SHISA: The IPv6 Mobility Framework for BSD Operating Systems

Keiichi Shima
Internet Initiative Japan Inc. *

Ryuji Wakikawa, Koshiro Mitsuya, Keisuke Uehara
Keio University †

Tsuyoshi Momose
NEC Corporation ‡

Abstract

Mobile IPv6 and Network Mobility Basic Support (NEMO BS) are the IETF standard mobility protocols for IPv6. When we consider the deployment of a new protocol, it is well understood that the existence of a free protocol implementation, which can be used as a reference implementation for both research and operation, plays important roles. SHISA is a free implementation of Mobile IPv6 and NEMO BS protocols built on top of BSD operating systems. The purpose of SHISA is to be a reference implementation of the mobility protocols and to accelerate the deployment. In this paper, we explain the background of the development, introduce the design and explain the implementation detail. SHISA consists of the modified kernel and the user space programs, which is a similar approach as the routing/forwarding mechanism implemented in many UNIX systems. We designed a new communication layer between kernel and a user space program and also between user space programs to exchange mobility related information. This design makes the implementation simple and extensible. SHISA also provides some advanced features such as multiple care-of address registration and IPv4 prefix support which will help the deployment in a real situation.

1 Introduction

The rapid growth of the IPv4 Internet raised a concern of the IPv4 address exhaustion. IPv6 was designed as the essential solution of the problem. We are now on the transition period from the IPv4 Internet to the IPv6 Internet. As a result, a vast number of IPv6 devices connected to the Internet using various communication technologies will appear in the future. The devices will not only be computers and PDAs but also cars, mobile phones, sensor devices and so on. Since many devices will potentially move around changing its point of attachment to the Internet, mobility support for IPv6 is considered necessary. The IETF has discussed the protocol specification and finally standardized two IPv6 mobility protocols, Mobile IPv6 [5] for host mobility and Network Mobility Basic Support (NEMO BS) [1] for network mobility. The status of these two protocols is currently "Proposed Standard" which means they are under verifying that if there is no significant problems. To become the final "Standard" status, the protocol must be verified its functions and inter-operability with 2 or more independent implementations and must have significant operational experience. We decided to provide a high quality free implementation to help the deployment of the IPv6 mobility protocols and to contribute the future mobile Internet. We designed the mobility stack, called *SHISA*[1], which supports both host mobility and network mobility to provide a full featured stack on top of BSD operating systems. In this paper, we will provide the basic knowledge of Mobile IPv6 and NEMO BS in Section 2 and discuss the design principal and implementation detail in Section 3 and Section 4. The future plan of our project is introduced in Section 5 and Section 6 concludes this paper.

2 Overview of Mobile IPv6 and NEMO BS

Mobile IPv6 is a protocol which adds a mobility function to IPv6. In Mobile IPv6, a moving node (*Mobile Node*, *MN*) has a permanent fixed address which is called a *Home Address* (*HoA*). HoA is assigned to the MN from the network to which the MN is originally attached. The network is called a *Home Network*. When

*Jinbo-cho Mitsui Bldg., 1-105, Kanda, Jinbo-cho, Chiyoda-ku, Tokyo 101–0051 Japan, Email: keiichi@iijlab.net
†5232, Endo, Fujisawa, Kanagawa 252–8520 Japan, Email: ryuji@sfc.wide.ad.jp, mitsuya@sfc.wide.ad.jp, kei@sfc.wide.ad.jp
‡7-1, Shiba 5-chome, Minato-ku, Tokyo 108–8001 Japan, Email: momose@az.jp.nec.com

[1]SHISA was named after a traditional roof ornament in Okinawa Japan, where we had the first design meeting.

Proceedings of the International Multi-Conference on Computing in the Global Information Technology (ICCGI'06)
0-7695-2629-2/06 $20.00 © 2006 IEEE

0-7695-2629-2/06/$20.00 (c) 2006 IEEE

the MN moves to other networks than the home network, the MN sends a message to bind its HoA and the address assigned at the foreign network. The message is called a *Binding Update* (*BU*) message. The address at the foreign network is called a *Care-of Address* (*CoA*) and the networks other than the home network is called *Foreign Networks*. The message is sent to a special node, called a *Home Agent* (*HA*) located in the home network. A bi-directional tunnel between the HA and the CoA of the MN is established after the binding information has been successfully exchanged. All packets sent to the HoA of the MN are routed to the home network by the Internet routing mechanism. The HA intercepts the packets and forwards them to the MN using the tunnel. Also, the MN sends packets using the tunnel when communicating with other nodes. The communicating nodes do not need to care about the location of the MN, since they see the MN as if it is attached to the home network. Figure 1 illustrates the operation of Mobile IPv6.

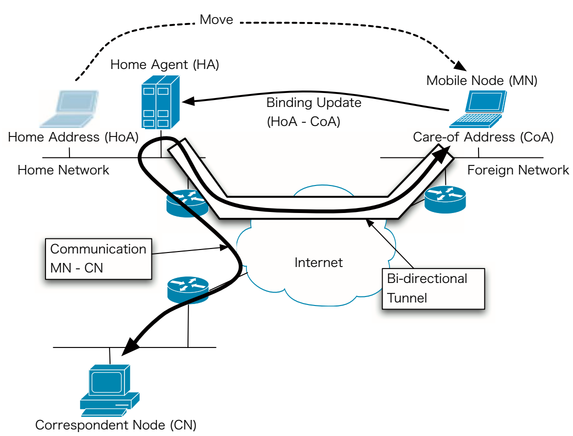

Figure 1. Basic Operation of Mobile IPv6

In Figure 1, the communication path between MN and its peer node is redundant since all traffic is forwarded via HA. Mobile IPv6 provides an optimized way to communicate with an IPv6 node which is aware of Mobile IPv6 protocol. Such a node is called a *Correspondent Node* (*CN*). The MN can also send a BU message to CN. When sending the BU message, the MN must perform a simple address ownership verification procedure, which is called the *Return Routability* (*RR*) procedure. The MN sends two messages to the CN, one from its HoA and the other from its CoA. The CN replies these two messages with cookie values. The MN generates a secret using these two cookies transmitted via different paths and sends a BU message protected by the secret. Once the CN receives the BU message, the MN can send a packet to the CN from its CoA. To provide HoA information to CN, MN stores it in a Destination Options Header as the *Home Address option* (*HAO*). The option is newly defined in the Mobile IPv6 specification. The CN can also send a packet directly to the MN using the *Type 2 Routing Header* (*RTHDR2*) newly defined routing header. This direct communication is called *Route Optimized* (*RO*) communication.

NEMO BS is an extension of Mobile IPv6. The basic operation of a moving router (*Mobile Router*, *MR*) is same with that of MN except the MR has a network (*Mobile Network*) behind it. The network prefix is called a Mobile Network Prefix (*MNP*). A node in the mobile network, which is called a *Mobile Network Node* (*MNN*), can communicate with other nodes as if they are attached to the home network, thanks to the tunneling between the HA and the MR. NEMO BS does not provide the RO feature. Figure 2 depicts the operation of NEMO BS.

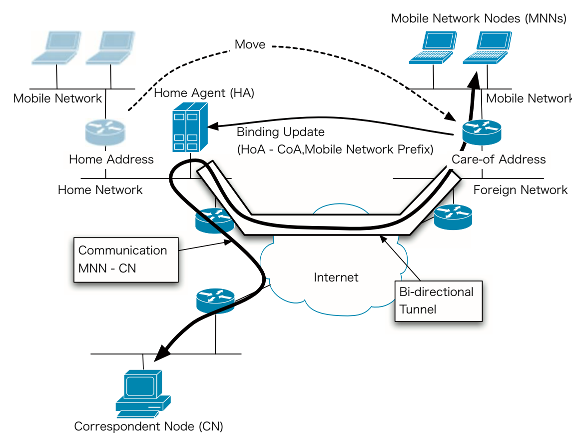

Figure 2. Basic Operation of NEMO BS

3 SHISA Design Principal

We defined the requirements of a mobility stack as follows.

- Separation of signaling and forwarding:
  The operation of Mobile IPv6 and NEMO BS is basically IP packet routing (forwarding or tunneling). To get better performance, packet routing should be done in kernel space. The signal processing should be done in user space, since the process is complex and it is easier to modify/update user space programs than kernel. This separation

will give the stack both good performance and efficiency in developing the stack.

- Flexibility of network device selection:
  We cannot predict the network devices that MN uses. To be able to adapt various network devices, we have to provide an abstraction layer by defining a logical network device of the MN that can be bound to the actual physical network devices.
- Adaptability to various movement scenarios:
  The movement assumption varies based on the demands of operators. The signaling mechanism and the movement detection mechanism must be separated. Because the signaling procedure usually never be changed unless the protocol specification does not change. The movement detection part must be replaceable depends on operators.
- Extensibility:
  IPv6 mobility technologies are one of most active area in the Internet research world. The design must provide an easy framework that can support future advanced functions.
- Minimum modification on existing kernel functions:
  We believe the mobility function will be a core function in future operating systems. To integrate the implementation to the target operating system, in our case, BSD operating systems, we have to minimize the modification on existing kernel functions.

In the following sections, we will discuss the implementation decisions we have made based on the above requirements.

4 SHISA Implementation

SHISA is implemented on top of the KAME IPv6 stack [4]. The KAME IPv6 stack is an IPv6 stack for BSD operating systems. Since it covers most of IPv6 functions and API functions necessary for SHISA, we can just implement mobility related part only by using the KAME IPv6 stack as a base system.

4.1 Supported Features

We provide the following functions currently:

- Mobile IPv6 functions for MN, HA and CN
- NEMO BS functions for MR and HA
- Multiple Care-of Address registration support [12]
- IPv4 Mobile Network Prefix support [8]

4.2 Program Organization

SHISA consists of several user space programs and the modified kernel. Table 1 shows the programs used by the SHISA stack and Table 2 lists the necessary program modules used by each type of node.

mnd	Provides the MN functions
had	Provides the HA functions
cnd	Provides the CN functions
babymdd	A simple movement detector of MN
mrd	Provides the MR functions
nemonetd	Provides the tunnel setup functions for NEMO BS

Table 1. SHISA programs

Based on the node type, one or several SHISA programs run on a node. In addition, a user can choose to drop or replace functions by stopping or changing programs. For example, if one does not need the CN functions when operating MN, he can stop the **cnd** program. Also one can replace **babymdd** program to other movement detection program which may be specialized to the user's network and device of the MN used in his network environment.

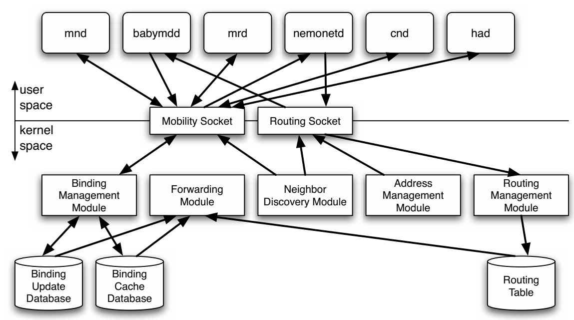

Figure 3. SHISA system configuration

Figure 3 shows the system configuration of the SHISA stack. SHISA programs communicate with kernel and other programs using the Mobility Socket dis-

Node type	Used programs
Mobile node	**mnd**, **babymdd**, **cnd**
Mobile router	**mrd**, **nemonetd**, **babymdd**
Home agent	**had**, **cnd**
Correspondent node	**cnd**

Table 2. SHISA programs categorized by the node types

Proceedings of the International Multi-Conference on Computing in the Global Information Technology (ICCGI'06)
0-7695-2629-2/06 $20.00 © 2006 IEEE
IEEE COMPUTER SOCIETY

cussed in Section 4.3. Events of address change used by a movement detector are notified from the Neighbor Discovery and the Address Management modules through the Routing Socket. The signaling messages are exchanged between the SHISA programs of two nodes, for example, **mnd** on MN and **had** on its HA. The binding information retrieved from the signal exchange is stored in the kernel as the Binding Update database for MN/MR and the Binding Cache database for HA/CN. These databases are used when the kernel sends or forwards packets based on the current location of the MN/MR.

4.3 Mobility Socket

The operation of Mobile IPv6 and NEMO BS is similar to existing routing operation. When we started the design of SHISA, we referred the existing BSD Routing Socket mechanism [10]. In the routing mechanism, the role of the kernel is to keep the routing table and forward packets based on the table. The kernel itself does not exchange any routing information. The information exchange is done by routing daemon programs running in user space. This design has several benefits. For example, we can avoid to implement a complex routing algorithm in kernel space, where a trivial mistake sometimes causes a serious problem like a kernel panic. Also debugging user space programs is easier than debugging a kernel, because we can utilize many advanced debugger programs.

We took a similar approach that the routing mechanism did. The kernel keeps a table for binding information and transmits packets based on the table. The update of the table is done by the user space programs which run on MN, HA and CN. We designed a new socket, called *Mobility Socket* (*MIPSOCK*) [6], to exchange the mobility related information between the kernel and user space programs. It might be possible to reuse the Routing Socket to exchange mobility information, however we did not choose the approach because of the following two reasons.

- To minimize the impact to the existing programs which rely on the Routing Socket
- Information semantics is different from routing information

We already have several programs which use the Routing Socket. The routing function is one of core functions for IP nodes to connect to the Internet. We must avoid to cause any problems by adding new messages to the Routing Socket. One of our goals is to merge our achievement to the original BSD distribution. Importing a big change to the existing stable functions is usually difficult. Designing a separate mechanism for a new feature seemed to be reasonable. Also, not all the information we exchange using MIPSOCK between SHISA programs and the kernel are routing related information. For example, the kernel sends a hint message to user programs that the node receives a tunneled packet. Another example is a hint message from the kernel to user space programs to insist sending an protocol error message. In that sense, using the Routing Socket does not seem to be appropriate.

`MIPM_BC_ADD`	Add/Update binding information on HA or CN
`MIPM_BC_REMOVE`	Remove binding information on HA or CN
`MIPM_BUL_ADD`	Add/Update binding information on MN or MR
`MIPM_BUL_DELETE`	Remove binding information on MN or MR
`MIPM_MD_INFO`	Notify movement with a new CoA to MN or MR
`MIPM_HOME_HINT`	Notify returning home to MN or MR
`MIPM_RR_HINT`	Notify there is tunneled traffic to MN
`MIPM_BE_HINT`	Notify a protocol error occurred in the kernel and insist to send an error message to a peer node

Table 3. MIPSOCK messages

A subset of MIPSOCK messages are listed in Table 3. The important point is that, these messages are not only used for communications between user space programs and the kernel. The messages are also used between user space programs to exchange information. This mechanism provides an easy way to get the current situation of the mobility stack asynchronously and provides extensibility to other programs related to mobility. For example, `MIPM_MD_INFO` is issued by the `babymdd` program and is received by the `mnd` or the `mrd` program. Another examples is `MIPM_BUL_ADD` and `MIPM_BC_ADD` messages. They are issued by the `mnd` and `had` to add binding information. When we extended the NEMO BS feature, what we had to do is to implement `nemonetd` which monitors these messages and establishes a bi-directional tunnel. Thanks to MIPSOCK, we could minimize the modification to `mnd` to support the NEMO BS feature.

4.4 Movement Detection

Movement detection is the most difficult part when we implement a mobility stack. Although the movement detection should be ideally done in IP layer, the performance of the detection often slow. We may need several seconds to detect movement. This delay is sometimes critical for real-time applications. It is known that if we use more information, e.g. layer 2 information, we can detect movement much faster. The movement detection module in our implementation is designed as a replaceable module. The user of our stack can implement a special movement detection module based on the scenario and the background layer 2 technologies.

4.5 Packet Forwarding

The performance of forwarding module will impact the total performance of the stack. We implemented the forwarding procedure in kernel space while all signaling procedures are implemented in user space. Doing all forwarding tasks in kernel will reduce the overhead of forwarding tasks compared to the case when they are done in user space, because passing packets from kernel space to user space (and vice versa) usually requires memory copy operation which processing costs is high.

A mobility stack requires the following functions with regard to packet forwarding:

- Inserting HAO on an outgoing packet by MN if binding information exists
- Inserting RTHDR2 on an outgoing packet by HA/CN if binding information exists
- Forwarding packets to HA using a tunnel interface by MN/MR if binding information does not exist
- Forwarding packets to MN/MR using tunnel interfaces by HA if binding information does not exist

The first two operation involves the operation of IPv6 extension headers implemented in kernel space. HAO is one of Destination options which hold the HoA of MN. We extended the processing function of the Destination options to handle HAO. RTHDR2 is newly defined routing header in Mobile IPv6 for RO. The basic format of RTHDR2 is same with the Type 1 Routing Header (RTHDR1) defined in the IPv6 specification, except some limitations. RTHDR2 has different type number from RTHDR1 and can only have either one of HoA or CoA of MN. We extended the existing Routing Header manipulation code to support RTHDR2.

The bi-directional tunnel is used when communication is not optimized. This situation can be detected by the kernel to check if there is valid binding information in binding databases. When the kernel does not find binding information, packets are tunneled just before the output function to the network interface driver on the MN/MR side. On the HA side, packets addressed to MN/MR and MNNs are once passed to the forwarding function, and in the function they are tunneled to the MN/MR using the tunnel interface between the HA and the MN/MR if no binding information is found.

4.6 Multiple Care-of Address

In the Mobile IPv6/NEMO BS specifications, it is impossible for MN/MR to register multiple CoAs at the same time. However considering the recent progress of wireless technologies, it is getting more common for mobile devices to have multiple network interfaces. Using multiple interfaces enables efficient usage of network property/bandwidth and increases fault tolerance in case of network problems [2]. We have implemented the Multiple Care-of Address support protocol [12]. Currently, this function is available only for MR. The extension is implemented by adding an unique identifier field to the `MIPM_BUL_ADD`, the `MIPM_BUL_ADD` and the `MIPM_MD_INFO` messages and adding a filed to keep the identifier in binding databases. The identifier is bound to each CoA assigned to the MR. Usually HA routes packets based on the HoA of each MR. In this case, the HA cannot distinguish the binding between (HoA, CoA1) and (HoA, CoA2). In this extension, each MR is identified by the pair of HoA and the identifier. In the SHISA implementation, the identifier is mapped to a tunnel interface. If MR has two network interfaces for CoAs and registers these two CoAs at the same time, the MR and its HA will have two tunnel interfaces between them, each tunnel is bound to each identifier assigned to the network interfaces of the MR. The MR and the HA can utilize these tunnels based on local policy. For example, they can use one tunnel as a primary interface and the other for backup. Or, if they have two interfaces which properties are different, e.g. one interface is low-bandwidth and low-latency, the other is high-bandwidth and high-latency, then they may use the former for urgent messages and the latter for data transmission. Since we have implemented this mechanism as a tunnel interface, we can use the basic packet filtering mechanism, such as IP Filter [7] to distribute traffic.

4.7 IPv4 Mobile Network Prefix

When NEMO BS was specified, most of people involved in the discussion did not think they would need IPv4 NEMO support. However, considering the current situation that IPv6 deployment requires more time than we originally expected, some kinds of IPv4 support must be necessary. We proposed a mechanism to carry IPv4 traffic over the tunnel interface created by NEMO BS mechanism [8]. With this mechanism, MR can have an IPv4 MNP in addition to an IPv6 MNP. The benefit of this mechanism is that the user of the MR that support this extension can operate IPv4 networks over the IPv6 only infrastructure. Such kind of operation encourages the existing IPv4 users to move IPv6 infrastructure [9], since NEMO BS can provide fault tolerance and can provide load-balance or traffic engineering using the Multiple CoA support as discussed in Section 4.6. SHISA is extended to keep IPv4 MNPs in its binding database and extended to forward IPv4 packets, which prefix is registered as a part of MNP, using the tunnel interface established between the MR and its HA.

5 Future Plans

The intention of the SHISA development is to provide a reference implementation of Mobile IPv6/NEMO BS. We are planning to complete the implementation and get the certificate logo issued by the IPv6 Forum [3]. Getting the logo indicates the implementation is fully compliant and inter-operable with other implementations. Also, we will continue to support additional features which are necessary to satisfy future requirements and to support IP mobility deployment. For example, the multiple CoA support is now being revised in the IETF Monami6 working group. We will follow the discussion and will provide the implementation for research purpose and achieving operational experience. The IPv4 mobile network prefix support is now integrated to the Dual Stack Mobile IPv6 (DSMIPv6) solution [11] which not only supports IPv4 MNPs but also IPv4 CoA that enables MN/MR to attach the IPv4 Internet. This technology will accelerate the deployment of both IPv4/IPv6 mobility technology. We are developing the DSMIPv6 and soon merge it to SHISA. Operational technologies are also important. AAA interaction and a home agent distribution mechanism are planned to be implemented.

6 Conclusion

SHISA is an implementation of IETF standard IPv6 mobility protocols (Mobile IPv6/NEMO BS) built on top of BSD operating systems. We process all mobility signals in user space to make it easy to extend. To pass the signals we designed a new communication socket mechanism, Mobility Socket. This design makes it possible to implement a complex signaling processing in user space and to implement packet forwarding, that is performance sensitive, in kernel simultaneously. SHISA is verified to be inter-operable with other implementations at several inter-operability test events, those are developed independently . We also support advanced specifications such as Multiple Care-of Address and IPv4 prefix support that are useful for real operation. The development will continue to make it more compliant with the specification and to support mobility deployment.

References

[1] V. Devarapalli, R. Wakikawa, A. Petrescu, and P. Thubert. Network Mobility (NEMO) Basic Support Protocol. Technical Report RFC3963, IETF, January 2005.

[2] T. Ernst, N. Montavont, R. Wakikawa, C. Ng, and K. Kuladinithi. Motivations and Scenarios for Using Multiple Interface and Global Addresses. Technical Report draft-ietf-monami6-multihoming-motivation-scenario-00, IETF, February 2006.

[3] IPv6 Forum. Web page. http://www.ipv6-forum.org/.

[4] T. Jinmei, K. Yamamoto, J. Hagino, S. Sakane, H. Esaki, and J. Murai. The IPv6 Software Platform for BSD. *IEICE Transactions on Communications*, E86-B(2):464–471, February 2003.

[5] D. B. Johnson, C. E. Perkins, and J. Arkko. Mobility Support in IPv6. Technical Report RFC3775, IETF, June 2004.

[6] T. Momose, K. Shima, and A. Tuominen. The application interface to exchange mobility information with Mobility subsystem (Mobility Socket, AF_MOBILITY). Technical Report draft-momose-mip6-mipsock-00, IETF, June 2005.

[7] D. Reed. IP Filter. Web page. http://coombs.anu.edu.au/~avalon/.

[8] K. Shima. IPv4 Mobile Network Prefix Option for NEMO Basic Support Protocol. Technical Report draft-shima-nemo-v4prefix-01, IETF, October 2005.

[9] K. Shima. The design and implementation of a dual-stack mobile network using IPv6 only network infrastructure. In *The First International Workshop on Network Mobility (WONEMO)*, January 2006.

[10] K. Sklower. A Tree-based Packet Routing Table for Berkeley UNIX. In *Proceedings of the Winter 1991 USENIX Conference*, pages 93–103. USENIX Association, January 1991.

[11] H. Soliman, G. Tsirtsis, V. Deverapalli, J. Kempf, H. Levkowetz, P. Thubert, and R. Wakikawa. Dual Stack Mobile IPv6 (DSMIPv6) for Hosts and Routers. Technical Report draft-ietf-mip6-nemo-v4traversal-01, IETF, March 2006.

[12] R. Wakikawa, T. Ernst, and K. Nagami. Multiple Care-of Addresses Registration. Technical Report draft-wakikawa-mobileip-multiplecoa-04, IETF, June 2005.