Master's Thesis

Title

Design and Development of
A Mobile IPv6-based Global Anycasting Mechanism

Supervisor

Professor Masayuki Murata

Author

Masakazu Hashimoto

February 15th, 2006

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Design and Development of A Mobile IPv6-based Global Anycasting Mechanism

Masakazu Hashimoto

Abstract

Anycast is a new IPv6 function, in which an anycast address can be assigned to multiple nodes that provide the same service. An anycast packet is transmitted to one appropriate node selected out of all the nodes assigned to the same anycast address. Anycasting provides such features as service discovery, load balancing, and robustness against breakdown. Furthermore, when anycast is implemented at the IP layer, existing applications can provide services over the anycast features without their source code needing to be modified. Some mechanisms for achieving anycast routing have already been proposed, though they are used only in a few applications because of two important limitations. One is that these mechanisms are not easy to deploy because the mechanisms require existing routers to be changed or require complex operational tasks; end users cannot even assign anycast addresses to their hosts. The other main limitation is that these mechanisms cannot provide "continuous"-type connections such as Transmission Control Protocol (TCP). Consequently, we propose a new anycast routing mechanism that uses Mobile IPv6, which is a routing mechanism for mobile nodes. In the proposed mechanism, anycast routing is achieved by the end nodes, removing the need to modify existing routers. End users themselves can assign anycast addresses to their hosts using our proposed mechanism. Furthermore, the proposed mechanism can provide "continuous"-type connections by using Mobile IPv6 functions. We validated the feasibility of the proposed mechanism through implementation experiments. In this thesis, we present and explain the proposed mechanism's design and development.

Keywords

Global Anycast

Routing Mechanism

Mobile IPv6

Continuous Connection

Contents

List of Figures

List of Tables

1 Introduction

The scale of the Internet has been expanding every year, and with it there has been a steady increase in the number of Internet users, as well as diversification of Internet services. The present Internet mainly uses the IPv4 (Internet Protocol version 4) [1] as a communication protocol. IPv4 was designed in 1970's, and has now been in use for decades. Unfortunately, however, IPv4 cannot function effectively enough to cope with this enormous increase in users and demand for the Internet. For example, the address space of IPv4 is not large enough to assign a unique address to all users who connect to the Internet. Moreover, no IPv4 specification has enough security or mobility functions that are now in great need, because security and mobility were not high on the list of requirements when IPv4 was designed. In response to IPv4's shortcomings, IPv6 (Internet Protocol version 6) [2] was designed as the next-generation protocol, and this is now gradually becoming the Internet's communication protocol. The grounds for this shift to IPv6 is that IPv6 is supported by many networks such as Internet Service Providers (ISP) and academic networks. Examples of IPv6's advantages over IPv4 are that the address space of IPv6 is so large that we do not have to worry about the amount of remaining address space even if a unique address were assigned to all Internet users, and that security and mobility functions are included in the IPv6 specifications. Furthermore, IPv6 includes some new functions that are currently in demand or are predicted to be useful in the future.

Anycast [3] is one of new functions defined in IPv6. There are actually three communication forms defined in IPv6; unicast, multicast, and anycast. The most characteristic feature of anycast is that an anycast address [4] can be assigned to any type of service. That is, multiple nodes providing the same service can be assigned the same anycast address. Also, in multicast, multiple nodes can be assigned a single multicast address. However, the point where anycast addresses are different from multicast address is that a packet addressed to an anycast address is transmitted to one appropriate node selected out of nodes assigned the anycast address.

Figure 1 shows an example of anycast communications. The single 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

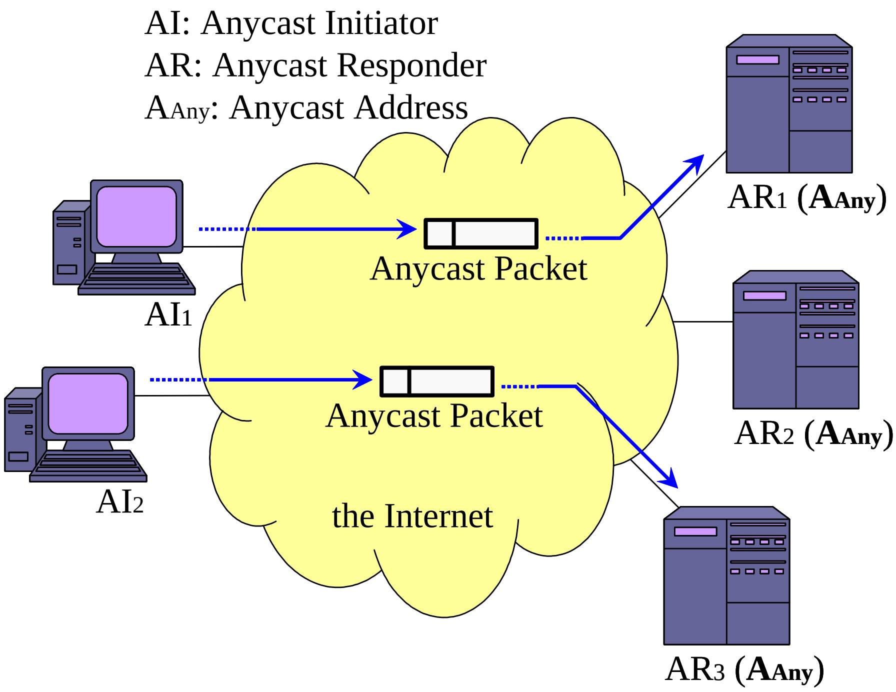


Figure 1: Communications in anycast

to a selection criteria for the anycast addresses. One of important features of anycasting is that the appropriate anycast responder for anycast packet may change according to the network conditions, location of the anycast initiator, or 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. Because the routing of anycast packets is performed on the IP-layer, 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.

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.

- Location-dependent service

  It is possible for us to connect to the nearest server using anycast. That is, we can connect to a location-specific server. For example, we can get the local time using the same anycast address even if we move between countries. It is similar to the “Emergency call” in the real world.

- 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 [5] to reduce the load of root servers. Another example is the assigning of an anycast address to Rendevous Points [6] for the multicast routing protocols (e.g., PIM-SM [7]). Also, the use of anycast is currently being discussed in [8], which presents a series of recommendations for distribution of services using anycast. Furthermore, some mechanisms to achieve anycast routing are proposed [9, 10, 11, 12]. However, these mechanisms are used only in a few applications because of two main limitations. One limitation is that these mechanisms are not easy to deploy because the mechanisms require existing routers to be changed or require hard operational tasks such as advertising of each anycast address to backbone networks of the Internet. End-users cannot even assign anycast addresses to their hosts because end-users cannot change/modify the routers, or advertise the routing information. Another limitation is that these mechanisms cannot provide "continuous"-type connections like Transmission Control Protocol (TCP). Currently, anycast is used only in "one shot"-type services like DNS. It is 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 "continuous"-type connections between an anycast initiator and an anycast responder using an anycast address, all anycast packets issued by the anycast initiator must be delivered to the same anycast responder.

Then, we propose a new anycast routing mechanism. Our proposed mechanism utilizes Mobile IPv6 which is a routing mechanism for mobile nodes. In the proposed mechanism, anycast routing is achieved by the end–nodes so that modifications of existing routers are not required. End-users themselves can assign anycast addresses to their hosts using proposed mechanism. Furthermore, the proposed mechanism can provide "continuous"-type connections by utilizing Mobile IPv6 functions.

The remainder of this paper is organized as follows: Section 2 explains anycast in more detail, and Section 3 discusses how we should design the anycast routing mechanism. Section 4 introduces our proposed mechanism, which achieves IPv6 global anycast routing by making good use of Mobile IPv6 mechanism. In Section 5, we validate the feasibility of our proposed mechanism in real machines. Finally, Section 6 presents our conclusions and future topics.

2 Anycast Taxonomy

For a smooth discussion in this thesis, we will describe details of anycast in this section. Firstly, we will introduce some terminologies which are essential in describing our proposed mechanism. Next, we will classify some types of anycast.

2.1 Anycast Terminology

We have arranged anycast terminology in an Internet-Draft, titled “IPv6 Anycast Terminology Definition” [13]. The terminology in this thesis follows the document. To assist with comprehension, we introduce the terminologies using Figure 2.

- Nodes
  - **Anycast Initiator** (AI):
    A node that issues an anycast packet.
  - **Anycast Responder** (AR):
    A node that can receive the anycast packet. AR_1, AR_2 and AR_3 in Figure 2 are anycast responders.
  - **Correspondent Anycast Responder** (CAR):
    A node actually receiving the anycast packet. AR_2 is a correspondent anycast responder.
- Links
  - **Anycast Initiator Link** (AIL):
    A link where an anycast initiator exists.
  - **Anycast Responder Link** (ARL):
    A link where an anycast responder exists. ARL_1, ARL_2, and ARL_3 in Figure 2 are all anycast responder links.
- Addresses

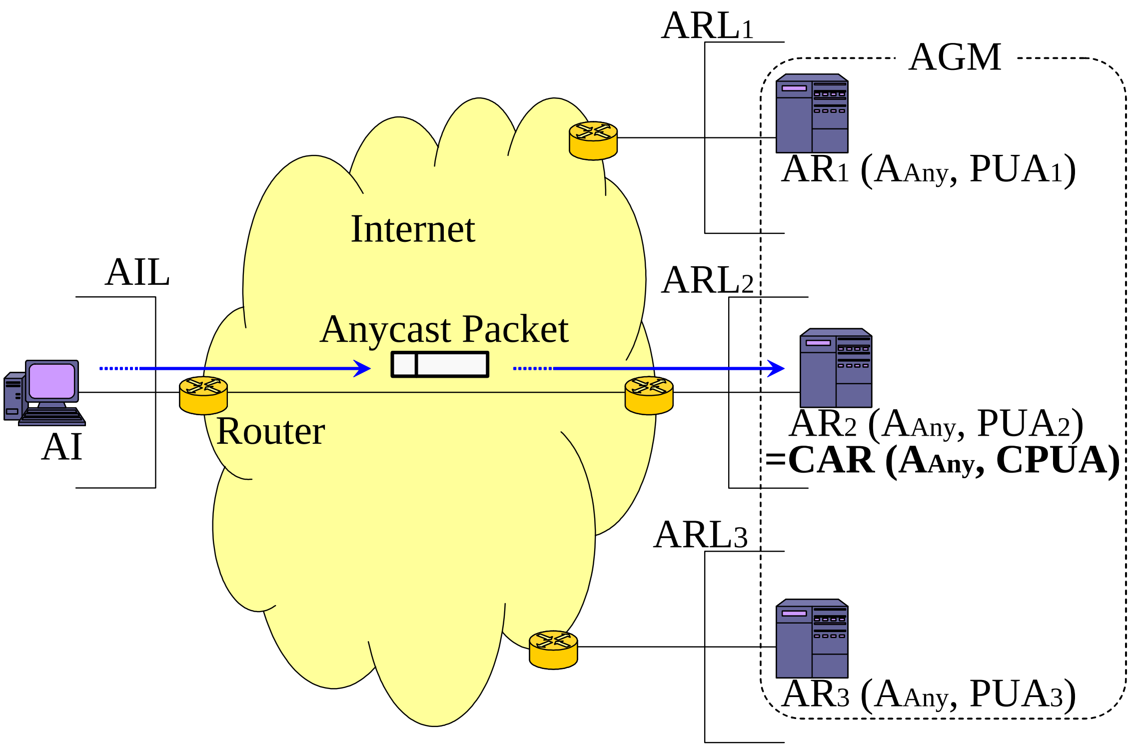


Figure 2: Basic terminology

– **Anycast Address** (A_{Any}):

  An address used for anycast communications. Defined in [4]. AR_1, AR_2, and AR_3 in Figure 2 are assigned the same anycast address A_{Any}.

– **Anycast Prefix**:

  A prefix part of an anycast address.

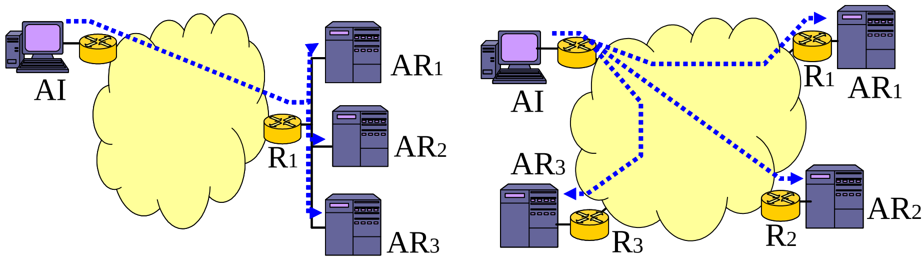
– **Peer Unicast Addresses** (PUA):

  One or more unicast addresses assigned to anycast responders. In Figure 2, PUA_1, PUA_2, and PUA_3 are assigned to AR_1, AR_2, and AR_3, respectively.

– **Correspondent Peer Unicast Address** (CPUA):

  One or more unicast addresses assigned to a correspondent anycast responder. PUA_2 in Figure 2 is a correspondent peer unicast address.

- Others



(a) Subnet Anycast (b) Global Anycast

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

- **Anycast Packet**:

  A packet whose destination address is filled with an anycast address.

- **Anycast Group Membership** (AGM):

  A group that consists of anycast responders that are assigned a same anycast address. 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 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 [14] to select the CAR. In fact, subnet anycast is already realized in some services such as Mobile IPv6 DHAAD [15] and Subnet-Router anycast [4]. 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 thesis.

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 thesis, we focus on the network layer anycast because our goal includes to use anycast with existing applications.

3 Designing an Anycast Routing Mechanism

In this section, we classify the functional components to achieve global anycasting, and describe our design choice.

3.1 Functional Components Required to Achieve Anycast Routing

In this subsection, we discuss functional components required to achieve anycast routing.

3.1.1 Nodes to Implement the Mechanism

Firstly, we determine where we implement the anycast mechanism. Either routers and hosts (i.e. end nodes) may implement the mechanism.

- Implementation in routers

  When we implement the anycast mechanism in routers, the routers should determine where to transmit an anycast packet. The approaches for implementing anycast mechanism within routers are studied in [9, 10, 11, 12]. Although these mechanism 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 mechanism 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 mechanism 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 mechanism 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.

3.1.2 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. 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 mechanism 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 mechanism.

3.1.3 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.

[16] suggests 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.

3.1.4 Selection of a Correspondent Anycast Responder

There are various criteria for selecting a correspondent anycast responder depending on the anycasting mechanisms and services provided. We show some examples of the criteria:

- Topological distance

  An anycast responder that is topologically closest to an anycast initiator is selected as a correspondent anycast responder. For example, a server that our ISP provides is selected as a correspondent anycast responder. This criterion reduces unnecessary lengthy communications and is the most suitable for a service that values a short round-trip time.

- Accessibility

  One of the available anycast responders is selected as a correspondent anycast responder, regardless of whether the correspondent anycast responder can provide the best service. This criterion is easier to implement because the necessary information is whether the anycast responders are working.

- Service-specific information

  According to the specific information provided by the anycast responders, an anycast responder is selected as a correspondent anycast responder. For example, multimedia streaming servers provide their server resource information such as throughput, and a server which provides the highest throughput is selected as a correspondent anycast responder. However, this mechanism needs the help of the application layer because the network layer cannot obtain the information such as throughput.

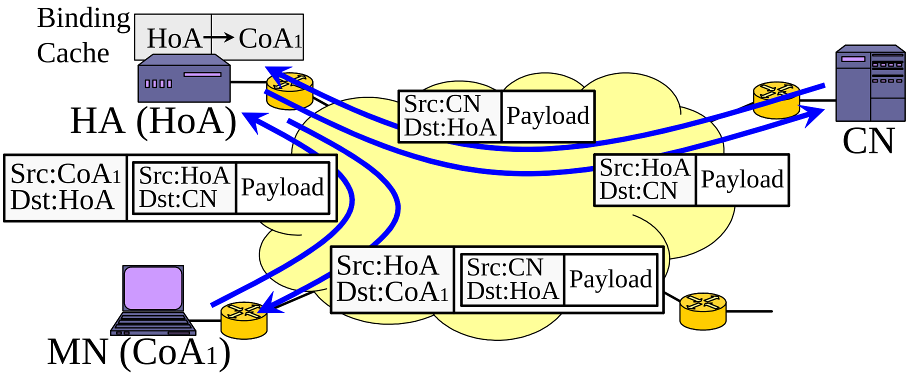


Figure 4: Communication between a mobile node and a correspondent node

3.2 Using Mobile IPv6 for Global Anycasting

When we use an existing mechanism to achieve an anycast routing, that anycast routing needs to be easy to deploy. We attempted to employ an existing mechanism that is ideal for achieving an anycast routing. The outcome was that we discovered Mobile IPv6 resolves the difficulty mentioned above and provides the functional components required in global anycasting. We will introduce the Mobile IPv6 mechanism in this subsection. Then we discuss the analogy between Mobile IPv6 and Global Anycast.

3.2.1 Brief Introduction of Mobile IPv6

We briefly introduce the concept of Mobile IPv6 before discussing the MGA mechanism. Mobile IPv6 is an mechanism for maintaining communication between a Mobile Node (MN) and a Correspondent Node (CN) wherever the mobile node connects to the Internet. A link where a mobile node originally exists is called a Home Link (HL), and the mobile node has an address called a Home Address (HoA), which the mobile node uses to communicate with other mobile nodes. Each mobile node also has an address called a 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 knowledge of a CoA for the mobile node prior to communication.

To solve these problems, communication between the mobile node and the correspondent node

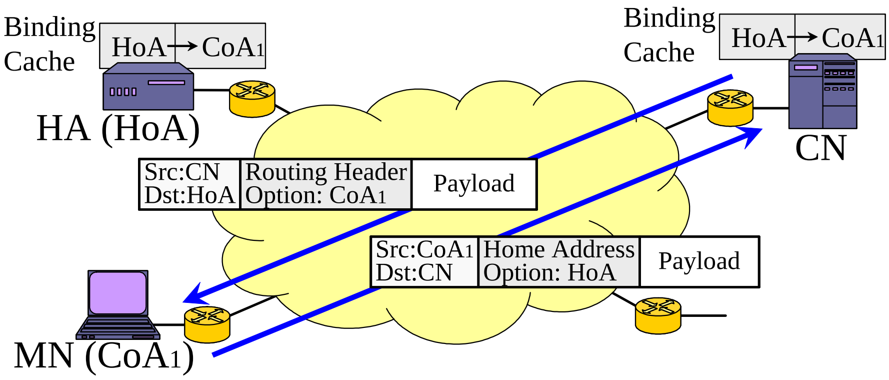


Figure 5: Route optimization in Mobile IPv6

needs to be established with the home address. In order to maintain communication using the home address, a Home Agent (HA) is deployed on the mobile node's home link. 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 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 first sent to HA. HA then encapsulates the packet, and sends to MN through the virtual tunnel. MN finally decapsulates the packet, and receives it. Similarly, from MN to CN, MN sends packets to CN through HA through 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.

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 through HA. The overhead is negligible for long-term communication. To avoid this, Mobile IPv6 defines a mechanism called Route Optimization. See Figure 5, where CN also has a binding cache to optimize the route. When MN decides the optimal route, MN sends a binding update to CN. After verification, CN updates its binding cache, and sends packets by using CoA_1 instead of using HoA. When MN moves to another network, a new binding update is sent to CN to notify the network of a

Table 1: Addresses used in Mobile IPv6 and global anycast

	Mobile IPv6	Global Anycast
Location-independent address	Home Address (HoA)	Anycast Address (A_{Any})
location-dependent address	Care-of Address (CoA)	Peer Unicast Address (PUA)

new care-of address.

3.2.2 Analogies Between Mobile IPv6 and Global Anycast

By analyzing the behavior of Mobile IPv6 and anycast, we found that there is an analogy between Mobile IPv6 and the generic anycast mechanism. The most significant analogy between Mobile IPv6 and global anycast is that both Mobile IPv6 and global anycast use two types of addresses: a location-independent address and a location-dependent address, which are shown in Table 1. In Mobile IPv6, mobile nodes use a home address to communicate with correspondent nodes, regardless of the network in which the mobile nodes exist. Similarly, in a global anycast, anycast responders use anycast addresses to communicate with anycast initiators regardless of the network where an anycast responder exists.

Furthermore, Mobile IPv6 functions are suitable for attaining the required functions described in Sub-section 3.1. That is:

- The routing information in Mobile IPv6 associates a location-independent address with a location-dependent address. Using this function, we can associate an anycast address with the peer unicast address of an anycast responder in global anycast.

- The routing in Mobile IPv6 is performed by end nodes. When we use this feature and design a global anycast routing mechanism in end nodes, we can achieve an anycast routing without modifying existing routers.

- The correspondent destination nodes of the anycast packets are decided by updating the routing information on the central node. Using this function, we can select the correspondent anycast responder in global anycast.

- State management in Mobile IPv6 is based on the soft-state approach. An anycast routing mechanism can be designed based on the soft-state approach.

Table 2: Correspondence between Mobile IPv6 and global anycast

	Mobile IPv6	Global Anycast
Nodes	Mobile Node (MN)	Anycast Responder (AR)
	Correspondent Node (CN)	Anycast Initiator (AI)
	Home Agent (HA)	Home Anycast Agent (HAA)
Links	Home Link (HL)	Anycast Home Link (AHL)
Addresses	Home Address (HoA)	Anycast Address (A_{Any})
	Care-of Address (CoA)	Peer Unicast Address (PUA)
Routing Information	Binding Cache	Anycast Binding Cache
	Binding Update	Anycast Binding Update

These analogies and functions enable us to design an anycast routing mechanism based on Mobile IPv6. When employing Mobile IPv6 to achieve global anycast, we map objects in Mobile IPv6 to ones in global anycast as shown in Table 2. The next section describes the proposed anycast routing mechanism.

4 Mobile IPv6-based Global Anycast (MGA)

We propose an anycast routing mechanism named Mobile IPv6-based Global Anycast (MGA). Characteristically, since MGA makes good use of the Mobile IPv6 functions. it can be implemented with a few modifications to the Mobile IPv6 functions in the initial phase of its deployment. In this section, we define two models in the proposed mechanism and describe operations in both models in detail.

4.1 Two Models in MGA

MGA is divided into two models, basic model and advanced model, according to whether we use the Mobile IPv6 mechanism as is.

- Basic model

  The basic model is designed to use Mobile IPv6 mechanism as is and does not require any further protocols. A great advantage of this model is that we can realize anycast directly (i.e., without any modification) by using Mobile IPv6 mechanism. However, because Mobile IPv6 does not cover all functionalities of anycast, the basic model has some limitations. These restrictions are resolved by the advanced model. The details of the limitations and the resolutions are described in subsection 4.3.

- Advanced model

  The advanced model is designed to extend the basic model (i.e., to extend the Mobile IPv6 mechanism). Advanced functions realized by the advanced model make up for limitations in basic model. We designed some advanced functions. The details of the advanced functions are described in subsection 4.3.

4.2 Basic Model

We first describe the functions of the basic model.

4.2.1 Communication Procedure

In MGA, home anycast agent (HAA) relays communication between anycast initiators and anycast responders. The home anycast agent exists on the anycast home link where the network prefix of

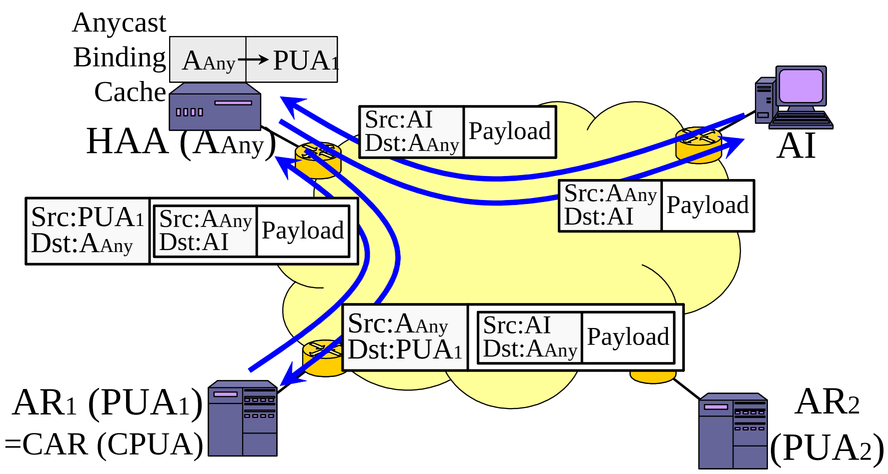


Figure 6: Communication between an anycast initiator and an anycast responder

the link equals the anycast prefix. That is, anycast packets are delivered to anycast home links by unicast routing. The anycast packets are then captured by HAA, and HAA relays the packets to anycast responders. 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 4.2.2.

Figure 6 illustrates some examples of basic communications in MGA.

- Transmitting an anycast packet from AI to AR

  (1) An anycast initiator AI issues an anycast packet addressed to an anycast address A_{Any}

  (2) The anycast packet is transmitted to the anycast home link by unicast routing

  (3) A home anycast agent HAA captures the anycast packet

  (4) HAA refers an anycast binding cache and receives a peer unicast address associated with the anycast address

  (5) HAA encapsulates the packet with the IP header in which the destination address is the peer unicast address

(6) HAA issues the encapsulated packet

(7) The packet is transmitted to AR by unicast routing

(8) AR decapsulates the packet and receives the original anycast packet

- Transmitting a packet from AR to AI

(1) AR encapsulates the packet addressed to AI with IP header in which the destination address is A_{Any}

(2) AR issues the encapsulated packet

(3) The packet is transmitted to the anycast home link by unicast routing

(4) HAA captures the packet

(5) HAA decapsulates the packet and receives the original packet addressed to AI

(6) HAA issues the decapsulated packet

(7) AI receives the packet

4.2.2 Managing Routing Informations

In MGA, the anycast binding cache is used as routing information. 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 subsubsection 4.2.3. 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 messages to the home anycast agent, just as mobile node updates the binding cache by sending the binding update messages to the home agent in Mobile IPv6. MGA basic model uses Mobile IPv6 mechanism as is, so the mechanism of the anycast binding cache in MGA basic model is same as the one in Mobile IPv6. Looking at the left side of Figure 7, the anycast binding cache in MGA basic model can only maintain a single relationship between an anycast address (A_{Any})

Figure 7: Anycast binding update in the basic model

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), 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.

Furthermore, 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 Mobile IPv6 is designed via a soft-state approach, the anycast binding cache in MGA can easily inherit the positive points of the soft-state mechanism.

4.2.3 "Continuous"-Type Connections with Route Optimization

To use anycast in various applications, we aim to keep the "continuous"-type connections in anycast communications. Then, we discuss a method for keeping the "continuous"-type connections. Because the home anycast agent transmits the anycast packet for various anycast responders case-by-case, it is difficult to keep the "continuous"-type connections. Therefore, we try to keep the connections via direct communication between the anycast initiator and the anycast responder (i.e., without being relayed by the home anycast agent). In Mobile IPv6, a function named route optimization is defined to enable the mobile node to communicate with a correspondent node directly. Via the route optimization, anycast initiators can communicate with anycast responders directly in MGA. We apply the route optimization mechanism to keep the stateful session in MGA. To execute route optimization procedure, firstly AI initiates communication with AR through HAA as shown in Figure 6. Secondly, AR issues the anycast binding update message to AI, and AI then maintain the anycast binding cache as well as HAA. AI and AR can then communicate directly, as shown in Figure 8.

However, this method might not keep the "continuous"-type connections when AI does not

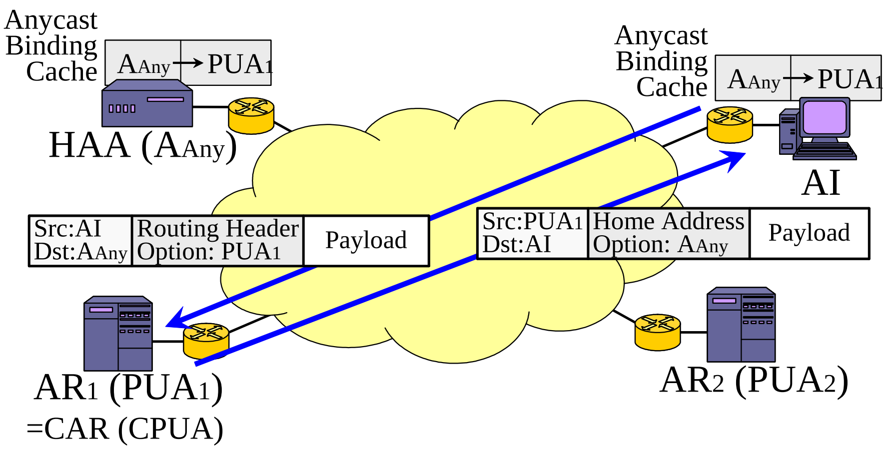


Figure 8: Route optimization in MGA

implement Mobile IPv6 functions. We will introduce a extended function to keep the "continuous"-type connections even when AI does not implement Mobile IPv6 functions in subsection 4.3

4.2.4 Limitations in Basic Model

As mentioned above, we achieved anycast routing via Mobile IPv6 functions in the basic model. However, the anycast functionalities in this basic model are not sufficient to provide various anycast services, because Mobile IPv6 was not originally designed with anycast capability. In this subsection, we describe limitations in the basic model caused by differences between Mobile IPv6 and global anycast. A main difference is that: *a home address is possessed by a single mobile node* in Mobile IPv6; on the other hand, *an anycast address is shared by multiple anycast responders* in global anycast. This difference causes some limitations as shown below in MGA basic model.

- Home anycast agents cannot distribute anycast packets to multiple anycast responders at one time. This is because only a single peer unicast address can be bound with a single anycast address in the basic model.

- "Continuous"-type connections may be destroyed when anycast initiators cannot execute route optimization procedure

- Traffic is centralized in a home anycast agent

We resolve these limitations via advanced functions defined in the MGA advanced model.

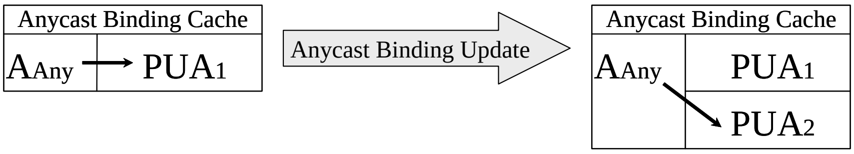


Figure 9: Anycast binding update in the advanced model

4.3 Advanced Model

To resolve the limitations of the MGA basic model, we designed some advanced functions for the updated version of MGA. These advanced functions are presented in this subsection.

4.3.1 Selection of Correspondent Anycast Responder

To select appropriate anycast responder out of available anycast responders, we define a function to associate a multiple peer unicast addresses with a single anycast address. To achieve this, we extended the anycast binding cache. Looking at the left side of Figure 9, 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 9), 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 basic model. When the home anycast agent transmits the anycast packet, either PUA_1 or PUA_2 is selected according to the application-specific metric.

4.3.2 "Continuous"-Type Connections without Route Optimization

We have designed a mechanism that fixes the communication between anycast initiators and anycast responders. In the mechanism, a home anycast agent maintains information about which anycast initiator communicates with which anycast responder. That is, a home anycast agent watches the source address (A_x) of the anycast packet and fixes the communication from A_x to the selected peer unicast address. A home anycast agent keeps the communication within the given time interval, and the anycast packets sent from A_x within that interval are transmitted to the same anycast responder. Using this mechanism, the "continuous"-type connections are kept even when anycast initiators cannot execute route optimization procedure.

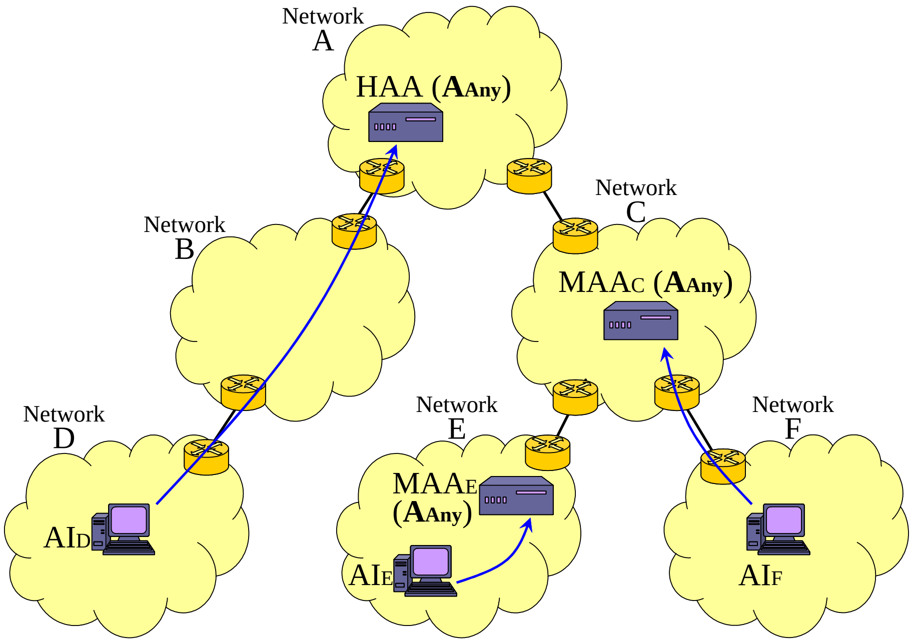


Figure 10: Load-balancing with midway anycast agents

4.3.3 Load-Balancing

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 traffic 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 be deployed 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.

See Figure 10. HAA is on network A, and the two 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.

5 Validation of the proposed anycasting mechanism

In this section, we demonstrate the feasibility of the proposed mechanism through implementation experiments. We show that MGA basic model works well utilizing an existing Mobile IPv6 implementation only. We first show the procedure of the anycast binding update performed by two anycast responders. Next, we show the transactions of anycast packets between anycast initiator and anycast responder.

We validate the proposed mechanism in the following environments.

- Network Environment: As shown in Figure 11, five network segments are connected to a single router. Anycast initiators AI_1 and AI_2 are located on networks 2001:db8:0:2::/64 and 2001:db8:0:3::/64 respectively. Two anycast responders AR_1 and AR_2 are located on networks 2001:db8:0:4::/64 and 2001:db8:0:5::/64 respectively. A home anycast agent HAA is located on network 2001:db8:0:1::/64.
- Nodes and addresses
  - Router: RT
  - Home anycast agent: HAA (2001:db8:0:1::2)
  - Anycast initiators: AI_1 (2001:db8:0:2::2), AI_2 (2001:db8:0:3::2)
  - Anycast responders: AR_1 (2001:db8:0:4::2), AR_2 (2001:db8:0:5::2)
  - Anycast address: A_{Any} (2001:db8:0:1::a)
- Operating System: FreeBSD 5.4 Release
- Mobile IPv6 Implementation: KAME SNAP (November 21st, 2005) [17]

The applications we use in the experiments are *ping* and *telnet*. *Ping* is used to examine behavior of a "one shot"-type connection, and *telnet* is used for a "continuous"-type connection. We observe sequences of packets by using *tcpdump* on each node. We assume that the home anycast agent HAA and all anycast responders AR_1 and AR_2 support Mobile IPv6 functions (i.e., integrate KAME SNAP implementation into the kernel). For anycast initiators, on the other hand, we validate two cases, 1) anycast initiators support Mobile IPv6, or 2) anycast initiators don't support Mobile IPv6 in order to validate the effect of a route optimization procedure.

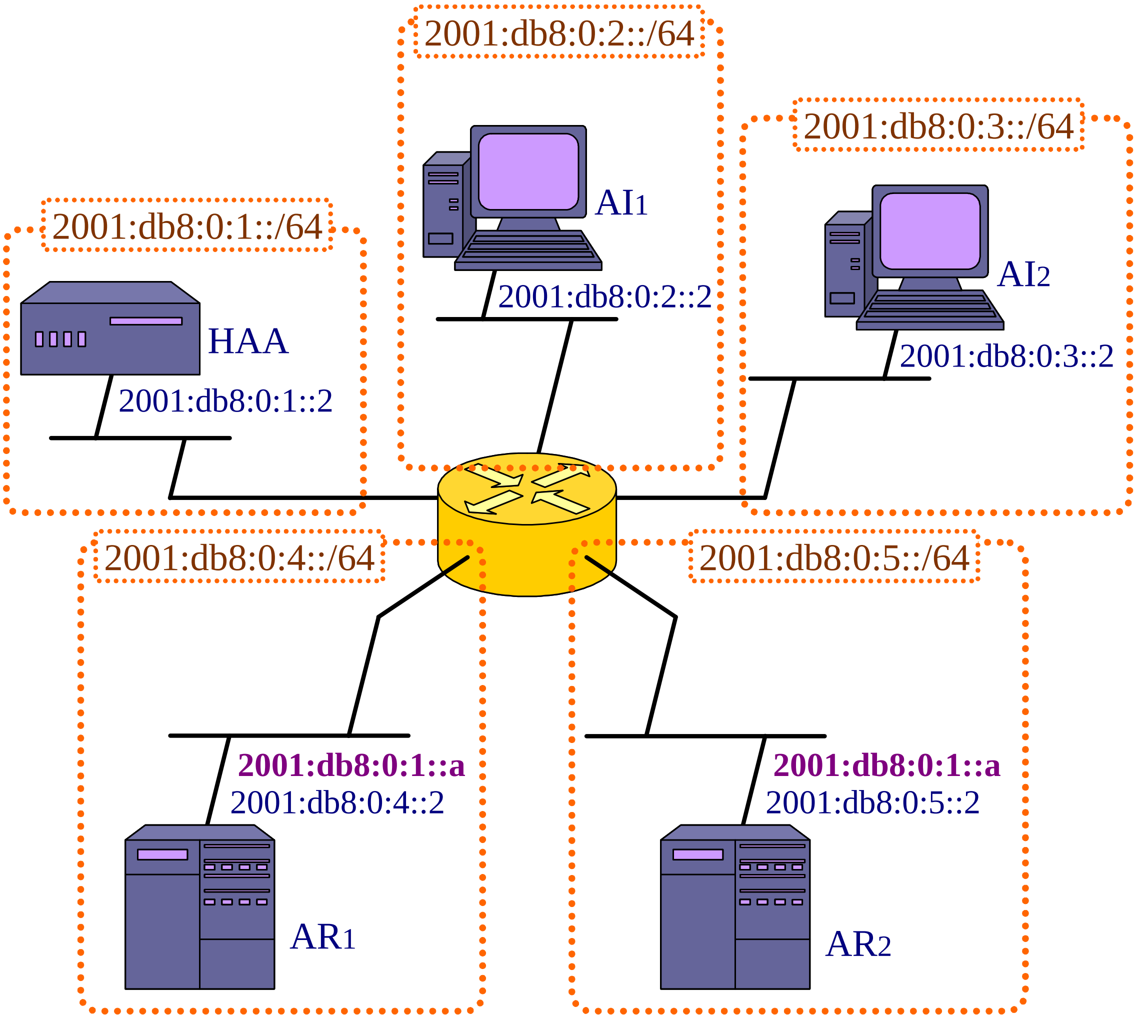


Figure 11: Network environment

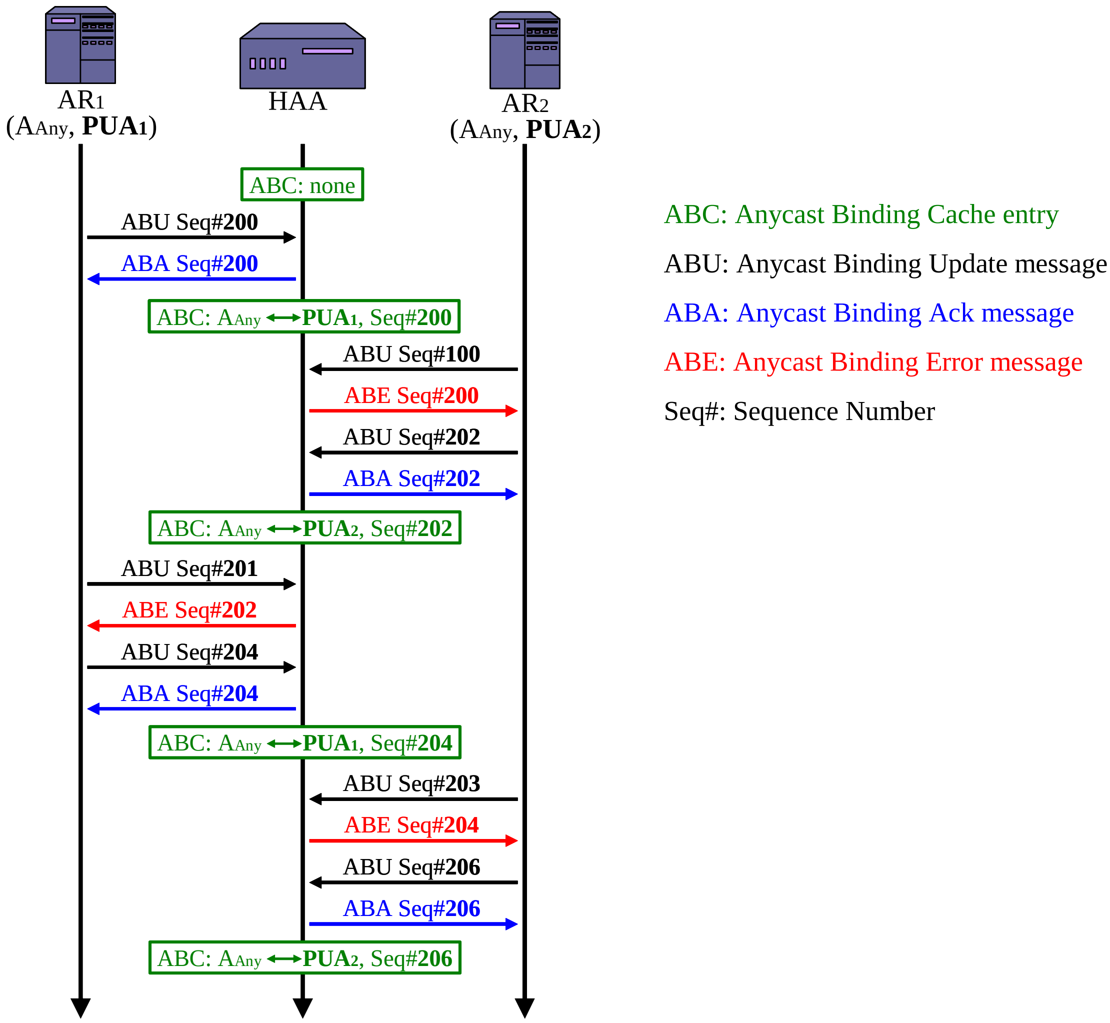


Figure 12: Anycast binding update performed by two anycast responders

5.1 Updating an Anycast Binding Cache

Before describing the procedure of the anycast binding update performed by two anycast responders, we introduce a sequence number which is embeded in both a binding cache and a binding update messages in Mobile IPv6. Sequence number is used to avoid a *repeated attack*, one of man-in-the-middle attacks where the attacker sends so many captured packets repeatedly. When a home agent receives a binding update message whose sequence number is already appeared in the previous received packet, the home agent sends *binding error* message to the sender of binding update message and do nothing on the binding cache.

We describe the procedure of the anycast binding update in MGA with Figure 12. Different from the case in Mobile IPv6, multiple anycast responders can send anycast binding update messages to a home anycast agent in MGA at the same time. That is, an anycast binding cache in the home anycast agent can be updated by multiple anycast responders. When the anycast responders AR_1 and AR_2 send an anycast binding update messages, each anycast responder doesn't care whether or not another anycast responder have updated the anycast binding cache in the home anycast agent HAA. It causes a problem that the sequence number in the anycast binding cache and one in the binding update message may not be consistent, and the anycast binding update message which has non-consistent sequence number is treated as an attack packet. Here, I show how MGA overcomes this problem. In Figure 12, AR_1 initially sends an anycast binding update message with a sequence number "200" to HAA. Then HAA replies to the message with a positive message called anycast binding ack and creates an anycast binding cache which has a sequence number "200" and associates an anycast address A_{Any} with PUA_1 which is AR_1's peer unicast address. Nextly, assume that AR_2 sends an anycast binding update message with a sequence number "100" to HAA. Because the sequence numbers in the anycast binding update message and the anycast binding cache in HAA are not consistent, HAA replies to the message with a negative message called anycast binding error and the anycast binding cache is not updated. However, because the anycast binding error includes the current sequence number "200", AR_2 can get a correct sequence number. Therefore, AR_2 immediately send a new anycast binding update message with a sequence number "202". For redundancy, AR_2 uses not "201" but "202" as the sequence number. Then HAA replies to the message with an anycast binding ack and creates an anycast binding cache which has a sequence number "202" and associates an anycast address A_{Any} with PUA_2 which is AR_2's peer unicast address. The anycast binding cache will similarly be updated by AR_1 and AR_2 after this. Thus, AR_1 and AR_2 can update the anycast binding cache individually.

5.2 Communications without Route Optimization

In this subsubsection, we show experimental results of communications between anycast initiators and anycast responders under the condition that the anycast initiators cannot execute route optimization procedure.

When HAA's anycast binding cache associates A_{Any} with PUA_1, all anycast packets issued by AI_1 and AI_2 are transmitted to AR_1 through HAA as shown in Figure 13(a). The packets AR_1 issued to AI_1 and AI_2 are transmitted to each anycast initiators through HAA. AR_1 sends an anycast binding update message to each anycast initiator to execute the route optimization procedure. However, AI_1 and AI_2 reply the message with an error message because AI_1 and AI_2 cannot recognize the anycast binding update message.

AR_2 executes anycast binding update, and HAA's anycast binding cache is updated to associate A_{Any} with PUA_2 as shown in Figure 13(b). Then all anycast packets issued by AI_1 and AI_2 are transmitted to AR_2 through HAA. Communications after that are similar to the case when the anycast binding cache in HAA associates A_{Any} with PUA_1. Whenever HAA's anycast binding cache is updated by another anycast responder, all anycast initiators begin to communicate with the anycast responder that updates HAA's anycast binding cache.

5.3 Communications with Route Optimization

Next, we show experimental results under the condition whether the anycast initiators support route optimization procedure or not.

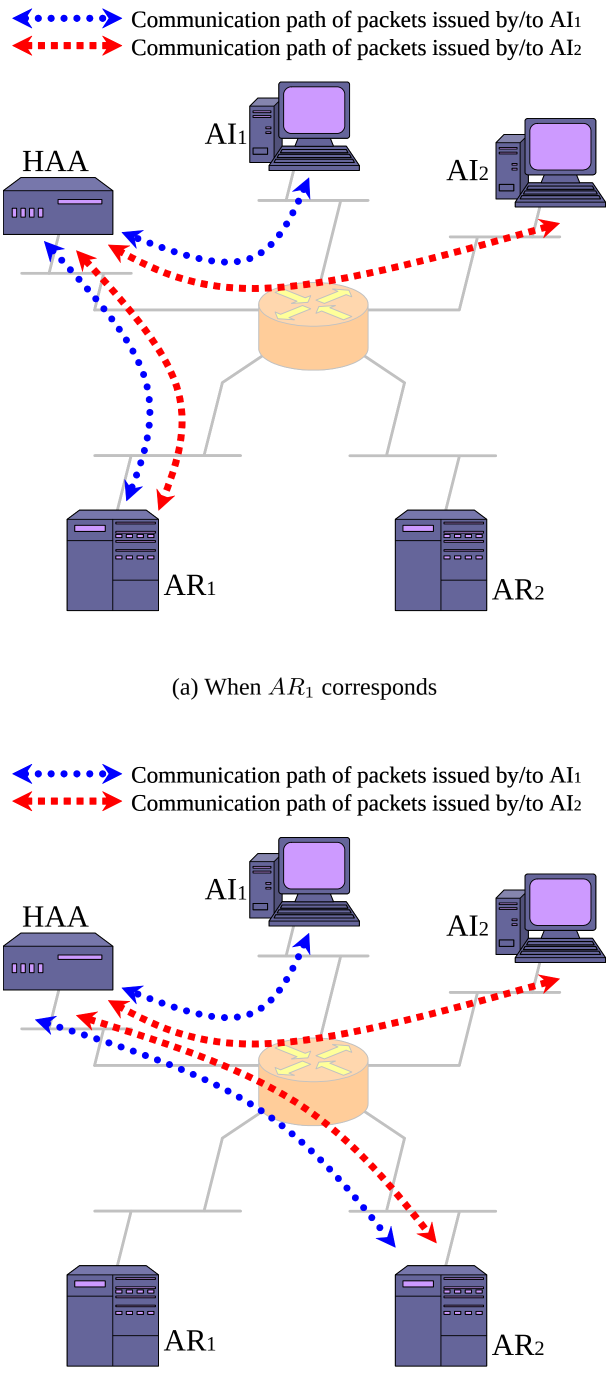
- When AI_2 supports route optimization and AI_1 doesn't

  See Figure 13(b), both AI_1 and AI_2 are communicating with AR_2 through HAA. AR_2 sends an anycast binding update message to AI_2 to execute the route optimization procedure. Then, AI_2 creates an binding cache which associates A_{Any} with AR_2's peer unicast address. And after that, as shown in Figure 14(a), AI_2 communicates with AR_2 directly (i.e., not through HAA). AI_1 continues to communicate with AR_2 through HAA.

  AR_1 executes anycast binding update and HAA updates its binding cache to associate A_{Any} with AR_1's peer unicast address. After that, as shown in Figure 14(b), AI_1 communicates with AR_1 through HAA and AI_2 communicates with AR_1 directly. Thus AI_1 and AI_2 communicate with AR_1 and AR_2 respectively.

- When both AI_1 and AI_2 supports route optimization

  See Figure 14(b), AI_1 is communicating with AR_1 through HAA, and AI_2 is communicating with AR_2 directly. AR_1 sends an anycast binding update message to AI_1 to execute the route optimization procedure. Then, AI_1 creates an binding cache which associates A_{Any}

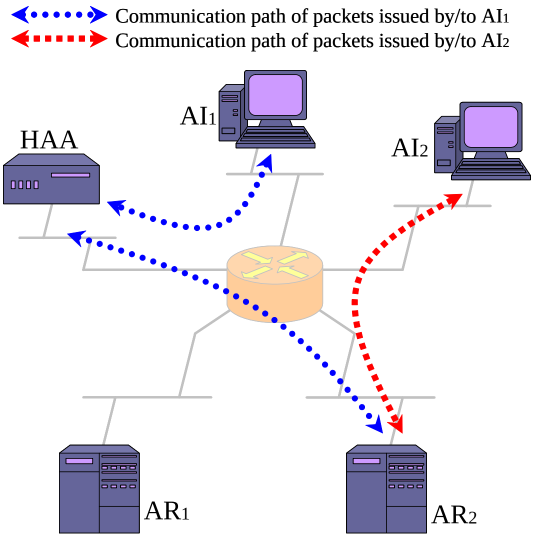


(a) When AR_1 corresponds

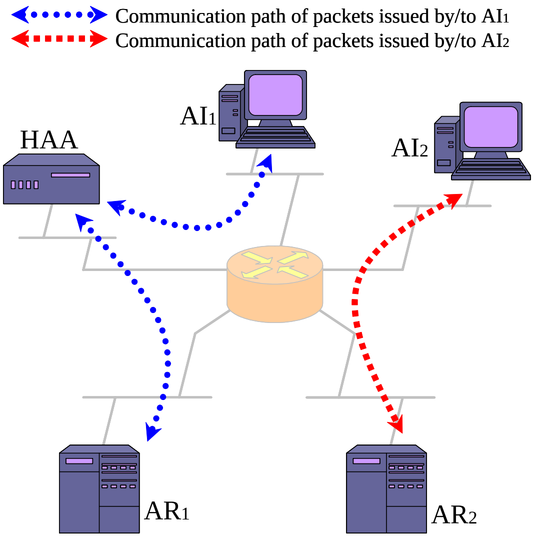
(b) When AR_2 corresponds

Figure 13: Communications without route optimization

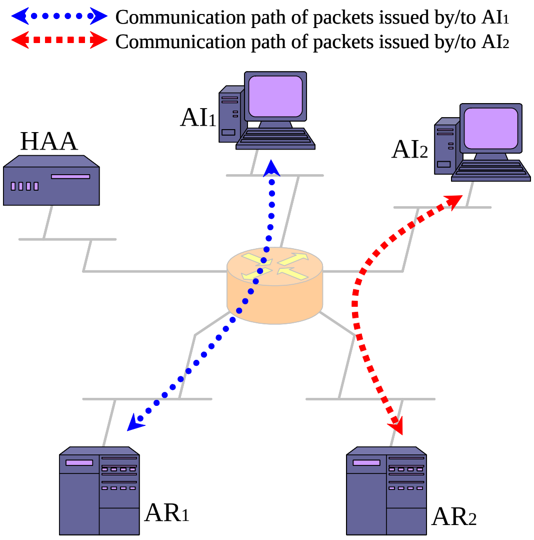
with AR_1's peer unicast address. And after that, as shown in Figure 14(c), AI_1 communicates with AR_1 directly. AI_2 continues to communicate with AR_2 directly. After that, even if HAA's anycast binding cache will be updated, Both AI_1 and AI_2 continues to communicate directly with AR_1 and AR_2 respectively.



(a) When AI_2 executes route optimization



(b) When AR_1 executes anycast binding update



(c) When AI_1 executes route optimization

Figure 14: Communications with route optimization

6 Conclusion

In this thesis, we proposed a new anycast routing mechanism based on Mobile IPv6. The proposed mechanism is achieved by using analogies between Mobile IPv6 and global anycast. We defined two models in the proposed mechanism: a basic model and an advanced model. The basic model employs Mobile IPv6 functions as is, thus all nodes that implement Mobile IPv6 can use it. In the basic model, anycast initiators and anycast responders can maintain “continuous”-type connections when they execute the route-optimization procedure defined in Mobile IPv6. The anycast functionalities provided by the basic model are not, however, sufficient to provide various anycast services, because Mobile IPv6 was not originally designed with anycast capability. The advanced model extends Mobile IPv6 functions to provide various anycast services. The advanced model features some additional functions such as an application-specific metric for home anycast agents to select appropriate anycast responders, a routing table for home anycast agents to maintain “continuous”-type connections, and the distributed deployment of anycast agents. We also validated the feasibility of our proposed mechanism in real machines.

Regarding future works, we plan to design and test further functions for the advanced model. Though functions in the basic model are limited to those already in Mobile IPv6, the functions in the advanced model are unlimited. Such additional functions in the advanced model will make it possible to develop new applications for anycast.

Acknowledgements

I would like to express my sincere appreciation and deep gratitude to Professor Masayuki Murata of Osaka University, who was my supervisor and has given me tremendous advice and valuable comments throughout my studies.

The work described in this thesis would not have been possible without the support of lecturer Shingo Ata of Osaka City University. He constantly provided me with appropriate guidance and invaluable advice.

I would also like to thank Dr. Hiroshi Kitamura of NEC Corporation. He was a great adviser to me, giving me plenty of ideas and inspiring me when my research did not advance.

I am most grateful to Professors Koso Murakami, Makoto Imase, Teruo Higashino, Hirotaka Nakano, and Tetsuji Satoh of Osaka University, for their appropriate guidance and invaluable firsthand advice.

I am also indebted to Professor Shinji Shimojo, Associate Professor Naoki Wakamiya, Associate Professor Hiroyuki Ohsaki, Associate Professor Go Hasegawa, and Research Assistant Shin'ichi Arakawa for their valuable support and advice.

Furthermore, I would like to express my gratitude to a number of friends and colleagues at the Graduate School of Information Science and Technology of Osaka University for their generous support, encouragement, and valuable advice.

References

[1] J. Postel, “Internet Protocol.” RFC791, Sept. 1981.

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

[3] C. Partridge, T. Mendez, and W. Milliken, “Host Anycasting Service.” RFC1546, Nov. 1993.

[4] R. Hinden and S. Deering, “Internet Protocol Version 6 (IPv6) Addressing Architecture.” RFC3513, Apr. 2003.

[5] T. Hardie, “Distributing Authoritative Name Servers via Shared Unicast Addresses.” RFC3258, Apr. 2002.

[6] 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, Jan. 2003.

[7] 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.

[8] J. Abley and K. Lindqvist, “Operation of Anycast Services.” Internet-Draft (work in progress), Jan. 2006.

[9] 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, Aug. 2003.

[10] 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.

[11] 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.

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

[13] M. Hashimoto, S. Ata, H. Kitamura, and M. Murata, "IPv6 Anycast Terminology Definition." Internet-Draft (work in progress), Jan. 2006.

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

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

[16] 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.

[17] "KAME SNAP." kame-20051121-freebsd54-snap.tgz, Nov. 2005. Available at http://www.kame.net/.

[18] M. Hashimoto, S. Ata, H. Kitamura, and M. Murata, "Making practical use of IPv6 anycasting: Mobile IPv6 based approach," in *Proceedings of International Symposium on Applications and the Internet (SAINT2006)*, Jan. 2006.