Design of an Application Based IP Mobility Scheme on Linux Systems

Kohei Tanaka†, Fumihito Sugihara‡, Katsuhiro Naito†, Hidekazu Suzuki*, Akira Watanabe*

†Faculty of Information Science, Aichi Institute of Technology, Toyota, Aichi 470-0392, Japan
{kohei, naito}@pluslab.org
‡Department of Electrical and Electronic Engineering, Mie University, Tsu, Mie 514-8507, Japan
*Graduate School of Science and Technology, Meijo University, Nagoya, Aichi 468-8502, Japan

Abstract -
Internet of Thing (IoT) systems have been attracting attention as one of the solutions for new services in the Internet. They usually employ a client-server model due to a difficulty of accessibility in practical networks. However, scalability will be a major issue in IoT systems because billion of IoT devices will be installed around the world in recent years. The authors have been proposed a new IP mobility mechanism called NTMobile (Network Traversal with Mobility) to realize end-to-end communication in IoT systems because end-to-end communication can improve system scalability by reducing traffic through servers. Conventional implementation employed a kernel module mechanism for NetFilter because the kernel module implementation is the best way of realizing high throughput performance. On the contrary, the kernel module should be maintained according to changes in NetFilter specifications. This paper designs an application based IP mobility scheme on Linux systems, where the developed IP mobility library can realize the IP mobility function in an application layer on Linux systems. As a result, developers can realize an end-to-end communication model by employing the enhanced IP mobility library. The proposed design ensures compatibility between the development library and the conventional NTMobile. Therefore, developers can select the implementation scheme according to the required performance.

Keywords: IP Mobility, Accessibility, Application library, Linux, NTMobile.

1 INTRODUCTION

Recent microcomputer boards implement some network interfaces to cooperate with another microcomputer boards[1], [2]. Almost all microcomputer boards are usually installed in a private network due to a security policy and limitation of assignable global IP addresses. Typical private networks prohibit accesses from the global Internet to a node in their private networks. Therefore, inter-connectivity is a big issue even if some applications should communicate with each other to realize their specific service. Additionally, operating systems select an interface to access to the Internet according to network condition of each interface and access policies when a device is a mobile node[3]. Therefore, a seamless connectivity scheme is also an important function.

IP mobility protocols are a solution to the requirement for inter-connectivity and seamless connectivity because they can realize continuous communication when an IP address for an interface changes due to switching of access networks[4]–[7]. They are classified into three types: IP mobility schemes for IPv4, IP mobility schemes for IPv6, and IP mobility schemes for IPv4 and IPv6. Mainstream of IP mobility schemes is for IPv6. On the contrary, the number of implementations for IPv4 is quite few though some mechanisms have been proposed [8], [9]. DSMIPv6(Dual Stack Mobile IPv6)[10] supports IPv4 and IPv6 networks. However, it still does not support the inter-connectivity between IPv4 and IPv6.

IPv4 is still the mainstream protocol under the present circumstances of the Internet. Therefore, private networks are usually used in practical IPv4 networks to reduce the number of required global IP addresses. Additionally, some Internet service providers start the service with large scale network address translator (LSN) in order to meet the shortage of IPv4 global addresses[11]. As a result, IP mobility in a private network behind NAT becomes an important issue.

The authors have developed a new IP mobility technology called NTMobile (Network Traversal with Mobility) [12]–[14]. The features of NTMobile are an IP mobility and an accessibility in both IPv4 and IPv6 networks. Therefore, each client of NTMobile can communicate with each other even when they use a different IP protocol version because they can communicate with virtual IP addresses that are independent addresses from physical IP addresses. NTMobile systems have some servers: account server (AS), direction coordinator (DC), notification server (NS), and relay server (RS). DC, RS and NS serve an IP mobility and an accessibility functions for each client, and AS serves an authentication service.

This paper proposes a new design of an application based IP mobility for NTMobile nodes. The original design can provide NTMobile functions by an application library instead of the special kernel module. Therefore, application programmers can obtain IP mobility and accessibility functions by using the proposed application library.

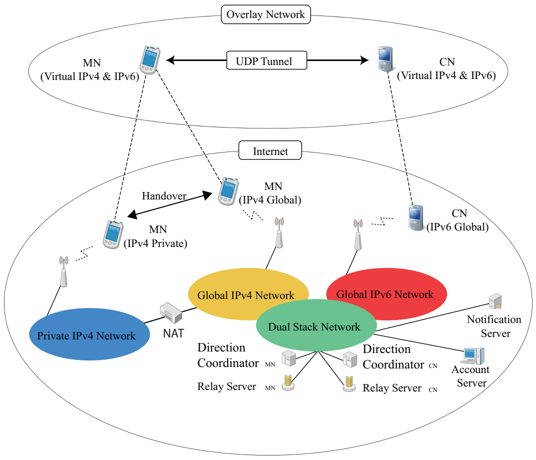

Figure 1: System model of NTMobile

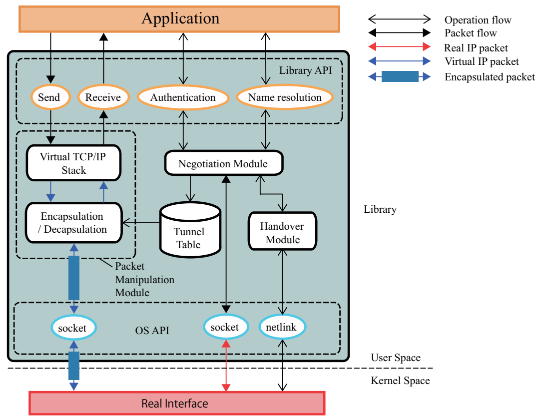

Figure 2: Overview of application library design

2 NTMOBILE

NTMobile can realize IP mobility and accessibility for IPv4 and IPv6 networks. Fig. 1 shows the overview of the NTMobile system model. The NTMobile system consists of an account server (AS), some direction coordinators (DCs) with some relay servers (RSs), notification servers (NSs) and NTMobile nodes. AS serves authentication service to all DCs, and each DC manages their RSs and NTMobile nodes. NTMobile node has IP mobility and accessibility functions by communicating with AS and its DC. Each DC has a virtual IP address pool for its NTMobile nodes, and assigns an address to each NTMobile node. Each NTMobile node constructs a UDP tunnel between NTMobile nodes according to a signaling direction from its DC, and communicate with each other by using their virtual IP addresses. As a result, each NTMobile node can communicate continuously even if real IP addresses at interfaces are changed because the virtual IP addresses are independent from physical IP addresses. The details of the system components are as followings.

2.1 Account Server (AS)

AS is an individual server that manages authentication information. Therefore, AS can distribute node information of each NTMobile node to initialize a setting for NTMobile nodes. Additionally, it bears responsibility for authentication by replying an authentication reply message when a DC makes inquiries about their NTMobile nodes authentication to AS. As a result, NTMobile nodes can get certified by AS through its DC.

2.2 Direction Coordinator (DC)

DC manages location information of each NTMobile node and indicates signaling processes for tunnel construction between NTMobile nodes. Each DC also owns the DNS(Domain Name Server) function. Therefore, DC can easily find another DC by searching a NS (Name Server) record in DNS. In addition, DC manages virtual IP addresses for NTMobile nodes, and assigns them to each NTMobile node without address duplication. In the tunnel construction process, it indicates the tunnel construction processes to both NTMobile nodes.

2.3 Notification Server (NS)

NTMobile typically uses a UDP protocol to communicate between a NTMobile node and its DC. Therefore, the NTMobile node should send keep-alive messages to its DC when it uses a private address under a NAT router. NS can provide a notification service with a TCP connection between NS and the NTMobile node, and can reduce the amount of messages for keep-alive.

2.4 Relay Server (RS)

The relay server function is to relay tunnels between NTMobile nodes when both NTMobile nodes exist under different NAT routers or exist under different version networks such as IPv4 and IPv6 networks. DC manages some relay servers to realize load balancing and to avoid a single point of failure. It also chooses a relay server to activate the relay function for dedicated NTMobile nodes.

2.5 NTMobile Node

Functions of NTMobile nodes are to realize IP mobility and accessibility in IPv4 and IPv6 networks. They obtain their own information from AS in the initialization phase, when they connect to the NTMobile network at first. Then, they inform their own network information to DC because DC should manage network information of each NTMobile node to realize IP mobility and accessibility. They also update the own network information when they switch access networks.

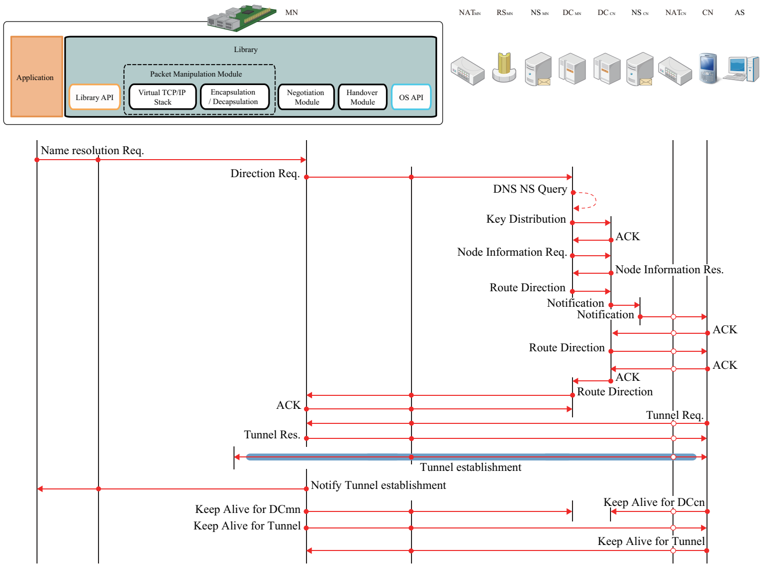

Figure 3: Signaling tunnel creation process of library

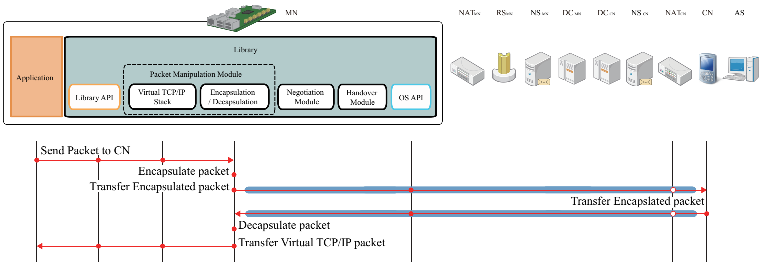

Figure 4: Signaling communication process of library

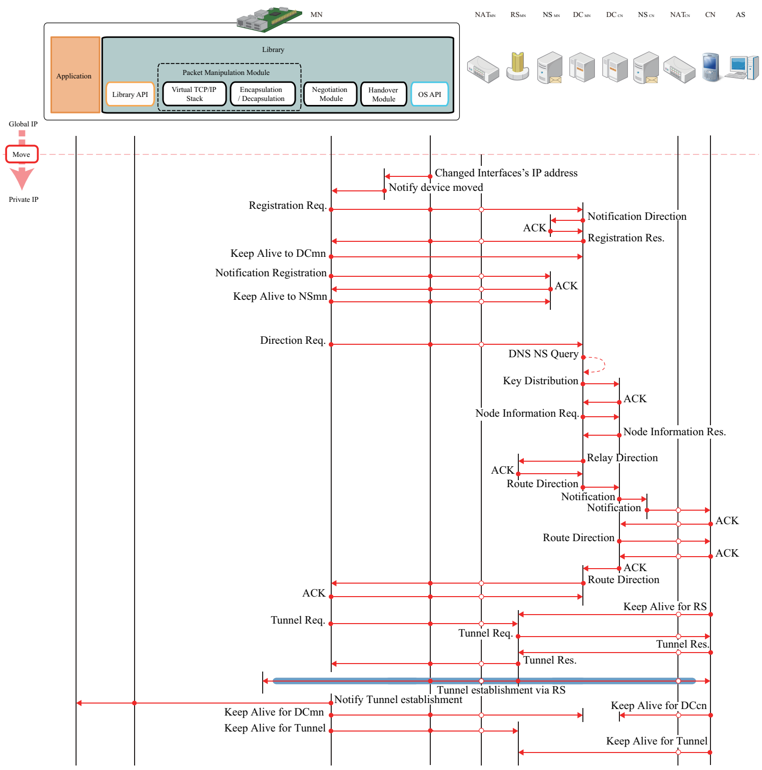

Figure 5: Signaling change network process of library

3 APPLICATION BASED IP MOBILITY

This paper proposes a new design of an application based IP mobility and accessibility for NTMobile nodes. The proposed design provides APIs (Application Programming Interface) for the NTMobile functions: an initialization, an authentication, a tunnel creation request, and network sockets for NTMobile communication. It also provides a virtual IP address for the network sockets. Our API provides BSD compatible network sockets for application developers. Therefore, application developers can easily implement their network application with virtual IP addresses for NTMobile network.

3.1 System Model

Fig. 2 shows the overview design of the proposed application based IP mobility and accessibility system for NTMobile nodes. The proposed application library is inserted between an original network socket in Linux OS and a developers' application. The application can request to the proposed application library for initialization of the NTMobile node functions, an authentication for NTMobile network, a tunnel creation request to a correspondent node. The actual data communication is performed with special network sockets for the NTMobile functions. The application library can perform IP capsulation/decapsulation and encryption/decryption processes between the special network sockets and the original Linux network sockets.

3.2 Library Modules

The proposed library consists of three main modules for negotiation, handover, and packet manipulation. The following is the functions of each module.

- Negotiation Module
  The negotiation module serves as a signaling function for NTMobile communication. Therefore, application developers do not care about the detail process of NTMobile communication because the negotiation module can handle IP mobility and accessibility functions in NTMobile. The negotiation module registers the specific information for packet manipulation into the tunnel table.

- Handover Module
  The handover module should check the network interface status to detect a change in the access network. It also handles reconstruction of a UDP tunnel to a correspondent node when IP address changes due to an access network change.

- Packet Manipulation Module
  The packet manipulation module consists of two main functions: a virtual TCP/IP stack and a capsulation function. The virtual TCP/IP stack provides a transport layer functions such as TCP and UDP to an application. The capsulation function handles the encapsulation and decapsulation for a UDP tunnel according to the information in the tunnel table.

3.3 Signaling Process

Figs. 3, 4 and 5 show the signaling processes for a tunnel creation in NTMobile. These figures are assumed that NTMobile node MN in a global IPv4 network requests to communication with NTMobile node CN in a private IPv4 network. Then, NTMobile node MN changes the network from the global IPv4 networks to a private IPv4 network. Direction coordinators $\mathrm{DC_{MN}}$ and $\mathrm{DC_{CN}}$ are the management servers for NTMobile node MN and CN respectively. The detail signaling process is as follows.

- Tunnel Creation

  Fig. 3 shows the signaling processes of the tunnel creation in NTMobile.

  1. The application calls the name resolution API to construct a UDP tunnel to NTMobile CN.
  2. The negotiation module requests the tunnel construction to $\mathrm{DC_{MN}}$ by transmitting the direction request message. The direction request message contains the FQDN of CN.
  3. $\mathrm{DC_{MN}}$ makes inquiries about NS record for the FQDN of CN since $\mathrm{DC_{CN}}$ is the domain name server and manages the domain for the FQDN of CN.
  4. $\mathrm{DC_{MN}}$ generates a shared key for encryption between $\mathrm{DC_{MN}}$ and $\mathrm{DC_{CN}}$ and distributes the shared key to $\mathrm{DC_{CN}}$.
  5. $\mathrm{DC_{CN}}$ replies the acknowledgement message to $\mathrm{DC_{MN}}$.
  6. $\mathrm{DC_{MN}}$ requests the information for CN to $\mathrm{DC_{CN}}$ by transmitting the NTM information request message.
  7. $\mathrm{DC_{CN}}$ replies the information about CN by replying the NTM information response message.
  8. $\mathrm{DC_{MN}}$ requests the tunnel construction to CN to $\mathrm{DC_{CN}}$ by transmitting the route direction message.
  9. $\mathrm{DC_{CN}}$ requests the notification to CN to $\mathrm{NS_{CN}}$.
  10. $\mathrm{NS_{CN}}$ sends the notification to CN based on the device type of CN.
  11. CN replies the acknowledgement message to $\mathrm{DC_{CN}}$.

12. DC_{CN} forwards the route direction message to CN.
13. CN replies the acknowledgement message to DC_{CN}.
14. DC_{CN} also replies the acknowledgement message to DC_{MN}.
15. DC_{MN} indicates the tunnel construction to CN to MN by transmitting the route direction message.
16. The negotiation module on MN replies the acknowledgement message to DC_{MN}.
17. CN transmits the tunnel request message to MN according to the direction from DC_{MN} because MN has a global IP address in this case.
18. MN replies the tunnel response message to CN to complete the tunnel construction process.
19. The negotiation module notifies the completion of the tunnel construction process to the application.
20. The negotiation module transmits a keep alive message to keep the NAT table on the route.
21. The application starts the communication through the constructed UDP tunnel. Then, it transmits packets by using special NTMobile socket.

- Data Communication

  Fig. 4 shows the signaling processes of data communication in NTMobile.

  1. The virtual TCP/IP stack in the packet manipulation module serves the transport layer function to the application.
  2. The encapsulation and decapsulation module performs the encapsulation the packet to CN.
  3. The encapsulated packet is transmitted to CN.
  4. The encapsulated packet is transmitted from CN.
  5. The encapsulation and decapsulation module performs the decapsulation the packet from CN.
  6. The virtual TCP/IP stack in the packet manipulation module sends the decapsulated packet to the application.

- Switching of Access Network

  Fig. 5 shows the signaling processes for switching the access network of MN.

  1. Linux kernel can notify the change of network configuration through netlink socket.
  2. The handover module notifies the negotiation module that the access network of the device has changed.
  3. The negotiation module registers the new information about the access network to DC_{MN} by transmitting the registration request message.
  4. DC_{MN} requests NS_{MN} to provide a notification service using TCP connection for MN.
  5. NS_{MN} replies the acknowledgement message to MN.
  6. DC_{MN} updates the network information of MN, and replies the registration response message.
  7. MN registers the information of MN to NS_{MN}.
  8. NS_{MN} replies the acknowledgement message to MN.
  9. The negotiation module requests the tunnel construction to DC_{MN} by transmitting the direction request message. The direction request message contains the FQDN of CN.
  10. DC_{MN} makes inquiries about NS record for the FQDN of CN.
  11. DC_{MN} may generate a shared key for encryption between DC_{MN} and DC_{CN} when the shared key is expired.
  12. DC_{MN} requests the information for CN to DC_{CN} by transmitting the NTM information request message.
  13. DC_{CN} replies the information about CN by replying the NTM information response message.
  14. DC_{MN} requests RS to relay the communication between both nodes.
  15. RS prepares the tunnel forwarding between both nodes, and replies the acknowledgement message to MN.
  16. DC_{MN} requests the tunnel construction to CN to DC_{CN} by transmitting the route direction message.
  17. DC_{CN} requests NS_{CN} to transmit a notification to CN.
  18. NS_{CN} sends the notification to CN according to the device type of CN.
  19. CN replies the acknowledgement message to DC_{CN}.
  20. DC_{CN} forwards the route direction message to CN.
  21. CN replies the acknowledgement message to DC_{CN}.
  22. DC_{CN} also replies the acknowledgement message to DC_{MN}.
  23. DC_{MN} indicates the tunnel construction to CN to MN by transmitting the route direction message.

24. The negotiation module transmits a keep alive message to keep the NAT table on the route.
25. MN transmits the tunnel request message to RS according to the direction from DC_{MN} because both nodes do not have a global IP address in this case.
26. RS transmits the tunnel request message to CN according to the direction from DC_{MN}.
27. CN replies the tunnel response message to RS to complete the tunnel construction process.
28. RS replies the tunnel response message to MN to complete the tunnel construction process.
29. The negotiation module notifies the completion of the tunnel construction process to the application.
30. The negotiation module transmits a keep alive message to RS to keep the NAT table on the route.

4 CONCLUSION

The authors have been proposed a new IP mobility mechanism called NTMobile (Network Traversal with Mobility) to realize end-to-end communication in IoT systems. This paper designs an application based IP mobility scheme on Linux systems, where the developed IP mobility library can realize the IP mobility function in an application layer on Linux systems. As a result, developers can realize an end-to-end communication model by employing the enhanced IP mobility library.

ACKKNOWLEDGEMENT

This work is supported in part by the Grant-in-Aid for Scientific Research (26330103, 15H02697), Japan Society for the Promotion of Science (JSPS) and the Integration research for agriculture and interdisciplinary fields, Ministry of Agriculture, Forestry and Fisheries, Japan.

REFERENCES

[1] M. Buddhikot, G. Chandranmenon, S. Han, Y. W. Lee, S. Miller and L. Salgarelli, “Integration of 802.11 and third-generation wireless data networks,” Proceedings of the IEEE INFOCOM 2003, Vol. 1, pp. 503-512, 2003.

[2] Q. Zhang, C. Guo, Z. Guo and W. Zhu, “Efficient mobility management for vertical handoff between WWAN and WLAN,” IEEE Communications Magazine, Vol. 41, No. 11, pp. 102-108, 2003.

[3] L. A. Magagula and H. A. Chan, “IEEE802.21-Assisted Cross-Layer Design and PMIPv6 Mobility Management Framework for Next Generation Wireless Networks,” Proc. IEEE WIMOB ’08, pp. 159-164, Oct. 2008.

[4] D. Le, X. Fu and D. Hogrere, “A Review of Mobility Support Paradigms for the Internet,” IEEE Communications surveys, 1st quarter 2006, Volume 8, No. 1, 2006.

[5] A. C. Snoeren and H. Balakrishnan, “An End-to-End Approach to Host Mobility,” Proceedings of the ACM Mobicom, Page(s): 155-166, August 2000.

[6] C. Perkins, “IP Mobility Support for IPv4, Revised,” RFC 5944, IETF (2010).

[7] http://www.mip4.org, retrieved: February , 2012.

[8] S. Salsano, C. Mingardi, S. Niccolini, A. Polidoro and L. Veltri ”SIP-based Mobility Management in Next Generation Networks,” IEEE Wireless Communication, Vol. 15, Issue 2, April 2008.

[9] M. Bonola, S. Salsano and A. Polidoro, “UPMT: universal per-application mobility management using tunnels,” In Proc. of the 28th IEEE conference on Global telecommunications (GLOBECOM’09) 2009.

[10] H. Soliman “Mobile IPv6 Support for Dual Stack Hosts and Routers,” RFC 5555, IETF, 2009.

[11] H. Levkowetz and S. Vaarala, “Mobile IP Traversal of Network Address Translation (NAT) Devices,” RFC 3519, April 2003.

[12] Katsuhiro Naito, Kazuma Kamienoo, Takuya Nishio, Hidekazu Suzuki, Akira Watanabe, Kazuo Mori, Hideo Kobayashi, “Proposal of Seamless IP Mobility Schemes: Network Traversal with Mobility (NTMobile),” IEEE GLOBECOM 2012, December 2012.

[13] Katsuhiro Naito, Kazuma Kamienoo, Hidekazu Suzuki, Akira Watanabe, Kazuo Mori, and Hideo Kobayashi, “End-to-end IP mobility platform in application layer for iOS and Android OS,” IEEE Consumer Communications & Networking Conference (CCNC 2014), January 2014.

[14] http://www.ntmobile.net