Making practical use of IPv6 anycasting: Mobile IPv6 based approach

Masakazu HASHIMOTO
Graduate School of Information Science and Technology, Osaka University
Yamadaoka 1–5, Suita-shi, Osaka, 565–0871 Japan
msk-hasi@ist.osaka-u.ac.jp

Shingo ATA
Graduate School of Engineering, Osaka City University
Sugimoto 3–3–138, Sumiyoshi-ku, Osaka, 558–8585 Japan
ata@info.eng.osaka-cu.ac.jp

Hiroshi KITAMURA
Solution Development Laboratories, NEC Corporation / University of Electro-Communications
(Igarashi Building 4F) Shibaura 2–11–5, Minato-ku, Tokyo, 108–8557 Japan
kitamura@da.jp.nec.com

Masayuki MURATA
Graduate School of Information Science and Technology, Osaka University
Yamadaoka 1–5, Suita-shi, Osaka, 565–0871 Japan
murata@ist.osaka-u.ac.jp

Abstract

Anycast is a new IPv6 function in which an anycast address can be assigned to multiple nodes that provide the same service. The Anycast packet is transmitted to one appropriate node selected out of all the nodes assigned to the same anycast address. Although it is expected that anycast will be used in various ways, the use of anycast is quite limited today. This is because anycast functionalities are still unclear, and it is difficult to realize a global anycast that requires existing routing protocols to be radically changed. In this paper, we clarify and classify some essential anycast issues and propose a new and practical communication architecture, based on Mobile IPv6 architecture, that overcomes the anycast difficulties.

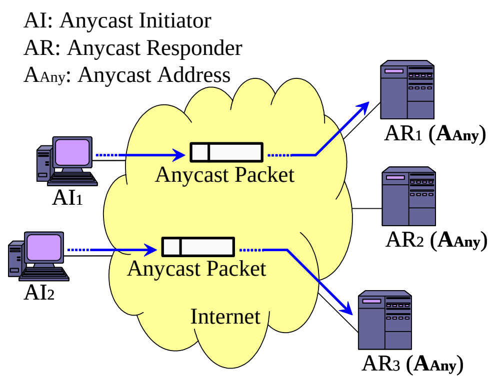


Figure 1. Communications in anycast

1. Introduction

Anycast [1] is a new communication form defined in IPv6 [2]. Unlike unicast and multicast, an anycast address is assigned to a service. Multiple nodes providing the same service can use the same anycast address [3]. Figure 1 shows an example of anycast. The same anycast address A_{Any} is assigned to the multiple nodes AR_1, AR_2, and AR_3, which are called Anycast Responders (AR). An anycast packet represents a packet whose destination address is anycast address. A node that sends an anycast packet is called an Anycast Initiator (AI). When an anycast initiator issues an anycast packet, the appropriate one of anycast responders will receive the anycast packet according to a selection criteria for the anycast addresses. One of important features of anycasting is that the appropriate anycast responder may change according to the network conditions, location of the anycast initiator, and the availability of anycast responders, even if the same anycast initiator sends multiple anycast packets. In Figure 1, the anycast initiators AI_1 and AI_2 send one anycast packet each. Unlike unicast and multicast, these anycast packets are not always delivered to same anycast responder, but are sent to e.g., anycast responders AR_1 and AR_3, respectively. The routing of anycast packets is performed on the network layer (i.e., the IP layer), and the applications in both the anycast initiator and the anycast responder do not require any knowledge about which anycast responder has been selected for the communication.

Thus, anycast has some unique and interesting characteristics. Following are some examples of anycast applications.

- Service discovery

  An anycast address enables us to assign a service-oriented address. To achieve a service discovery, we first assign an anycast address to each service, and then give the anycast address to the nodes on which the associated service is running. As a result, we can find the (appropriate) node providing the service simply by specifying the correspondent anycast address. For example, when we assign a well-known anycast address to a DNS server, the DNS query packets destined for the anycast address would be forwarded to an appropriate DNS server. Therefore, we can receive a response from an appropriate DNS server regardless of where we connected to the Internet.

- Load-balancing

  As the numbers of anycast responders having the same anycast address increase, anycast initiators can communicate with the appropriate anycast responders in each case. If anycast responders are globally distributed, the loads of anycast responders achieve balance.

- Robustness against a breakdown

  When an anycast responder fails, another responder with the same anycast address can receive the anycast packet. Therefore, the service for anycast initiators can be provided continuously even after the failure of an anycast responder.

Today, anycast is used in some applications. Some DNS root servers (e.g., F-, I- and K-root servers) are deployed using an anycast address. Due to the DNS specifications, the number of root DNS servers is limited in the world. The main purpose of assigning an anycast address is to expand the number of root DNS servers [4] to reduce the load of root servers. Another example is the assigning of an anycast address to Rendevous Points [5] for the multicast routing protocols (e.g., PIM-SM [6]). Also, the use of anycast is currently being discussed in [7], which presents a series of recommendations for distribution of services using anycast.

However, there are only limited practical uses for anycast today. While anycast DNS root servers and anycast RPs are examples of some practical applications, these applications are only for use by network operators. That is, there is still no way for individuals (e.g., end users, commercial companies) to use anycast address. The main purpose of this paper is to explain how to make anycast more widely available to end users. To this end, the following two principal issues need to be resolved.

- Lack of anycast routing architecture

  Although some anycast routing protocols have been proposed [8, 9, 10, 11], a more practical anycast routing scheme is needed. This is because routing protocols need some modifications of the deployment of routers across the Internet. To verify the effectiveness of anycast, it is necessary to make anycast easier to use (i.e., make a few modifications to the current Internet architecture).

- Difficulty in maintaining stateful sessions

  It is difficult to keep maintain sessions such as TCP when using anycast because each anycast packet issued by an anycast initiator is not always delivered to a particular anycast responder. Most of current Internet applications do not assume that communication peers may change during a communication. To keep a stateful session between an anycast initiator and an anycast responder using an anycast address, all packets addressed by the anycast initiator to the anycast address must be delivered to the anycast responder.

In this paper, we aim to design an anycast routing architecture that can maintain a stateful session so that end users can use many kinds of applications with anycast.

The rest of this paper is organized as follows: Section 2 introduces terminology and classifies types of anycast. Section 3 identifies the functions required to realize a global anycasting service. Section 4 introduces our proposed architecture, called Mobile IPv6-based IPv6 Global Anycast (MGA), which realizes IPv6 global anycast with making good use of Mobile IPv6 architecture. Finally, Section 5 presents our conclusions and future works.

2. Taxonomy of anycast

In this section, before discussing the anycast architecture, we introduce the terminologies used in this paper and classify some types of anycast.

2.1. Terminology

The terminologies in this paper follow the document titled ”IPv6 Anycast Terminology Definition” [12]. To assist with comprehension, we introduce the terminologies using Figure 2.

- Nodes

  An Anycast Initiator (AI) is a node that issues an anycast packet. An Anycast Responder (AR) is a node that can receive the anycast packet. A node actually receiving the anycast packet is called a Correspondent

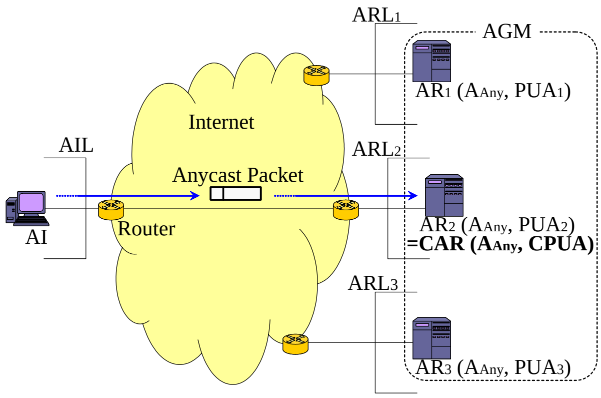


Figure 2. Basic Terminology

Anycast Responder (CAR). AR_1, AR_2 and AR_3 in Figure 2 are anycast responders, and AR_2 is also a correspondent anycast responder for the anycast packet shown in middle of the figure.

- Links

  A link where an anycast initiator exists is called an Anycast Initiator Link (AIL), while a link where an anycast responder exists is called an Anycast Responder Link (ARL). Thus, ARL_1, ARL_2, and ARL_3 in Figure 2 are all anycast responder links.

- Addresses

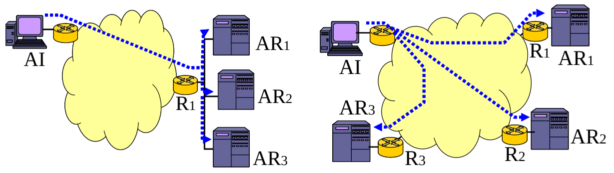
  An Anycast Address (AA) is defined in [3]. The prefix part of an anycast address is called an Anycast Prefix. You can see that AR_1, AR_2, and AR_3 in Figure 2 are assigned the same anycast address A_{Any}. Each of the anycast responders are also assigned one or more unicast addresses, which are called Peer Unicast Addresses (PUA). Specifically, a unicast address assigned to a correspondent anycast responder is called a Correspondent Peer Unicast Address (CPUA).In Figure 2, PUA_1, PUA_2, and PUA_3 are assigned to AR_1, AR_2, and AR_3, respectively. PUA_2 is also a correspondent peer unicast address for the anycast packet.

- Others

  A packet whose destination address is filled with anycast address is called an Anycast Packet. A group that consists of anycast responders that are assigned the same anycast address is called Anycast Group Membership (AGM). In Figure 2, AR_1, AR_2, and AR_3 are included in the same AGM.

2.2. Types of anycast

Anycast is categorized into two types according to the view point of the routing range. One type is a Subnet Anycast, where all anycast responders are in the same subnet (See Figure 3(a)). In the subnet anycast, the anycast packet



(a) Subnet Anycast

(b) Global Anycast

Figure 3. Two types of anycast according to view point of routing range

is transmitted to edge router R_1 by the existing unicast routing, and the edge router selects the correspondent anycast responder. The edge router can use a Neighbor Discovery mechanism [13] to select the CAR. Therefore, we have enough mechanisms to achieve a subnet anycast. In fact, subnet anycast is already realized in some services such as Mobile IPv6 Home-Agents anycast [14] and Subnet-Router anycast [3]. Another type is a Global Anycast, in which anycast responders are widely distributed across the Internet (See Figure 3(b)). A global anycast is more difficult to achieve than the subnet anycast because the anycast responders are not in a range that single router can manage. However, a global anycast can provide a wider anycast service, thus we aim to realize a global anycast through the work discussed in this paper.

Anycast is also categorized into two types according to view point of the layer. One type is a network layer (i.e., an IP layer) anycast, where routers or operating system in end nodes implement the anycast mechanism. When anycast is realized in the network layer, there is the advantage that the anycast functions can be added to existing applications without editing the source codes. Another type is the application layer anycast, where the application implements the anycast mechanism. In this case, we should edit or make new source codes for the applications. In this paper, we focus on the network layer anycast because our goal also includes to use anycast with existing applications.

3. Required functional components

Currently, there are no document defining how anycast should be achieved. In this section, we first classify the way to achieve global anycasting, and describe our design choice from the requirements.

3.1. Location of deployment

Firstly, we should determine where we implement the anycast architecture. Either routers and end nodes can implement the architecture.

- Implementation in routers

  When we implement the anycast architecture in routers, the routers should determine where to transmit an anycast packet. The approaches for implementing anycast architecture within routers are studied in [8, 9, 10, 11]. Although these architecture can forward anycast packet more flexibly, these approaches require the replacement of existing routers on the Internet. Therefore, it is not practical at the first phase of the anycast deployment scenario.

- Implementation in end nodes

  When we implement the anycast architecture in end nodes, the end nodes determine where to deliver an anycast packet. The implementation in the end nodes doesn't require router modification, so we can use the anycast architecture only by end node modification.

The modification/expansion on end nodes are easier than one on routers. In this paper, we aim to realize anycast architecture in end nodes because the motivation of our work is *practical*, that is, we desire to use anycast with a small modification to the current Internet.

3.2. Routing information management

A routing information related to the anycast address is essential to locate a correspondent anycast responder because the current unicast routing cannot manage the anycast address. In this subsection, we discuss the routing information required to locate a correspondent anycast responder. When the end nodes select the correspondent anycast responder, the end nodes need to maintain the anycast routing information to locate the correspondent anycast responder. An end node that selects a correspondent anycast responder may be an anycast initiator, or it may be another node such as a central server.

- Case of the anycast initiator

  The anycast initiator can select appropriate correspondent anycast responder for the anycast initiator's self. However, it is difficult for an anycast initiator to always maintain the information about anycast responders.

- Case of the central server

  The central server can maintain the information about anycast responders centrally. However, the traffic concentrates on the central server.

To make up for each other's weak point, we aim to use both an anycast initiator and a central server. That is, the routing information is intensively maintained in the central server, and the anycast initiator might maintain the routing information optionally to avoid the concentration of the traffic. The anycast architecture is easy to deploy when we use existing routing mechanisms to manage anycast routing information. So, we aim to use an existing routing mechanism to design our anycast architecture.

3.3. State management

When designing an protocol, there are two approaches to manage states of the protocol – a soft-state approach and a hard-state approach.

- Soft-state approach

  In a soft-state, the state is created and periodically refreshed by a ”refresh” message. The state is deleted if no matching ”refresh” messages arrive before the expiration of a given timeout interval. The state may also be deleted by an explicit ”teardown” message. This approach has trouble resistance because a state of node that fails is deleted within a given time interval. However, unnecessary traffic occurs in this approach to ”refresh” the state.

- Hard-state approach

  In a hard-state, in the absence of some event to trigger a protocol response, the protocol's state will remain unchanged for an unbounded time period. Unnecessary traffic doesn't occur in this approach because ”refresh” message is not used. However, once the system failed, the influence of the failure continues long because the state will be not refreshed.

[15] says that the soft-state approach is suitable for mechanisms used on the Internet because the soft-state mechanism possesses a strong robustness. We use this soft-state approach to manage states of anycast responders, that is, the routing information that relates the anycast address to the peer unicast address is managed via soft-state approach.

3.4. Transmission of an anycast packet

The nodes which implement the anycast architecture should transmit an anycast packet to a correspondent anycast responder.

- Implementation in routers

  The routers transmit the anycast packet with selecting next hop router according to anycast routing information which is maintained router-by-router.

- Implementation in end nodes

Because routers can transmit packets only by existing unicast routing, the end node should send the anycast packet to the correspondent anycast responder using existing unicast routing. To do so, the end node should maintain the information that relates the anycast address to the correspondent peer unicast address.

As discussed in Section 3.1, we implement the anycast architecture in end nodes because we aim to avoid router modification.

3.5. Design policies

As a conclusion, we decided to adopt following design policies:

- The anycast architecture is implemented in end nodes.
- The routing information is maintained by the central server and the anycast initiator.
- The routing information is managed via soft-state approach.
- The unicast routing is used to deliver the anycast packet.

Based on these policies, we propose a anycast mechanism named Mobile IPv6-based IPv6 Global Anycast (MGA) in the next section.

4. Mobile IPv6-based IPv6 Global Anycast (MGA)

In this section we propose a practical anycast architecture named Mobile IPv6-based IPv6 Global Anycast (MGA). Because MGA is designed to be implemented in the end nodes, replacing the existing routers is not required. Characteristically, MGA makes good use of Mobile IPv6 (MIPv6) [14] architecture, it can be used with a few modifications to MIPv6 source codes.

4.1. Brief introduction of MIPv6

First, we briefly introduce the concept of MIPv6 before discussing the MGA mechanism. MIPv6 is an architecture for maintaining the communication between Mobile Node (MN) and Correspondent Node (CN) whereever the mobile node connects to the Internet. A link where a mobile node originally exists is called a Home Link (HL). The mobile node has an address called a Home Address (HoA), which is used to communicate with the mobile node. The mobile node also has an address called Care-of Address (CoA).

There are two problems when a mobile node communicates with a correspondent node. First, it is not suitable to use the CoA of the mobile node directly because the CoA may change when the mobile node moves to another network. Second, from a correspondent node it is difficult to have a knowledge of CoA for the mobile node before the communication.

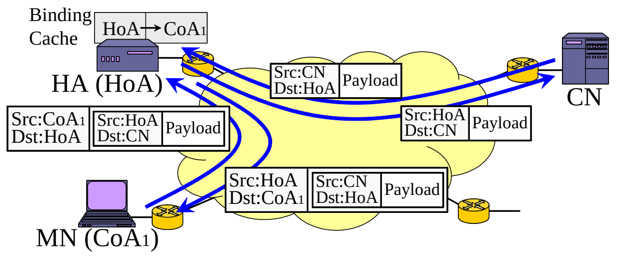


Figure 4. Communication between MN and CN

For these problems, the communication between the mobile node and the correspondent node is established with the home address. In order to maintain communication using the home address, Home Agent (HA) is deployed on the home link of the mobile node. Figure 4 shows a communication between a mobile node MN and a correspondent node CN through a home agent HA. HA can receive packets addressed to MN's home address HoA because HoA is assigned to HA. When MN is outside from the home link, MN establishes a virtual tunnel between MN and HA. To communicate from CN to MN, CN sends a packet whose destination address is HoA. The packet is firstly sent to HA. HA then encapsulates the packet, and sends to MN through the virtual tunnel. MN finally decapsulates the packet, and receive it. Similarly, from MN to CN, MN sends packets to CN through HA by a reverse tunnel.

HA maintains information called a Binding Cache. When MN moves to a different network, the binding cache is updated by MN using a message called a Binding Update, which includes a new care-of address of MN.

The communication between MN and CN is achieved via HA, however, this communication requires additional network resources and delays compared with a direct communication between MN and CN, since packets of both directions are always passed HA. The overhead is negligible for the long-term communication. To avoid this, MIPv6 defines a mechanism called Route Optimization. See Figure 5, CN also has a binding cache to perform the route optimization. When MN decides the route optimization, MN sends a binding update to CN. After the verification, CN updates its binding cache, and sends packets by using CoA_1 instead of the use of HoA. When MN moves to another

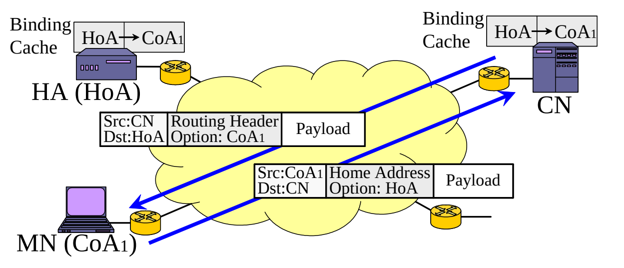


Figure 5. Route Optimization

network, a new binding update is sent to CN for notifying a new care-of address.

By analyzing the behavior of MIPv6 and anycast, we found that there is the same analogy between MIPv6 and the anycast architecture we want to design. These are (1) both communication architectures are implemented in end nodes, and (2) having the mechanism to manage routing information, and (3) having a location-free address (i.e., HoA or anycast address). We consider that we can design the anycast architecture making good use of MIPv6 architecture. Therefore we propose a new anycast communication architecture based on MIPv6, called MGA (Mobile IPv6-based IPv6 Global Anycast).

4.2. Basic mechanisms of MGA

The architecture of MGA is similar to MIPv6. In MGA, Home Anycast Agent (HAA) relays communication between an anycast initiator and a correspondent anycast responder. The home anycast agent exists on an Anycast Home Link (AHL) where the network prefix of the link equals the anycast prefix. That is, a anycast packet is delivered to home anycast agent. Figure 6 shows the mechanism of MGA. An anycast initiator AI issues an anycast

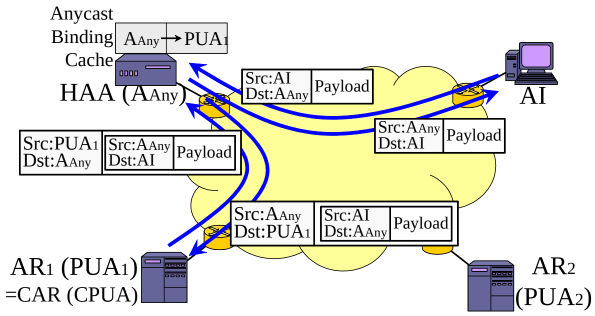


Figure 6. Comunication between AI and CAR

packet addressed to an anycast address A_{Any}, and a home anycast agent HAA receives the anycast packet.

HAA forwards the packet to a correspondent anycast responder CAR (i.e., HAA addresses the packet to a correspondent peer unicast address PUA_1) by a priori established tunnel. Finally CAR can receive the anycast packet. Similarly, CAR sends a packet to AI through HAA by reverse tunneling transmission. CAR can send a packet whose source address is filled with the anycast address even if CAR is on another network from the anycast home link.

To relay a communication of an anycast initiator and a correspondent anycast responder, a home anycast agent maintains information called an Anycast Binding Cache, which associates the anycast address to the correspondent peer unicast address. The anycast binding cache is managed with the previously mentioned soft-state approach, and is updated by the anycast responders using a message called an Anycast Binding Update. We show the details of the anycast binding cache in Subsection 4.4. Table 1 shows the correspondence of terminology between MGA and MIPv6.

MGA	MIPv6
Anycast Initiator (AI)	Correspondent Node (CN)
Anycast Responder (AR), Correspondent AR (CAR)	Mobile Node (MN)
Home Anycast Agent (HAA)	Home Agent (HA)
Anycast Home Link (AHL)	Home Link (HL)
Anycast Address (AA)	Home Address (HoA)
Peer Unicast Address (PUA), Correspondent PUA (CPUA)	Care of Address (CoA)
Anycast Binding Cache	Binding Cache
Anycast Binding Update	Binding Update

Table 1. Terminology Correspondence

4.3. Communication models

The MGA is divided into two models, Basic Model and Advanced Model, according to whether we use the MIPv6 architecture as is.

- The MGA Basic Model (MGA-BM)

  The MGA Basic Model is designed to use MIPv6 architecture as is and does not require any further protocols. We call this model MGA-BM for short. A great advantage of this model is that we can realize anycast directly (i.e., without any modification) by using MIPv6 architecture. However, because MIPv6 does not cover all functionalities of anycast, MGA-BM has the following two restrictions:

  – MGA-BM cannot manage multiple anycast responders simulteneously because the binding cache in MIPv6 cannot maintain multiple entries, therefore this model cannot select the appropriate correspondent anycast responder for AI basis.

  - The function for keeping the stateful session in this model is not sufficient. (details are described in Subsection 4.5)

  These restrictions are overcome by the MGA Advanced Model.

- The MGA Advanced Model (MGA-AM)

  The MGA Advanced Model is designed to extend MGA-BM (i.e., to extend the MIPv6 architecture). We call this model MGA-AM for short. New functions that make up for restrictions in MGA-BM can be realized by MGA-AM. In this paper, we propose following functions:

  - A function for managing multiple anycast binding cache to select the appropriate correspondent anycast responder for each AI.
  - A function for keeping a stateful session on various conditions.

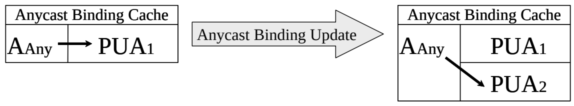
4.4. Anycast binding cache management

In the MGA, the anycast binding cache is used for routing. The anycast binding cache is maintained by a home anycast agent, and contains information that relates an anycast address to an peer unicast address. The anycast binding cache may also be maintained by an anycast initiator when an anycast responder executes a routing optimization procedure, discribed in Subsection 4.5. The home anycast agent refers the anycast binding cache, and selects an anycast responder which is to receive the anycast packet. The anycast responders update the anycast binding cache by sending the anycast binding update message to the home anycast agent, just as mobile node updates the binding cache by sending the binding update message to the home agent in MIPv6. To implement a soft-state mechanism, the anycast binding cache is periodically updated (i.e., refreshed) by the anycast binding update message, and is deleted if no anycast binding update message (i.e., refresh messages) arrives before the expiration of a given timeout interval. Because the binding cache in MIPv6 is designed via a soft-state approach, the anycast binding cache in MGA can easily inherit the positive points of the soft-state mechanism. The architectures of the anycast binding cache in MGA-BM and in MGA-AM might be different because MGA-BM is extended by MGA-BM; therefore, we discuss architecture of the anycast binding cache for MGA-BM and MGA-AM separately.

- Anycast binding cache in MGA-BM

  MGA-BM uses MIPv6 architecture as is, so the architecture of the anycast binding cache in MGA-BM is same as the one in MIPv6. Looking at the left side of Figure 7(a), the anycast binding cache in MGA-BM can only maintain a single relationship between an anycast address (A_{Any}) and a peers unicast address (PUA_1). When the anycast binding cache is updated by the anycast binding update message (see right side of Figure 7(a)), the relationship between A_{Any} and PUA_1 is overwritten by the new peer unicast address (PUA_2). At that point, the old relationship between A_{Any} and the PUA_1 is lost.

(a) Updating in MGA-BM



(b) Updating in MGA-AM

Figure 7. Updating anycast binding cache

- Anycast binding cache in MGA-AM

  Now, we discuss an extended function of MGA-AM. As previously mentioned, MGA-AM can possess a function for managing the information of multiple anycast responders. That is, the anycast binding cache in MGA-AM can contain multiple relationships between the anycast address and the peer unicast address. Looking at the left side of Figure 7(b), the anycast binding cache contains a single relationship between the A_{Any} and the PUA_1. When the anycast binding cache is updated by the anycast binding update message (see right side of Figure 7(b)), the relationship between A_{Any} and PUA_2 is added to the anycast binding cache. In this case, the old relationship between A_{Any} and PUA_1 is not lost, which differs from the case in MGA-BM. When the home anycast agent transmits the anycast packet, either PUA_1 or PUA_2 is selected according to the particular selection criteria.

4.5. Keeping a stateful session

To use anycast in various applications, we aim to keep the stateful session in anycast communications. In this subsection, we discuss a method for keeping the session. Because the home anycast agent transmits the anycast packet for various anycast responders case-by-case, it is difficult to keep the session in anycast. Therefore, we try to keep the session via direct communication between the anycast initiator and the anycast responder (i.e., without being relayed by the home anycast agent). In MIPv6, Route Optimization is defined to enable the mobile node to communicate with a correspondent node directly. Via route optimization, the anycast initiator can communicate with the anycast responder directly in MGA, too. So, we apply the route optimization mechanism to keep the stateful session in MGA. To practice route optimization, firstly AI initiates communication with CAR through HAA as shown in Figure 6. Secondly, CAR issues the anycast binding update message to AI, and AI can then maintain the anycast binding cache as well as HAA. In MGA, AI and CAR can then communicate directly, as shown in Figure 8. However,

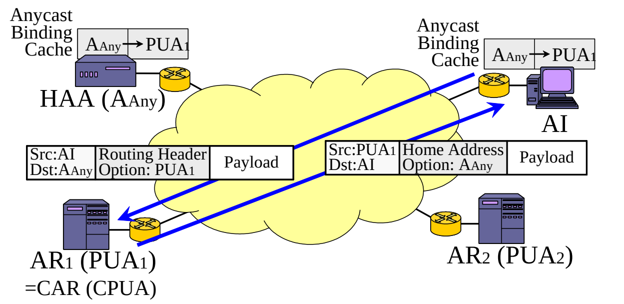


Figure 8. Route Optimization

this method might not keep the session completely because there is the problem that HAA might transmit the anycast packet to various anycast responders before the route optimization procedure is finished. To avoid this problem, we propose a mechanism that fixes the communication between AI and CAR. In the mechanism, HAA maintains the information regarding which AI communicates with which CAR. That is, HAA watches the source address (A_x) of the anycast packet and fixes the communication path from A_x to the selected CAR. HAA keeps the communication path within the given time-interval, and the anycast packets sent from A_x within the time-interval are transmitted to the CAR. Using this mechanism, the session can be kept. This mechanism cannot be realized by the MIPv6 mechanism, so this mechanism is realized in not MGA-BM, but in MGA-AM as an extended function.

4.6. Load-balancing and robustness

If the home anycast agent relays all the anycast packets addressed to a particular anycast address, traffic concentrates on the home anycast agent. To avoid concentration on the home anycast agent, we deploy multiple nodes named Midway Anycast Agents (MAAs) on the Internet. The midway anycast agent has the same functionalities as the home anycast agent and can exist on everywhere. In the network where the midway anycast agent exists, routing information is modified to transmit the anycast packet to the midway anycast agent. We must keep the modified routing information inside the network to avoid routing confusion [7]. See Figure 9, HAA is on network A, and the two

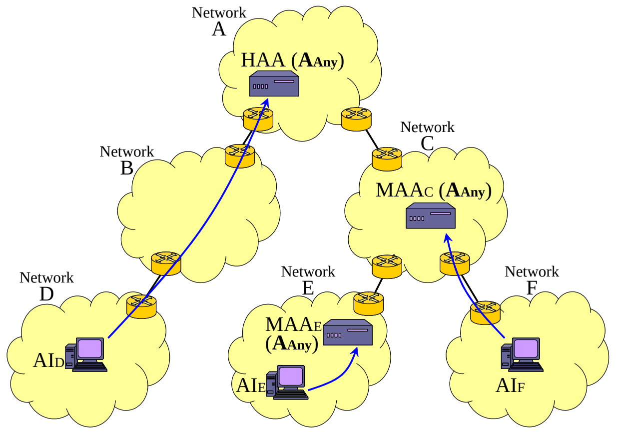


Figure 9. Load-balancing with MAA

midway anycast agents MAA_C, MAA_E are on network C and E. No midway anycast agent exists on the path from AI_D to HAA. In this case, an anycast packet AI_D issues is routed toward HAA using the unicast routing. MAA_C and MAA_E are respectively on the path from AI_F and AI_E to HAA. In this case, the anycast packets issued by AI_F and AI_E are received by MAA_C and MAA_E. MAA_C and MAA_E then relay the anycast packets to correspondent anycast responders selected by MAA_C and MAA_E respectively. By increase the number of midway anycast agents, we can distribute loads of anycast communications.

However, even if we deploy many midway anycast agents, all anycast packets not received by midway anycast agents are delivered to the home anycast agent. That is, the failure of the home agent leads the failure of whole service associated with the anycast address. Therefore, we should increase the robustness of the home anycast agent to ensure the availability of the service. We use a subnet anycast introduced in Subsection 2.2. We deploy multiple home anycast agents on the anycast home link (Figure 10). The home anycast agents are assigned to the same anycast address A_{Any}. The anycast packets addressed to A_{Any} are delivered to either HAA_1 or HAA_2. Thus we can ensure robustness against a breakdown of the home anycast agent. How-

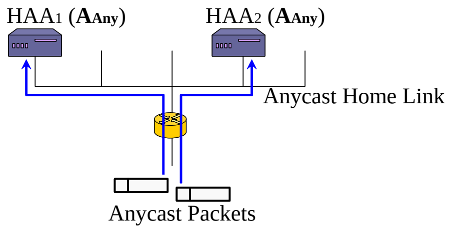


Figure 10. Robustness with multiple HAA

ever, the edge router has aweekness still. It is a problem to be solved.

5. Conclusions and future works

In this paper, we have highlighted the issues associated with anycast and have discussed our design for a new anycast mechanism, MGA. There are many methods for realizing anycast. We have first classified these into various types. Based on similarities from Mobile IPv6 architecture, we have proposed a new practical anycast communication architecture called MGA. Updating routing information, keeping stateful sessions, load balancing of home agents, and robustness against failure have also been discussed. We have two future works concerning MGA. Firstly, we will study further functions in the MGA advanced model. Secondly, protocol validation and implementation are also needed.

Acknowledgements

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

References

[1] C. Partridge, T. Mendez, and W. Milliken, "Host Anycasting Service." RFC1546, November 1993.

[2] S. Deering and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification." RFC2460, December 1998.

[3] R. Hinden and S. Deering, "Internet Protocol Version 6 (IPv6) Addressing Architecture." RFC3513, April 2003.

[4] T. Hardie, "Distributing Authoritative Name Servers via Shared Unicast Addresses." RFC3258, April 2002.

[5] D. Kim, D. Meyer, H. Kilmer, and D. Farinacci, "Anycast Rendevous Point (RP) mechanism using Protocol Independent Multicast (PIM) and Multicast Source Discovery Protocol (MSDP)." RFC3446, January 2003.

[6] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, and L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification." RFC2362, June 1998.

[7] J. Abley and K. Lindqvist, "Operation of Anycast Services." Internet-Draft (work in progress), February 2005.

[8] S. Doi, S. Ata, H. Kitamura, M. Murata, and H. Miyahara, "Protocol Design for Anycast Communication in IPv6 Network," in *Proceedings of 2003 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM'03)*, pp. 470–473, August 2003.

[9] S. Doi, S. Ata, H. Kitamura, and M. Murata, "IPv6 Anycast for Simple and Effective Service-Oriented Communications," *IEEE Communication Magazine*, vol. 42, pp. 163–171, May 2004.

[10] D. Xuan, W. Jia, W. Zhao, and H. Zhu, "A Routing Protocol for Anycast Messages," *IEEE Transactions on Parallel and Distributed Systems*, pp. 571–588, June 2000.

[11] D. Katabi and J. Wroclawski, "A Framework for Scalable Global IP-Anycast," in *Proceedings of ACM SIGCOMM*, pp. 3–15, August 2000.

[12] M. Hashimoto, S. Ata, H. Kitamura, and M. Murata, "IPv6 Anycast Terminology Definition." Internet-Draft (work in progress), February 2005.

[13] T. Narten, E. Nordmark, and W. Simpson, "Neighbor Discovery for IP Version 6 (IPv6)." RFC2461, December 1998.

[14] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6." RFC3775, June 2004.

[15] J. C. S. Lui, V. Misra, and D. Rubenstein, "On the Robustness of Soft State Protocols," in *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP' 04)*, 2004.