Server Mobility Using Domain Name System in Mobile IPv6 Networks¹

Hocheol Sung and Sunyoung Han

Department of Computer Science and Engineering, Konkuk University
1 Hwayangdong, Kwangin-gu, Seoul, 143-701, Korea
{bullyboy, syhan}@cc1ab.konkuk.ac.kr

Abstract. A mechanism using DNS to support server mobility in Mobile IPv6 networks is proposed in this paper. The name server in the mobile server's home domain maintains the mobile server's home address and care-of address. When the mobile server changes its link and gets a new care-of address, it sends a dynamic DNS update request to the name server to update its care-of address. Clients perform the DNS lookup to find the mobile server's home address and care-of address and set the connection directly to the mobile server.

1 Introduction

When a correspondent node begins communication with a mobile node in Mobile IPv6 networks, the correspondent node has no binding for the mobile node. Thus the data sent from the correspondent node should be routed to the mobile node via its home agent [1]. If a server is mobile, called a mobile server, the total number of clients that send requests to connect to mobile servers at the same time would increase in proportion to the number of mobile servers that are registered to a home agent. As a result, requests to mobile servers are concentrated in the home agent so that the load of the home agent increases. Moreover, as a mobile server becomes more distant from its home link, transmission delay over the tunnel between the home agent and the mobile server also increases. In the worst case, if a client cannot reach a mobile server's home agent because of the home agent crash or link failure, the client cannot even connect to the mobile server.

To connect to the mobile server as it changes its link, we take advantage of DNS and its ability to support dynamic updates. Most Internet users may use the hostname rather than complex IPv6 address at the beginning of a connection. The client application performs DNS lookup and receives the mobile server's IPv6 address as a DNS response [2]. When a mobile server moves to another link, it sends a dynamic DNS update to a name server in its home domain updating its current location [3]. As a result, the name server has the name-to-care-of address mapping for the mobile server and clients can get the mobile server's care-of address by means of DNS lookup.

¹ This work is supported by the Korean Science and Engineering Foundation under grant number R01-2001-000-00349-0(2003)

2 Operations

The following cases are described in this paper:

- When a client starts communication with a mobile server.
- When the mobile server responds to the client.
- When the mobile server moves to another link from the current link.

2.1 When a Client Starts Communication with a Mobile Server

A client that wants to connect to a mobile server performs a DNS lookup to find the mobile server's home address and the care-of address. According to the transport layer protocol or the application program, it is necessary that the transport layer connection such as TCP session be established before sending data and maintained during communication. In order to maintain the TCP connection when a mobile server moves to another link, clients have to establish the connection to the mobile server with its home address at the beginning of the connection. In this case, the client has to send the initial packet such as TCP SYN segment with the Routing header containing the mobile server's home address. Before the packet is sent, the mobile server's care-of address becomes the destination address of the packet and the mobile server's home address is moved to Address 1 in the Routing header. If the client is also mobile, the source address in the IPv6 header is set to the client's care-of address and the packet includes a Destination Options Header with the Home Address option containing the client's home address.

2.2 When the Mobile Server Responds to the Client

Upon receiving the packet from the client, the mobile server adds an entry for the client to its binding update list and sends a responding packet with a Destination Options header that contains the Home Address option and the Binding Update option. The Home Address option indicates the mobile server's home address. If the client is also mobile, the destination address in the IPv6 header of the packet is set to the client's care-of address and the packet includes a Routing header with the client's home address.

2.3 When the Mobile Server Moves to Another Link from the Current Link

As the mobile server moves to another link, it acquires a new care-of address through the conventional IPv6 mechanisms. Whenever the mobile server changes its care-of address, it sends a dynamic DNS update request to the name server in its home domain updating its current care-of address. In addition, the mobile server should also send Binding Updates to its home agent and clients as described in Mobile IPv6 [1].

3 Experiments

3.1 CAAAA Resource Record

To store the mobile server’s care-of address, a new resource record, CAAAA, is defined in this paper. The type field is set to CAAAA and the data section of the record simply contains the mobile server’s care-of address. A CAAAA resource record type is a record specific to the Internet class that stores a single mobile server’s care-of address. The value of TTL field must be set to zero since the record applies only to the current transaction and should not be cached. A CAAAA query for a mobile server’s host name in the Internet class returns the mobile server’s care-of address in the additional section of the response.

3.2 Simulated Results

The following cases are simulated using OMNeT++, a network simulation tool, for estimating the approach proposed in this paper [4].

- When the number of clients increases
- When the mobile server becomes more distant from its home link

We simulated the RTT (round trip time) in each case for measuring the response time between a client and a mobile server. In all following figures, graph (a) shows the simulation result when the connecting request from the client is sent directly to the mobile server and the graph (b) shows the simulation result when the connecting request from the client is routed to the mobile server via the home agent.

Fig. 1. RTT variation as the number of clients increase

Fig. 1 shows RTT variation caused by the increment of the number of the client. Transmission delay on the network, Processing time on the home agent and the mobile server are considered as parameters in this simulation. It is note that the RTT

increases in case (b) much more than in case (a) according to the increment of the number of clients.

The following figures show RTT variation when the mobile server becomes more distant from its home link. Hop count variation indicates distance variation between the mobile server and its home agent or between the mobile server and the client in our simulation. Transmission delay on the network and per hop, processing time on the home agent and the mobile server are considered as parameters in this simulation.

Fig. 2. As the mobile server becomes more distant from its home link, 1) the mobile server also becomes more distant from its client. 2) the mobile server becomes nearer to its client.

4 Conclusion

This paper presents a mechanism using DNS for server mobility in Mobile IPv6 networks. Although the extra processing time is needed for processing IPv6 extension headers, it is far smaller than the extra transmission delay caused by being routed via the home agent. The simulated results show that it can always shorten the round trip time when clients connect to the mobile server directly.

References

1. Perkins, C. E., and Johnson, D. B.: Mobility support in IPv6, Internet Draft, IETF, June 30, 2003 draft-ietf-mobileip-ipv6-24.txt.
2. Mockapetris, P.: Domain Names – Implementation and Specification, RFC 1035, IETF, November 1987.
3. Vixied (Ed.), P., Thomson, S., Rekhter, Y. and J. Bound: Dynamic Updates in the Domain Name System, RFC 2136, IETF, April 1997.
4. OMNeT++ Community Site, <http://www.omnetpp.org>.