Proposal of Seamless IP mobility schemes: Network traversal with Mobility (NTMobile)

Katsuhiro NAITO, Takuya NISHIO,
Kazuo MORI and Hideo KOBAYASHI
Graduate School of Engineering
Mie University
1577 Tsu, Mie, Japan
Email: naito, kmori, koba@elec.mie-u.ac.jp
nishio@com.elec.mie-u.ac.jp

Kazuma Kamienoo, Hidekazu SUZUKI
and Akira WATANABE
Graduate School of Science and Technology
Meijo University
Tenpaku-ku, Nagoya, Aichi, Japan
Email: hsuzuki@meijo-u.ac.jp, wtnbakr@meijo-u.ac.jp

***Abstract*—IP mobility is the technology that realizes a continuous communication even when communicating wireless nodes switch access networks. NTMobile (Network Traversal with Mobility) is a new type of IP mobility technology for the IPv4 and IPv6 networks. NTMobile nodes can also achieve IP mobility in global IP networks as well as in private IP networks by creating a tunnel route between a pair of NTMobile nodes. In NTMobile, network applications use virtual IP addresses to achieve a continuous communication at the time when real IP addresses change due to the switching of access networks or node movements. In NTMobile, we implement a packet manipulation method in the Linux kernel module so as to achieve high levels of throughput performance.**

I. Introduction

Various wireless communication technologies have been developed to achieve a higher data rate, wider coverage and improved mobility. Internet Protocol (IP) is used as the fundamental communication protocol in current wireless network services. In addition, users can select an optimal wireless network such as a 3G cellular system, IEEE 802.16e, IEEE 802.11 according to the connectivity of the services [1], [2]. The switching technology used between two different network services is called vertical handover, and is standardized as IEEE 802.21 [3].

Applications manage connections by using IP addresses in the Internet architecture. However, IP addresses assigned to nodes change to different addresses when the access networks change due to node movements or switching of access networks. Then, all connections are to be disconnected as the result of the change in IP addresses. Thus, IP mobility technologies have been developed to achieve a continuous communication even when IP addresses change [4], [5], [6], [7]. In almost all implementations conducted till now, recent IP mobility technologies have been developed for IPv6 networks because IPv6 is suitable for providing support for the mobility on heterogeneous networks. In the meantime, IPv4 is still being used in almost all networks, and therefore, these new technologies cannot be applied to the current networks. Furthermore, conventional technologies generally require a special device like a home agent for the relaying of data, which causes degradation of throughput and inefficient network usage due to detour routes through the home agent [8], [9], [10]. Additionally, a special agent called GFA (Gateway Foreign Agent) should be deployed in foreign networks even if regional registration mechanisms are employed to improve throughput and delay performance in Mobile IP v4 [11].

The number of IPv4 addresses is approaching its exhaustion in the IPv4 networks, NAT (Network Address Translation) is generally used to reduce the consumption of IP addresses. Generally, nodes in internal networks get access to the Internet through NAT routers. On the contrary, the nodes in the Internet cannot get access to the nodes in internal networks because the NAT router blocks packets from the Internet [12]. Mobile IP supports the mobility behind NAT routers [11], [12], [13]. However, performance degrades due to UDP capsuling and retour routes through a home agent. The authors have developed a technology called Mobile PPC (Mobile Peer to Peer Communication) in the past to achieve network mobility [14]. Mobile PPC is an IP mobility technology for IPv4 networks. Although IP mobility can be achieved by the pair of end nodes, Mobile PPC requires a global IP address assigned at either of the nodes. Then, it is difficult to employ this technology when either or both nodes use private IP addresses behind NAT routers.

In this paper, we propose a new type of network mobility technology, which we name NTMobile (Network Traversal with Mobility) [15]. NTMobile has been developed for both IPv4 and IPv6 networks. NTMobile node can achieve network mobility by connecting nodes directly to improve end-to-end throughput performance when one of nodes has a global IP address. NTMobile nodes use Relay Servers (RSs) to connect each other to achieve mutual connectivity between nodes behind NAT routers when both nodes exist behind NAT routers. Additionally, NTMobile network can employ additional RSs to decrease traffic load at each RS. Hence, NTMobile has a good scalability for large size of networks. Finally, NTMobile can achieve high throughput and short delay even if special agents in foreign networks are not required as in the case of Mobile IP. By using this technology, nodes can achieve a continuous connectivity in both present as well as future Internet environment. By conducting experiments, we confirmed that the overhead of packet manipulation is small and our proposed implementation can achieve a high throughput rate on the Linux system.

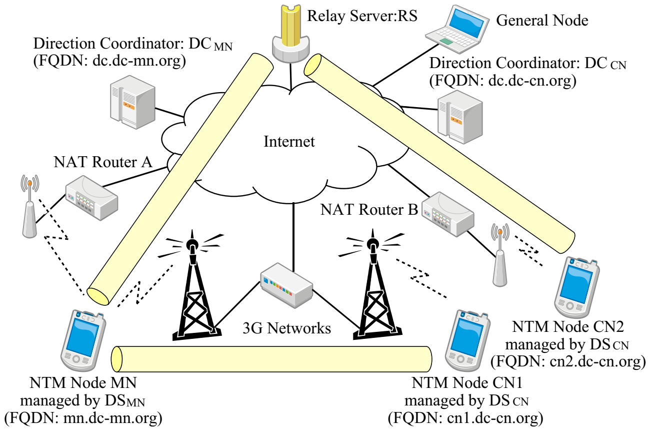

Fig. 1. Overview of NTMobile network.

II. NTMobile

Fig. 1 shows the overview of the NTMobile network. The NTMobile network uses three types of nodes; namely, NTMobile node, Direction Coordinator (DC), and Relay Server (RS). NTMobile node is a special type of end-user node that is equipped with our proposed method. NTMobile nodes have virtual IP addresses assigned to network applications in order to achieve a continuous communication. Then, the network applications use the virtual IP addresses to set up a connection and the IP datagram with virtual IP addresses is transmitted through a tunnel route by using real IP addresses. In this way, NTMobile nodes can communicate with each other even if both real addresses of NTMobile nodes change due to switching of networks or node movements, because the virtual IP addresses maintain their constant value. DC is a special type of server that manages virtual IP addresses for NTMobile nodes and controls the process between a pair of NTMobile nodes. Each network manager can install a DC into their own network to manage their NTMobile users. RS is another special type of server that relays encapsulated data packets between a pair of NTMobile nodes behind NAT routers, or relays data packets between an NTMobile node and a general node that does not support the NTMobile function. Since the proposed method employs virtual IP addresses for mutual communication, both IPv4 and IPv6 addresses can be accepted as real IP addresses to create tunnels between NTMobile nodes. Meanwhile, we describe mutual communication between IPv4 real addresses to simplify the explanation of the proposed method in this paper.

In Fig. 1, DC$_{\mathrm{MN}}$ manages NTMobile node MN and assigns a virtual IP address to it. DC$_{\mathrm{CN}}$ also manages nodes CN1 and CN2, and assigns virtual IP addresses to them. In this example, MN creates a tunnel towards CN1 directly because CN1 has a global IP address and can receive packets from MN directly, whereas MN creates a tunnel towards CN2 through RS because both nodes exist behind NAT routers and cannot communicate with each other directly. Furthermore, RS can create a tunnel towards MN and forward data packets to a general node. In this way, MN can achieve IP mobility by communicating with the general node through RS.

TABLE I
Assumed changing patterns of networks.

Pattern	Address type of NTMobile node	Address type of Correspondent node	Tunnel route after changing of addresses
1	G ⇒ G	G (NTMobile)	End-to-End
2	G ⇒ G	G (General)	via RS
3	G ⇒ P(A)	G (NTMobile)	End-to-End
4	G ⇒ P(A)	G (General)	via RS
5	P(A) ⇒ G	G (NTMobile)	End-to-End
6	P(A) ⇒ G	G (General)	via RS
7	P(A) ⇒ P(A)	G (NTMobile)	End-to-End
8	P(A) ⇒ P(A)	G (General)	via RS
9	P(A) ⇒ P(B)	G (NTMobile)	End-to-End
10	P(A) ⇒ P(B)	G (General)	via RS
11	G ⇒ G	P(A) (NTMobile)	End-to-End
12	G ⇒ P(A)	P(A) (NTMobile)	via RS
13	G ⇒ P(B)	P(A) (NTMobile)	via RS
14	P(B) ⇒ G	P(A) (NTMobile)	End-to-End
15	P(B) ⇒ P(A)	P(A) (NTMobile)	via RS or End-to-End
16	P(A) ⇒ P(B)	P(A) (NTMobile)	via RS
17	P(B) ⇒ P(C)	P(A) (NTMobile)	via RS

G:Global address, P(A):Private address in Network A

A. Node management

In the NTMobile network, some DCs are deployed on the Internet. Each DC has the function of extended DNS (Domain Name System). Therefore, DCs can manage host information as DNS A records and special NTMobile records called NTM records in DNS . DNS A records are used for general nodes to communicate with NTMobile nodes. The special NTMobile records are used for communication between NTMobile nodes. Therefore, the information about NTMobile network is not noticed by general nodes in the Internet.

In Fig. 1, DC$_{\mathrm{MN}}$'s FQDN (Fully Qualified Domain Name) is dc.dc-mn.org. Therefore, DC$_{\mathrm{MN}}$ can manage all the nodes such as mn.dc-mn.org in its domain. Additionally, the NTMobile network can support many DCs in different domains such ash dc-mn.org and dc-cn.org. In Fig. 1, DC$_{\mathrm{CN}}$ manages the nodes CN1 and CN2. Destined node information is searched by FQDN in the NTMobile network. Therefore, the node management in NTMobile is a scalable mechanism similar to DNS.

B. Changing patterns of networks

NTMobile can support the changing patterns of networks described in Table I to achieve IP mobility and mutual connectivity between a pair of NTMobile nodes, or between NTMobile node and a general node. Table I shows the tunnel route when real IP addresses of NTMobile nodes change due to the switching of networks or node movements. For example, a pair of NTMobile nodes communicate with each other directly in Pattern 1 because both nodes have global IP addresses and support the NTMobile function. In Pattern 2, NTMobile node and a general node communicate with each other through RS because general nodes do not support the NTMobile function, and RS communicates with a general node and forward data packets between them. In the case of Pattern 12, NTMobile nodes also communicate with each other through RS because both NTMobile nodes exist behind NAT routers and cannot make a direct communication.

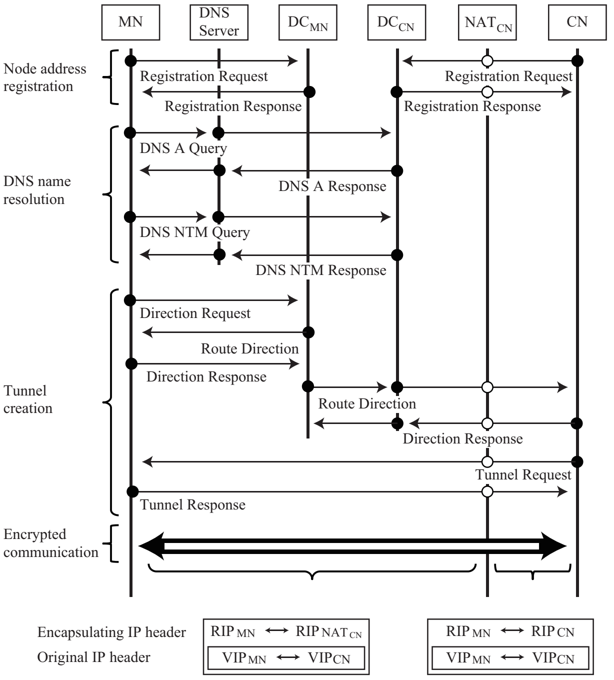

Fig. 2. Connection creating process between global IP and private IP.

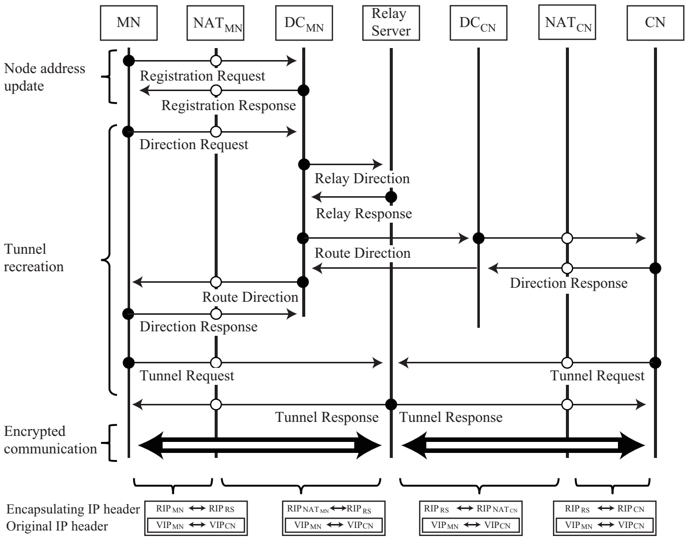

Fig. 3. Connection recreating process between private IP and private IP.

C. Connection creating procedures

Fig. 2 shows the connection creating procedures of the NTMobile method. In Fig. 2, NTMobile node MN tries to communicate with NTMobile node CN behind the NAT router. In our proposed method, the following procedures are followed to create a tunnel route between MN and CN. The NTMobile nodes share the tunnel route using UDP to exchange NTM messages and data packets. Therefore, all NTM messages are transfered through the same User Datagram Protocol (UDP) port. In the NTMobile network, we assume that connections between each NTMobile node and its DC are encrypted, and connections between all servers are encrypted as well.

Node address registration: Each NTMobile node registers its real IP address with its DC by transmitting a registration request message. Thus, MN registers its IP address with $\mathrm{DC_{MN}}$ and CN also registers its IP address with $\mathrm{DC_{CN}}$. In addition, each DC also registers NAT router's IP address when NTMobile node exists behind the NAT router, and DC replies a registration response message to MN. We have defined special types of DNS records to manage the registered information. In Fig. 2, $\mathrm{DC_{CN}}$ registers NAT router's IP address when it receives the registration request message from CN. In addition, each DC allocates a virtual IP address to NTMobile nodes to achieve mutual communication between them. Therefore, $\mathrm{DC_{MN}}$ allocates $\mathrm{VIP_{MN}}$ to MN, and $\mathrm{DC_{CN}}$ allocates $\mathrm{VIP_{CN}}$ to CN. After node address registration, NTMobile node transmits keep-alive messages to maintain the connection to its DC.

DNS name resolution: NTMobile methods employ DNS as a trigger to start connection establishing procedures because almost all network applications use FQDN to start a communication. In Fig. 2, the network application requires the resolution of FQDN, and DNS resolver transmits a DNS A query message to the primary DNS server. NTMobile method also transmits a DNS NTM query message to obtain the information registered in CN's registration [16].

Tunnel creation: In Fig. 2, MN is an initiator node and CN is an destination node. Therefore, MN transmits a direction request message to its $\mathrm{DC_{MN}}$. The direction request message consists of node ID, virtual IP address, correspondent node's ID, and real IP address, correspondent DC's address and correspondent's virtual IP address. These information are included in DNS NTM records.

$\mathrm{DC_{MN}}$ transmits the route direction message to MN. MN replies the direction response message to $\mathrm{DC_{MN}}$. $\mathrm{DC_{MN}}$ transmits the route direction message to CN through $\mathrm{DC_{CN}}$ after receiving the direction response message from MN. Then CN replies the direction response message to $\mathrm{DC_{MN}}$. The route direction message consists of node IDs, real IP addresses, external NAT router's IP addresses, virtual IP addresses, management DC's IP addresses of both nodes, and a path ID for tunnel identification. In this example, MN is the only node that has a global IP address. The tunnel request message, requesting to create a tunnel, is transmitted from CN to MN. MN replies a tunnel response message to create the tunnel route.

Encrypted communication: A pair of NTMobile nodes mutually transmit all data packets through the created tunnel to communicate with each other. Each NTMobile node shares a key with its DC by using SSL (Secure Socket Layer) authentication etc. In the data packet encapsulation procedures, all data packets are encrypted by NTMobile nodes with a temporal shared key which is encrypted by the above shared key and distributed from DC. In Fig. 2, $\mathrm{DC_{MN}}$ informs the encryption key with the route direction messages.

D. Tunnel recreating procedures after changing of IP addresses

NTMobile nodes need to recreate a tunnel route when an IP address has changed due to the switching of access networks or node movements. Fig. 3 shows the connection recreating procedures when NTMobile node MN switches from a global network to a private network. The difference between Figs. 2 and 3 is that there is no DNS name resolution procedure because NTMobile nodes after switching of networks already have information about correspondent nodes.

In our proposed method, the following steps are taken to recreate the tunnel route between MN and CN.

Node address update: The NTMobile node, that detected the change of its real IP address due to node movements or switching of networks, transmits a registration request message and informs its real IP address to its DC, and DC replies a registration response message to MN. In this way, node MN updates its IP address in $\mathrm{DC_{MN}}$. $\mathrm{DC_{MN}}$ also updates the NTM record according to the registration request message.

Tunnel recreation: NTMobile nodes transmit a direction request message to its DC to start tunnel creation procedures. The concept of tunnel recreation procedures of NTMobile is completely the same with the tunnel creation procedures shown in Fig. 2. DC transmits to both NTMmobile nodes a route direction message to create a new tunnel.

In Fig. 3, MN transmits a direction request message to $\mathrm{DC_{MN}}$. In this example, both nodes exist behind NAT routers, and thus, both nodes cannot communicate with each other directly. In the NTMobile method, RS connects two tunnels between RS and both NTMobile nodes, and forwards encrypted data packets through these two tunnels. $\mathrm{DC_{MN}}$ transmits a relay direction message to RS to accept the tunnel request messages from both MN and CN. The relay direction message contains path ID for tunnel identification. RS replies the direction response message to $\mathrm{DC_{MN}}$.

$\mathrm{DC_{MN}}$ transmits the route direction message to CN after receiving the direction response message from RS. CN replies the direction response message to $\mathrm{DC_{MN}}$. Then $\mathrm{DC_{MN}}$ transmits the route direction message to MN after receiving the direction response message from CN. MN replies the direction response message to $\mathrm{DC_{MN}}$.

$\mathrm{DC_{MN}}$ transmits the route direction message to MN after receiving the direction response message from RS. MN replies the direction response message to $\mathrm{DC_{MN}}$. Then $\mathrm{DC_{MN}}$ transmits the route direction message to CN after receiving the direction response message from MN. CN replies the direction response message to $\mathrm{DC_{MN}}$. MN and CN transmit tunnel request messages to RS to create a new tunnel when they receive a route direction message. RS replies tunnel response messages to create tunnels to both MN and CN.

Tunnel recreation causes temporal breaks in communication if nodes have only one communication interface. Most of the break time is due to the IP address acquisition of DHCP (Dynamic Host Configuration Protocol) sequence. If nodes have plural interfaces like 3G and WiFi, the break time can be reduced dramatically because communication and DHCP sequence can run in synchronization.

Encrypted communication: RS forwards all encrypted data packets between NTMobile nodes. In the forwarding process, source and destination addresses of capsulated header are modified. Encryption and decryption processes are not required at RS. In Fig. 3, RS simply forwards encrypted data packets between MN and CN. Therefore, the communication overhead between a pair of NTMobile nodes through RS can be reduced in the NTMobile method.

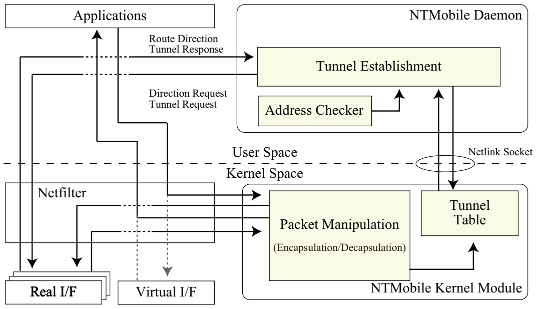

Fig. 4. Module configuration of NTMobile nodes.

III. Implementation and Evaluation

The targeted devices of NTMobile are Android OS terminals. Thus, we have implemented our proposed method on Linux OS. In the implementation of NTMobile, the packet manipulation method was implemented as Linux kernel module for Linux Netfilter. Therefore, only the installation of an implemented kernel module was sufficient in adding the NTMobile function to Linux, and modification of Linux kernel was not required. In addition, the kernel module implementation had a big advantage because it could reduce the overhead of encapsulation and encryption processes. If we mention our implementation more in detail, our code did not require memory copy of socket buffer to achieve encapsulation and encryption in the kernel. As the results, we could achieve a high throughput rate even if data packets were encapsulated or encrypted.

A. NTMobile nodes

Fig. 4 shows the module configuration of NTMobile nodes. The implementation of NTMobile nodes is categorized into two types; namely, one is user space implementation called NTMobile daemon and the other is kernel space implementation called NTMobile kernel. NTMobile daemon and NTMobile kernel are connected by the Linux Netlink socket.

1) User space implementation: NTMobile daemon has the address checking module and the tunnel creation module. The address checking module is to detect changes in IP addresses due to the node movements or the switching of networks. The tunnel creating module is to run the sequence described in Sec. II.

2) Kernel space implementation: NTMobile kernel has two functions. The first function is to encapsulate and decapsulate IP datagrams. In NTMobile, network applications use virtual IP addresses to communicate with NTMobile nodes. NTMobile kernel encapsulates IP datagrams including the virtual IP addresses to transfer them through created tunnels. NTMobile kernel also decapsulates received IP datagrams and forwards the decapsulated IP datagrams including the virtual IP addresses to network applications.

The second function is to encrypt and decrypt IP datagrams. Both of NTMobile nodes receive an encryption key from DC. In this way, NTMobile also supports encrypted communications. In these packet manipulations, memory copy of the socket buffer is not required. Therefore,our proposed implementation can achieve high throughput performance.

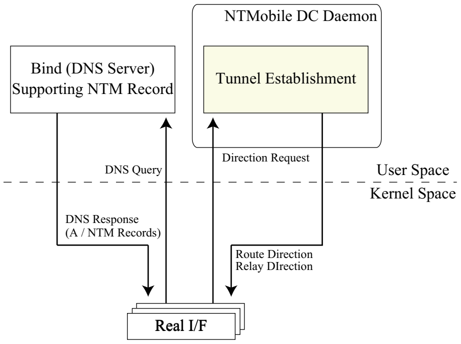

Fig. 5. Module configuration of Direction Coordinators.

B. Direction Coordinators

Fig. 5 shows the module configuration of DC. The main functions of DC are to manage NTMobile nodes information as special DNS records called NTM records, and to control NTMobile nodes to create tunnels. These functions are implemented by NTMobile DC daemon in the user space.

The first function is to manage locational information of NTMobile nodes. The procedures to manage locational information are described in the node address registration in Fig. 2 and in the node address update in Fig. 3. DC implements special DNS server supporting NTM records. They register or update NTM record information when they receive a registration request message. In addition, DC replies locational information of NTMobile nodes by using DNS NTM responses when they receive a DNS NTM Query message. Accordingly, the location management of NTMobile has a similar scalability to DNS.

The second function is to control procedures to create a tunnel between a pair of NTMobile nodes. The procedures are described in the tunnel creation in Fig. 2 and in the tunnel recreation in Fig. 3.

The third function is to manage RS to forward data packets between a pair of NTMobile nodes. The procedures are described in the tunnel recreation in Fig. 3. In relay communications, DC informs path ID for tunnel identification to RS.

C. Relay Servers

Fig. 6 shows the module configuration of RSs. The function of RS is to forward encapsulated IP datagrams between NTMobile nodes. In NTMobile, many NTMobile nodes communicate with their correspondent nodes through RS. Thus, scalability of RS is important. In NTMobile network, some RSs can be installed to achieve load balance. Therefore, DC selects an adequate RS for each NTMobile nodes.

In the relay operations, RS only modifies source and destination IP addresses of the encapsulated IP datagram header described in Fig. 3. Therefore, encryption and decryption processes are performed solely at NTMobile nodes. In addition, the packet manipulation method is implemented on the Linux kernel module. As a result, our proposed implementation scheme can handle data packets in the Linux kernel module, and achieve low overhead of packet manipulation.

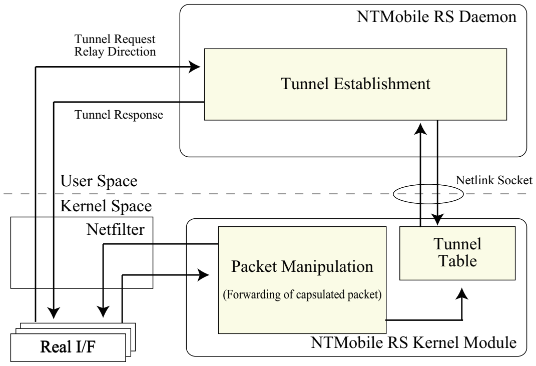

Fig. 6. Module configuration of Relay Servers.

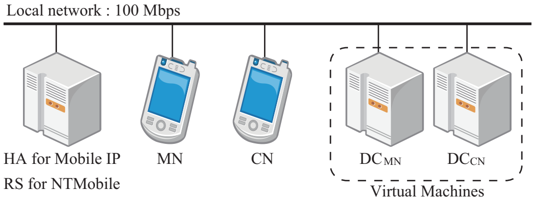

Fig. 7. Performance evaluation model.

There are two types of implementation of Relay Servers; namely, one is user space implementation called NTMobile RS daemon and the other is kernel space implementation called NTMobile RS kernel. NTMobile RS daemon and NTMobile RS kernel are connected by a Linux Netlink socket.

1) User space implementation: The NTMobile RS daemon has the tunnel creating method for NTMobile nodes. It registers or updates the tunnel table when it receives relay direction messages from DC.

2) Kernel space implementation: NTMobile RS kernel module has the forwarding function for NTMobile nodes and general nodes. In the forwarding operation between a pair of NTMobile nodes, it changes only source and destination addresses of the outer IP datagram header, whereas in forwarding operation between NTMobile nodes and general nodes, it decapsulates or encapsulates the IP datagrams through tunnels, and communicates with general nodes instead of NTMobile nodes.

D. Performance evaluation

NTMobile employs a capsulation process of IP datagrams to achieve IP mobility. Generally, it is known that the overhead of capsulation process causes deterioration in the throughput performance. In this paper, we evaluated the overhead of capsulation in the implemented kernel module comparing to Linux default kernel performance and Mobile IP performance. Fig. 7 shows the evaluation model.

For the purpose of our evaluation, we implemented our proposed method on the Ubuntu Linux 10.04. The kernel version is linux-2.6.32-24-generic. Both end nodes, the home agents for Mobile IP and the relay server for NTMobile have Intel Core i7 870 2.93GHz CPU, 10 GBytes physical memory. Both direction coordinators are operated as the virtual machines on Ubuntu server 10.04. Iperf [17] is used to evaluate the

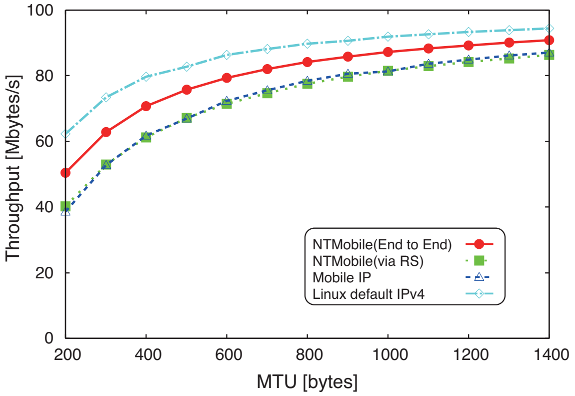

Fig. 8. Throughput performance without encryption.

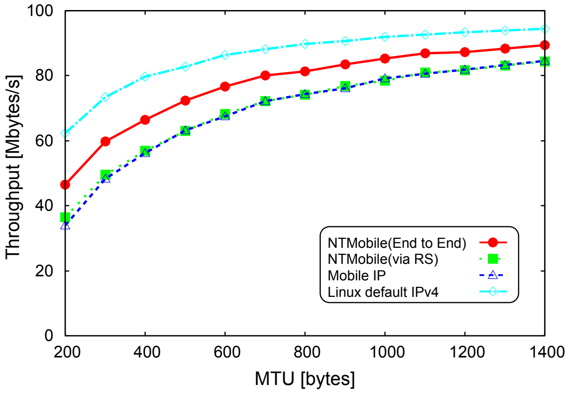

Fig. 9. Throughput performance with encryption.

throughput performance between both end nodes. The results are based on the average of evaluations that we conducted 10 times.

Fig. 8 shows the throughput performance without encryption versus MTU (Maximum Transfer Unit). From the results, we can find that the performance of direct communication in NTMobile has better performance comparing to Mobile IP. Additionally, the throughput performance of relay communication in NTMobile is almost same as the throughput performance of Mobile IP. Comparing to the throughput performance of Linux default kernel, we can find that the overhead of our proposed method decreases in the increasing of MTU size. In NTMobile method, 56 Bytes including NTM header and encapsulated IP datagram header are added to original IP datagrams. Therefore, the effect of adding the special NTM data increases in small size of MTU. In the meantime, we can find that the throughput deterioration of NTM method is almost the same as the deterioration due the addition of the NTM header and IP header for encapsulation. Therefore, the overhead of packet capsulation is especially quite small.

Fig. 9 shows the throughput performance with encryption versus MTU (Maximum Transfer Unit). In the encryption mode, additional bytes between 72 and 87 Bytes are added as NTM header, encapsulated IP datagram header, hash-based message authentication code, and padding. From the results, we can find that the degradation of throughput performance due to encryption is small. Finally, we confirmed that the implemented kernel module can achieve high throughput performance. Therefore, NTMobile can achieve IP mobility with with high levels of throughput.

IV. Conclusion

In this paper, we have proposed a new type of network mobility technology which we named NTMobile (Network Traversal with Mobility). NTMobile has been developed for the IPv4/IPv6 coexisting environment. The unique features of NTMobile are to employ virtual IP addresses to achieve continuous connectivity of network applications and to create a direct tunnel route between a pair of NTMobile nodes or through RSs to achieve IP mobility in the current Internet environment. By conducting experiments, we confirmed that the overhead of packet manipulation is small and our proposed system can achieve a high throughput rate on the Linux system.

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

[6] M. Ishiyama, M. Kunishi, K. Uehara, H. Esaki, and F. Teraoka, “LINA: A New Approach to Mobility Support in Wide Area Networks,” IEICE Trans. Comm., Vol.E84-B, No.8, pp.2076– 2086, 2001.

[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] M. Bonola and S. Salsano, “S-UPMT: a secure Vertical Handover solution based on IP in UDP tunneling and IPsec,” GTTI Riunione Annuale 2010, (online), http://www.gtti.it/GTTI10/papers/gtti10_submission_29.pdf, 2010.

[11] E. Fogelstroem, A. Jonsson, and C. Perkins, “Mobile IPv4 Regional Registration,” RFC 4857, June 2007.

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

[13] G. Montenegro, “Reverse Tunneling for Mobile IP, revised,” RFC 3024, January 2001.

[14] H. Suzuki, K. Terazawa and A. Watanabe, “Implementation of NAT Traversal for Mobile PPC with the Principle of Hole Punching,” in Proc. of the IEEE International Region 10 Conference 2009 (TENCON2009), Nov. 2009.

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

[16] A. Gustafsson, “Handling of Unknown DNS Resource Record (RR) Types,” RFC 3597, September 2003.

[17] http://iperf.sourceforge.net