Implementation and Evaluation of DSMIPv6 for MIPL*

Mingli Wang[1], Bo Hu[1], Shanzhi Chen[2], and Qinxue Sun[1]

[1] State key Lab of Switching and Networking Technology,
Beijing University of Posts and Telecommunications, Beijing 100876, China
[2] State key Lab of Wireless Mobile Communication,
China Academy of Telecommunication Technology, Beijing 100083, China
wmlibx@163.com

Abstract. Mobile IPv6 performs mobility management to mobile node within IPv6 networks. Since IPv6 is not widely deployed yet, it's an important feature to let Mobile IPv6 support mobile node moving to IPv4 network and maintaining the established communications. DSMIPv6 specification extends Mobile IPv6 capabilities to allow dual stack mobile nodes to move within IPv4 and IPv6 networks. This paper describes the implementation of this feature based on open source MIPL under Linux. By performing experiments on testbed using the implementation, it is confirmed that the DSMIPv6 works as expected.

Keywords: Mobile IPv6, DSMIPv6, MIPL.

1 Introduction

Mobile IP allows mobile nodes to move within the Internet while maintaining reachability and ongoing sessions, using a permanent home address. It can serve as the basic mobility management method in the IP-based wireless networks. Mobile IPv6 (MIPv6) [1] shares many features with Mobile IPv4 (MIPv4) and offers many other improvements. There are many different versions of MIPv6 available today, such as MIPL [2] and SunLab's Mobile IP [3]. Some works have been done on testbed for MIPv6, such as [4][5], they only support IPv6 network.

However, since IPv6 has not been widely deployed, it is unlikely that mobile nodes use IPv6 addresses only for their connections. It is reasonable to assume that mobile nodes will move to networks that might not support IPv6 and would therefore need the capability to support IPv4 Care-of Addresses. Since running MIPv4 and MIPv6 simultaneously is problematic (as illustrated in [6]), The Dual Stack MIPv6 (DSMIPv6) [7] specification extends Mobile IPv6 capabilities to allow dual stack mobile nodes to request their home agent forward IPv4/IPv6 packets, addressed to their home addresses, to their IPv4/IPv6 care-of addresses. Mobile nodes can maintain

* Foundation item: This work is supported by the National High-Technology(863) Program of China under Grant No. 2006AA01Z229, 2007AA01Z222, 2008AA01A316 and Sino-Swedish Strategic Cooperative Programme on Next Generation Networks No. 2008DFA12110.

P. Primet et al. (Eds.): GridNets 2008, LNICST 2, pp. 98 – 104, 2009.

their communications with other nodes by using DSMIPv6 only, regardless the visiting network is IPv4 or IPv6.

There are already some implementations of DSMIPv6, such as UMIP-DSMIP [8] and implementation for SHISA [9]. Although the two implementations realized the basic features of DSMIPv6, both of them have some features not implemented yet, such as supporting private IPv4 care-of address. In this paper, we mainly describe our implementation of DSMIPv6 to support mobile nodes move to IPv4 networks and use private IPv4 care-of addresses. We also performed experiments by using our implementation and the results are shown in this paper too. This paper is organized as follow. We give an overview of DSMIPv6 and its existing implementations in Sec.2. We then present the design of our implementation in Sec.3. We performed experiments by using our implementation and the results are shown in Sec.4. In Sec.5, we give the conclusion of this paper.

2 DSMIPv6

2.1 Principle of DSMIPv6

Dual stack node is a node that supports both IPv4 and IPv6 networks. In order to use DSMIPv6, dual stack mobile nodes need to manage an IPv4 and IPv6 home or care-of address simultaneously and update their home agents' bindings.

The basic features of DSMIPv6 are: the possibility to configure an IPv4 care-of address on the mobile node, the possibility to connect the mobile node behind a NAT device, the possibility to connect the home agent behind a NAT device, the possibility to use an IPv4 Home Address on the mobile node.

A mobile node, which is dual stack node, has both IPv4 and IPv6 home addresses. Home agent is connected to both IPv4 and IPv6 networks. When in IPv4 network, mobile node gets an IPv4 care-of address and registers it to home agent. Both IPv4 and IPv6 home addresses are bound to the address. IPv4 traffic goes through IPv4-in-IPv4 tunnel between mobile node and home agent. IPv6 traffic goes through IPv6-in-IPv4 tunnel. In a similar way, if mobile node moves to IPv6 network and registers IPv6 care-of address, traffic goes through IPv4-in-IPv6 tunnel or IPv6-in-IPv6 tunnel. NAT devices between mobile node and home agent can be detected.

2.2 Implementations of DSMIPv6

There are already some implementations of DSMIPv6, such as UMIP-DSMIP [8] and implementation for SHISA [9]. UMIP-DSMIP is an implementation of DSMIPv6 for the UMIP stack. It is developed by Nautilus6 [8] and is shipped as a set of patches for UMIP 0.4 and 2.6.24 kernels. The implementation for SHISA introduced in [9] is based on BSD operating systems.

Although some basic features of DSMIPv6 are realized in the two implementations, both of them have some features not implemented yet, such as Network Address Translator (NAT) [10] Traversal feature. Since public IPv4 addresses are not enough for use, NAT is widely used to solve this problem. Suppose a mobile node moves to GPRS or CDMA network, it can only get private IPv4 care-of addresses. To use the care-of addresses to maintain the communication, NAT traversal feature must be realized in this situation. Our implementation includes this feature.

The MIPv6 implementations based on are also different, UMIP-DSMIP is based on UMIP stack and the other is based on SHISA. Our implementation is based on MIPL (Mobile IPv6 for Linux), which is open source under GNU GPL. MIPL is developed by HUT to run on Linux environments that support IPv6. The latest version of MIPL is 2.0.2, which requires the kernel version of 2.6.16.

3 Implementations

3.1 Overview

We implemented the basic features of DSMIPv6 including the possibility to configure an IPv4 care-of address on the mobile node and the possibility to connect the mobile node behind a NAT device. By using our implementation, mobile node can move to IPv4-only network and keep the communications. Private IPv4 care-of address is supported. Fig.1 shows the supported scenes. Our implementation is based on the version of MIPL 2.0.2.

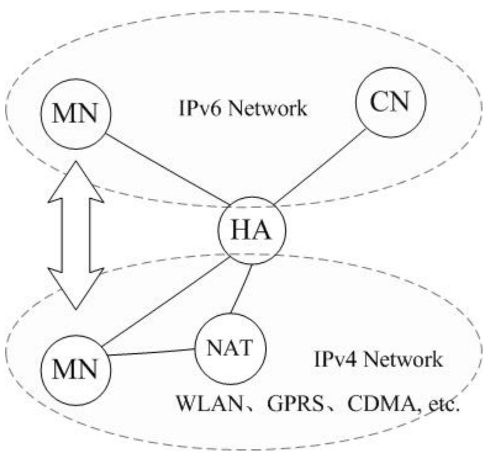

Fig. 1. Supported scenes of our implementation

When a mobile node moves to IPv6 network and gets an IPv6 care-of address, the basic MIPL works. In addition to this, it should update the new members of binding update list and binding cache described in 3.3 and 3.4.

If the mobile node moves to IPv4 network and gets an IPv4 care-of address, it needs send a binding update message to home agent's IPv4 address. The binding update message contains the mobile node's IPv6 home address in the home address option. However, since the care-of address is IPv4 address, the mobile node must include its IPv4 care-of address in the IPv6 packet. After accepting the binding update message, the home agent will update the related binding cache entry or create a new binding cache entry if such entry does not exist. The binding cache entry points to the mobile node's IPv4 care-of address. All packets addressed to the mobile node's home address will be encapsulated in a tunnel that includes the home agent's IPv4 address in the source address field and the mobile node's IPv4 care-of address in the destination address field. After creating the corresponding binding cache entry, the home agent sends a binding acknowledgment message to the mobile node.

3.2 Tunnel

In the implementation of MIPL, when mobile node moves to IPv6 foreign network, an IPv6-in-IPv6 tunnel will be created, whose name is "ip6tnl". All packets sent to mobile node's home address will be encapsulated to this tunnel from home agent to mobile node's care-of address. When mobile node moves to IPv4 foreign network, we should setup an IPv6-in-IPv4 tunnel to transmit packets between mobile node and home agent. However, if mobile node's IPv4 care-of address is private, the communication link between mobile node's care-of address and home agent will consist of NAT devices. Since common NAT devices do not support IPv6-in-IPv4 packets (which type is 41), we should tunnel IPv6 packets in UDP and IPv4, i.e. IPv6-in-UDP tunnel. Fig.2 shows the differences between the two tunnels.

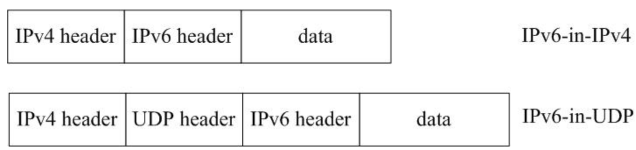

Fig. 2. Differences between the two type tunnels

Linux kernel realized the IPv6-in-IPv4 tunnel, which named "sit". It needs to configure the two end-points before using it to transmit packets. We modified the codes to realize the tunnel IPv6-in-UDP by adding UDP header between IPv4 header and inner IPv6 header before transmitting. After receiving tunnel packets, pull out the UDP header as well as the IPv4 header.

There are two reasons for tunnel's implementation at the kernel space. Firstly, it has good efficiency; Secondly, the encapsulation and decapsulation are the common function of mobile node, home agent and access router. Separating the function with others and modeling of tunnel just provide the common interface to upper applications, so that applications do not need consider the tunnel. This method can enhance the expand property.

3.3 Mobile Node Modifications

When IPv4 care-of address is detected, IPv4 care-of address option is created and included in the Mobility Header including the binding update message sent from the mobile node to home agent. Before sending binding update message, mobile node sets the sit tunnel in itself and sends signal to home agent to let it set another end-point of tunnel. After receiving the successful message to set tunnel in home agent, mobile node create or update a binding update list entry for its home address. Then, binding

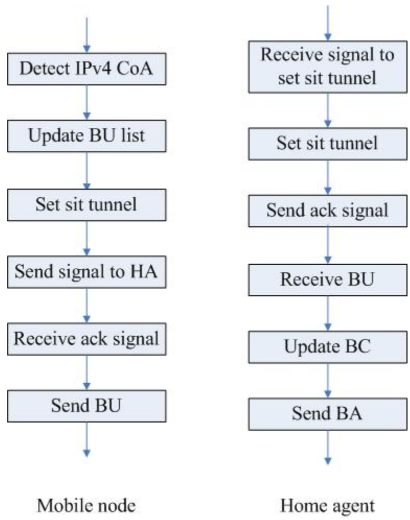

Fig. 3. Process flow for IPv4 care-of address

update message is encapsulated in IPv6-in-UDP tunnel and sent to home agent if everything is successfully processed. The process flow is shown in Fig.3.

Based on the structure of binding update list in MIPL, we add two members, one is to store IPv4 care-of address, the other is to mark the current type of care-of address: IPv4 or IPv6. By doing this, we use only one entry for one home address.

After receiving the binding acknowledgement message, mobile node updates the corresponding bind update list entry to change the binding status. If the binding acknowledgement shows the binding failure, should delete the tunnel set before.

3.4 Home Agent Modifications

Home agent processes the received binding update message, create or update a binding cache entry. IPv4 address acknowledgement option is created and included in the Mobility Header including the binding acknowledgement message sent from home agent to mobile node. This option indicates whether a binding cache entry was created or updated for the mobile node's IPv4 care-of address. If home registration failed, should delete the tunnel set before. The process flow is shown in Fig.3.

Like the binding update list, we add two members to the structure of binding cache too. One is to store the IPv4 care-of address; another is to mark the current type of care-of address.

4 Evaluations

4.1 Testbed Details

The testbed's network topology is shown as Fig.4. The column are PC router, which is exploited on Linux platform. CN, which is corresponding node, is a PC connected

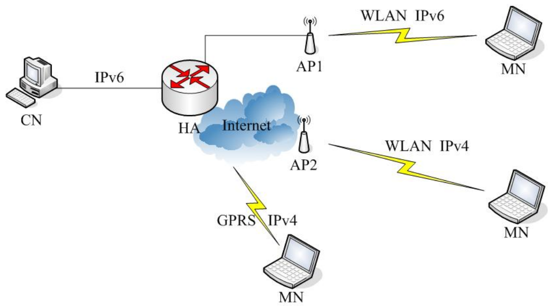

Fig. 4. Network topology of testbed

with home agent (HA) via IPv6 network. MN, which is mobile node, moving between home network and different foreign networks, is a laptop with Linux OS. Protocol is IPv6 in home network and IPv4 in foreign network. Modified MIPL is running in MN and HA.

MN has IPv6 home address and can get both IPv6 and IPv4 care-of address. CN has only IPv6 address and communicates with MN. HA has both IPv6 and IPv4 addresses, in which IPv4 address is public and is known by MN. MN can get private IPv4 address as its care-of address in WLAN or GPRS network.

Initially MN is at home network, it communicates with CN via IPv6. After moving to foreign network, WLAN or GPRS, it gets IPv4 care-of addresses. MN will back to home network and use IPv6 address too.

4.2 RTT (Round Trip Time) Test

We use "ping6" command to test the RTT and perform 200 times to get the mean values. Table.1 is the RTT comparison in different networks, separated by the scenes that MN in home network and in foreign network.

Table 1. RTT comparisons in different networks

	MN in home network.	MN in WLAN.	MN in GPRS
Mean RTT (ms)	2.559	223.6	615.3

The RTT in home network is smallest in the three network since MN is directly linked with HA. The RTT in GPRS network is the biggest, for the packets are tunneled and pass more hops in the road. The results confirmed that the implementation of DSMIPv6 worked as our expected, most of all, connection was maintained by private IPv4 address and tunnel in WLAN and GPRS.

4.3 Handover Delay Test

The "crontab" tool of Linux is used to implement handover automatically. And we collect 100 times' handover to analyze. The results are listed in Table 2.

Table 2. Handover delays in different situations

	Home to WLAN.	WLAN to GPRS.	GPRS to Home
Mean time (s)	1.942	2.038	2.846

As Table 2 shows, the delay of moving from home network to foreign network and moving between different foreign networks is approximate 2 seconds. The delay of moving from foreign network to home network is a little bigger, for there is more operations while MN back home.

These tests also confirmed the validation of our implementation. We also run services, such as ftp and video, to test the effect of the handover between networks. The interval was not very obvious, and didn't cause a bad effect for user experience.

5 Conclusions

This paper describes the DSMIPv6 implementation on MIPL for Linux systems. The DSMIPv6 implementation was designed to support mobile node moving between IPv4 and IPv6 networks while maintaining the established communications. An IPv6-in-UDP tunnel is setup to transmit IPv6 packets via IPv4 link and traverse NAT devices. An IPv4 care-of address and an address flag are added to binding update list and binding cache, function of which are separately storing IPv4 care-of address and marking the address type. An IPv4 care-of address option and IPv4 address acknowledgement option are used to register IPv4 care-of address to home agent. By evaluating the implementation, it is confirmed that the DSMIPv6 implementation works as expected. It is thus said that the specification is stable to work in true environment.

References

1. Johnson, D.: Mobility Support in IPv6, RFC3775, the Internet Engineering Task Force (June)
2. Mobile IPv6 for Linux, `http://www.mobile-ipv6.org/`
3. Networking and security center.: Sun Microsystems Laboratories, `http://playground.sun.com`
4. Busaranun1, A., Pongpaibool, P., Supanakoon, P.: Handover Performance of Mobile IPv6 on Linux Testbed. ECTI-CON (2006)
5. Lei, Y.: The Principle of Mobile IPv6 and Deployment of Test Bed. Sciencepaper Online
6. Tsirtsis, G., Soliman, H.: Mobility management for Dual stack mobile nodes, A Problem Statement, RFC 4977, The Internet Engineering Task Force (August 2007)
7. Soliman, H. (ed.): Mobile IPv6 support for dual stack Hosts and Routers (DSMIPv6), Internet Drafts draftietf-mip6-nemo-v4traversal-02, The Internet Engineering Task Force (June 2006)
8. UMIP-DSMIP, `http://www.nautilus6.org/`
9. Mitsuya, K., Wakikawa, R., Murai, J.: Implementation and Evaluation of Dual Stack Mobile IPv6, AsiaBSDCon 2007 (2007)
10. Egevang, K.: The IP Network Address Translator (NAT), RFC1631, The Internet Engineering Task Force (May 1994)