the discovery of the anycast receiver, and the number of anycast packets from the same client would not be so large. For this reason, PIA-SM does not support the cutthrough (i.e., SPT) capability.

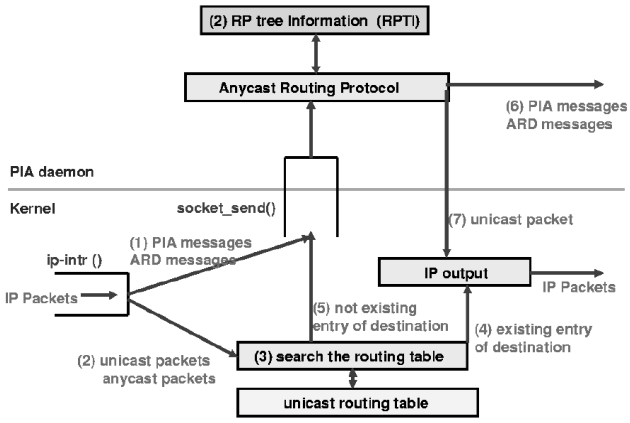
By PIA-SM designed in this section, an anycast packet is forwarded to the appropriate anycast receiver for the RP. But the anycast receiver which receives the anycast packet may not be the appropriate receiver for the sender of the packet. This issue will be resolved if the sender notifies the criteria for the selection of the appropriate receiver to a router. But one of the features of anycast communications is that a sender can connect to the appropriate receiver by only setting the anycast address as the destination address of the packet. It is because we design PIA-SM which realises anycast routing based on only routers and receivers.

3 Implementation of PIA-SM

In this section, we discuss issues on implementation of PIA-SM. It is the easiest way for implementation of PIA-SM because the source code of PIM-SM with IPv6 support is available on the BSD system. Therefore we implement PIA-SM on the BSD system by modifying the existing source code of PIM-SM. In the BSD system, PIM-SM is implemented in both the kernel and the daemon process. Since we implement PIA-SM by modifying the

PIM-SM; PIA-SM is also implemented in the kernel and the daemon process (called *PIA daemon*). Figure 5 shows the architecture of PIA-SM. Processes of PIA-SM are performed in either the kernel or the PIA daemon. Like the original BSD system, the kernel first invokes *ip-intr*() API when the IP packet arrives. *ip-intr*() then selects the appropriate function according to the information (e.g., protocol number, destination address) of the packet, and calls the selected function. After that, the function for routing messages calls PIA daemon if the received packet is a PIA message or ARD message. On the other hand, the function for data packets searches the destination address of the packet from the unicast routing table in the kernel. This function is modified from the original BSD system because the function must support anycast addresses. The problem raised here is that the kernel cannot determine whether the destination address of the packet is an anycast address or a unicast address. In multicast, because the multicast address is allocated its identical addressing space, PIM-SM can easily identify the multicast packet from received packets. However, in PIA-SM, the kernel cannot determine whether the received packet is the anycast packet or the unicast packet. For this problem, the kernel first looks up the unicast routing table, which is the same process as the original kernel. The difference is when the kernel fails to lookup the entry in the unicast routing table for the destination address of the received packet. In the original kernel, if the associated routing entry does not exist in the routing table, the packet is forwarded to the default router. On the other hand, the modified (PIA-SM) kernel passes the packet to the PIA daemon in order to check whether the received packet is the anycast packet or not.

Figure 5 Overview of the PIA daemon



3.1 Receiving routing message

We assume that PIA and ARD messages are assigned a new protocol number. That is, the kernel can easily identify PIA and ARD messages in *ip-intr*() API. By detecting PIA and ARD messages *ip-intr*() invokes *socket_send*() (see (1) in Figure 5) API to pass the message through the PIA daemon. The PIA daemon collects information of RPT from PIA and ARD messages. This information is stored into the RPT Information (RPTI) table. Table 2 shows the structure of RPTI for each anycast address. PIA daemon has RPTI table as the list of each anycast address's RPTI. The unicast address of the RP for the anycast address *G* is configured by the administrator. PIA daemon searches the next hop of the RP based on unicast routing and the next hop is the upstream PIA router for the address *G*. After that the PIA daemon writes entries of downstream routers/receivers into RPTI of the address *G*.

Table 2 The structure of RPTI

Anycast address		
Unicast address of the RP		
Link-local address of the upstream router		
Link-local address of downstream router	Metric	Interface
Link-local, address of anycast receiver	Metric	Interface

3.2 Receiving data packets

All packets except routing messages are passed through *ip6_forward*() API by *ip-intr*(). *ip6_forward*() API lookups the entry of the unicast routing table for the destination address of the packet. If the entry for the destination address exists, the packet is forwarded by calling *ip6_output*() (see Figure 5(4)). Otherwise, i.e., *ip6_forward*() cannot find the entry for the destination address and the packet is processed by the PIA daemon through *rip6_input*() API (see Figure 5(5)). It is different from the original kernel which simply forwards the packet to the default router through *ip6_output*(). On the other hand the kernel of PIA-SM then checks whether the destination address of the packet is anycast or unicast. This check is performed by the PIA daemon. The PIA daemon searches RPTI table to find the address specified in the packet. If the PIA daemon finds the address in the RPTI table, the daemon decides that the destination address of the packet is the anycast address. The PIA daemon encapsulates the anycast packet with the unicast address of the RP and send out the packet as a PIA Capsule. If the PIA daemon cannot find the address in the RPTI table, the daemon decides that the packet is unicast, and calls *ip6_output*() to forward the packet to the default router.

4 Demonstration of PIA-SM

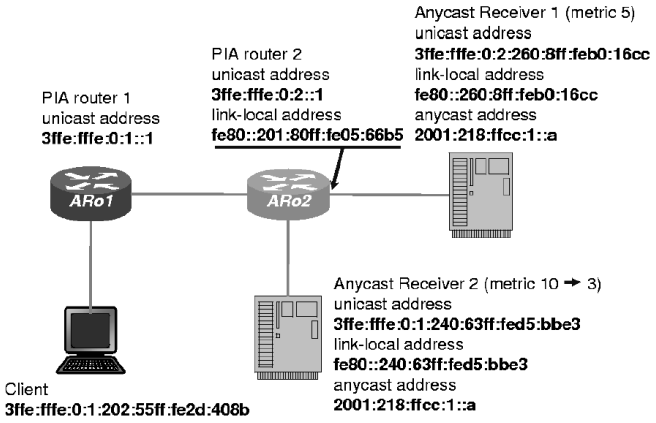
In this section, we show that anycast packets are forwarded to the selected anycast receiver by routers which the implemented PIA daemon works on.

Figure 6 shows the environment used in our experiments. There are two anycast receivers which have the same anycast address *G* (2001:218:ffcc:1::a). There are also two PIA routers *ARo*1 and *ARo*2. Anycast receivers are connected to *ARo*2, and the client is connected to *ARo*1.

- We test send ICMPv6 Echo Request to the address *G*. We check which receiver replies to Echo packet to the client. In addition, we confirm that the selected receiver will be changed by sending a new ARD report message with smaller metric
- We show that the client communicates with an appropriate web server. In this experiment, we run a HTTP server program on both anycast receivers.

We then connect to one of receivers by specifying the anycast address *G* on the web browser. That is, we open the web page of the URL http://[2001:218:ffcc:1::a]/. Note again that the anycast address cannot be set as the source address of the packet. The reasons are described in Hagino and Ettikan (2003). To support the stateful communication in anycasting, we run AARP (Doi et al., 2003) on the client. We also check which server is selected, by changing the metric of ARD reports.

Figure 6 The network components for the experiments



4.1 Test of echo request/reply with the anycast address

We use *ping6* for a roundtrip packet delivery. *Ping6* periodically sends ICMPv6 Echo Request packets to the destination specified by the argument of the program. The node having the destination address then receives the ICMPv6 Echo Request packets, and returns ICMPv6 Echo Reply packets to the sender. First we set metrics of Anycast Receiver 1 and 2 to be 5 and 10, respectively. We then run *ping6* program by setting the destination address to the anycast address *G* (2001:218:ffcc:1::a). During running *ping6*, we change the metric of Anycast Receiver 2 to be 3. We also check the source address ICMPv6 Echo Reply packets from the output of ping6.

Figure 7 shows the result of *ping6* program. From this result, the source address of the first two ICMPv6 replies are Anycast Receiver 1 (*3ffe:fffe:0:2:260:8ff:feb0:16cc*). However, after the third packet (icmpseq = 2), the source address of the last three packets are changed to Anycast Receiver 2(*3ffe:fffe:0:1:240:63ff:fed5:bbe3*). It is because we change the metric of Anycast Receiver 2 to be 3 at the time between receiving the second reply and sending the third request. Figure 8 shows the output of tcpdump program running on ARo2. In this figure there are ten packets in total. Also, packets #1, #2, #3 and #8 are indicated by multicast listener query/report packets while actual packets are ARD query/report packets. It is because the format of MLD and ARD packets are the same and *tcpdump* does not have a filter for ARD messages. So we regard such MLD packets as ARD packets.

Figure 7 Output of ping6 on the client

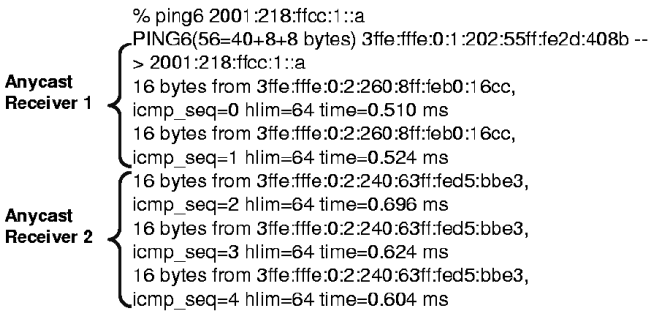
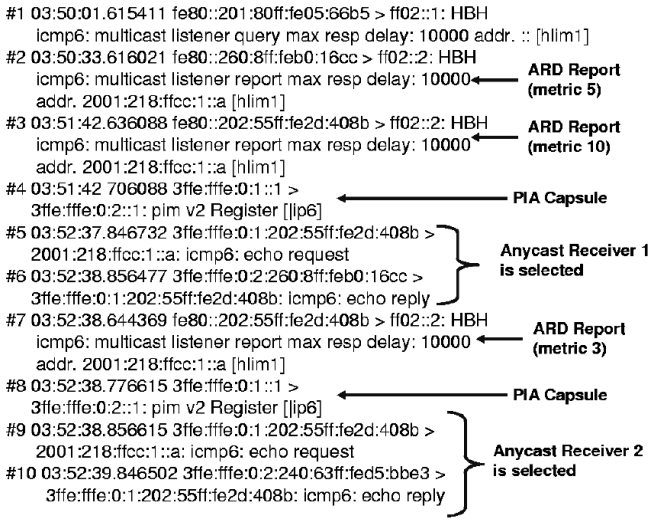


Figure 8 The experiment with echo request/reply



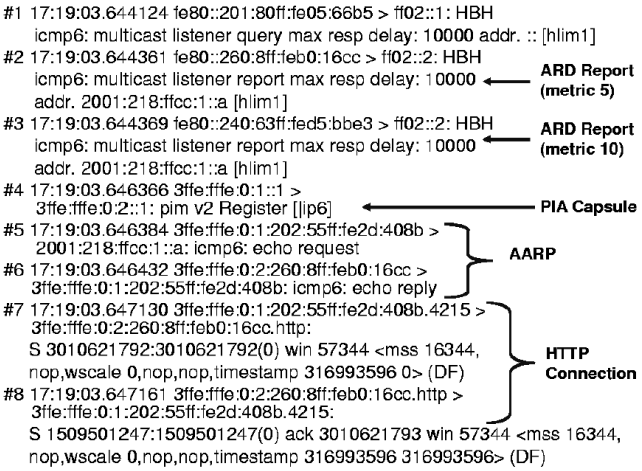
First, ARo2 sends ARD Query message in order to discover anycast receivers (#1). By the trigger on setting metrics, both anycast receivers send ARD Report messages to ARo2 (packet #2 is from Anycast Receiver 1, packet #3 from Anycast Receiver 2). Then the client starts *ping6* program for the anycast address G. Packet #4 and #8 are the encapsulated anycast packet (i.e., Echo Request) from the client. This packet is forwarded to the RP (*ARo*2) from the SR (*ARo*1) of the client. Packets #5 and #9 are ICMPv6 Echo Request messages destined to the anycast address 2001:218:ffcc:1::a. Packets #6 is ICMPv6 Echo Reply messages sent from Anycast Receiver 1 (*3ffe:fffe:0:2:260:8ff:feb0:16cc*). After packet #7 is captured, Anycast Receiver 2 changes its metric to 3. Then the ARD Report message is sent to ARo2 (packet #7). After that, ICMPv6 Echo Reply (packet #10) for packet #9 is sent from Anycast Receiver 2, because the metric of Anycast Receiver 2 is smaller than the one of Anycast Receiver 1 at that time.

4.2 Test of TCP connection with the anycast address

We run AARP to establish a TCP connection to an anycast address on the client. The client working AARP on sends ICMPv6 Echo Request to the destination address prior to beginning to establish a TCP connection. After that the client checks the source address of the received ICMPv6 Echo Reply. As AARP resolves an anycast address to a corresponding unicast address (i.e., the source address of the Echo Reply), the client establishes a TCP connection to the anycast receiver by its unicast address. As a result the client can establish a HTTP connection to the server and can browse a web page. Other conditions are the same in the first test.

Figure 9 shows the output of tcpdump on *ARo*2. In this figure there are eight packets in total. Also, packets #4 is indicated by PIM Register packet while actual packets are PIA Capsule packet. It is because the format of the PIM Register and the PIA Capsule are the same and tcpdump does not have a filter for PIA messages similar to the previous experiment. So we regard such PIM Register packets as PIA Capsule packets.

Figure 9 The experiment with web servers



First, similar to the previous experiment, ARo2 sends ARD Query message (#1), and both anycast receivers send ARD Report messages to ARo2 (#2 from Anycast Receiver 1 and #3 from Receiver 2). Then the client starts to establish a HTTP connection to the anycast address *G*. AARP running on the client sends an Echo Request in order to resolve an anycast address to the corresponding unicast address before establishing a TCP connection. Packet #4 is the encapsulated anycast packet (i.e., Echo Request) from the client. This packet is forwarded to the RP (*ARo*2) from the SR (*ARo*1) of the client. Packet #5 is the Echo Request message destined to the anycast address 2001:218:ffcc:1::a. This packet is obtained on the RP *ARo*2, and forwarded to Anycast Receiver 1 from *ARo*2. Packets #6 is the ICMPv6 Echo Reply message sent from Anycast Receiver 1 (3ffe:fffe:0:2:260:8ff:feb0:16cc). As AARP receives this Echo Reply, the AARP resolves the anycast address 2001:218:ffcc:1::a to the unicast address (3ffe:fffe:0:2:260:8ff:feb0:16cc) of the Anycast Receiver 1. After the anycast address is resolved by AARP, the client establishes the HTTP connection to the resolved unicast address of Anycast Receiver 1. #7 and #8 are the packets for the HTTP connection between the client and the Anycast Receiver 1. This experiment shows that the PIA-SM works on BSD systems and a client can connect to a most appropriate anycast receiver by PIA-SM.

5 Conclusions

In this paper we design a new anycast routing protocol called PIA-SM which realises IPv6 anycast communications. In addition we implement the PIA-SM router on an existing system and verify that the router can forward anycast packets to most appropriate anycast receiver.

The implemented PIA-SM selects an anycast receiver but that selection does not depend on any client but the RP gathers traffic of anycast packets on the RP. In addition, PIA router adds the entry of anycast addresses as host entries (i.e., entries with prefixlen 128) into its unicast routing table. These problems generate the serious scalability problem when we use PIA-SM as the anycast routing protocol in the global network. We should solve these problems in the future.

Ackowledgements

This work is partly supported by the Strategic Information and Communications R&D Promotion Programme (SCOPE-I) from Ministry of Internal Affairs and Communications of Japan.

References

Deering, S. and Hinden, R. (1998) 'Internet protocol, version 6 (IPv6) specification', *RFC2460*, December.

Doi, S., Ata, S., Kitamura, H., Murata, M. and Miyahara, H. (2003) 'Protocol design for anycast communication in IPv6 network', *Proceedings of 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03)*, Victoria, August, pp.470–473.

Doi, S., Ata, S., Kitamura, H. and Murata, M. (2004) 'IPv6 Anycast for simple and effective communications', *IEEE Communications Magazine*, Vol. 42, No. 5, May, pp.163–171.

Haberman, B. and Thaler, D. (2002) 'Host-based anycast using MLD', *Internet Draft draft-haberman-ipngwg-host-anycast-01. txt*, May, Expired November 2002.

Hagino, J. and Ettikan, K. (2003) 'An analysis of IPv6 anycast', *Internet Draft Draft-ietf-ipngwg-ipv6-anycast-analysis-02.txt*, June, work in progress.

Hinden, R. and Deering, S. (2003) 'IP version 6 addressing architecture', *RFC3513*, April.

Patridge, C., Mendez, T. and Milliken, W. (1993) 'Host anycasting service', *RFC1546*, November.