Implementation of NAT Traversal for Mobile PPC with the Principle of Hole Punching

Hidekazu Suzuki
Research Fellow of the Japan Society for the Promotion of Science,
Graduate School of Science and Technology
Meijo University, Nagoya 468-8502, JAPAN
Email: h.suzuki@wata-lab.meijo-u.ac.jp

Keiji Terazawa and Akira Watanabe
Graduate School of Science and Technology
Meijo University, Nagoya 468-8502, JAPAN
Telephone and Fax: +81–52–838–2406
Email: k.terazawa@wata-lab.meijo-u.ac.jp
wtnbakr@ccmfs.meijo-u.ac.jp

***Abstract*—In the IPv4 network, a mobile node is expected to move between a global address network and a private address network during communication. We have already proposed a mobility mechanism that applies the principle of hole-punching, widely known as "NAT traversal technology", when a NAT exists on a communication path. It can realize mobility over different types of address areas. In this paper, we describe implementation of the above-mentioned function to Mobile Peer-to-Peer Communication (Mobile PPC) that can realize mobility solely with end nodes.**

I. Introduction

With the growth of mobile terminals and wireless networks, users are now able to get access to networks from anywhere. While the use of cellular phone networks is overwhelmingly popular at present, there exist recently mobile terminals equipped with wireless LAN or WiMAX as well, and it is expected that there will be more and more users who can have connection with IP networks seamlessly. In the case of IP network, however, communications are disconnected if and when users move during communication or change wireless communication devices, because IP addresses change at that occasion. In order to solve this problem, various kinds of mobility technologies have been studied [1].

Many of the technologies to realize mobility are based on the assumption of using IPv6. However, since IPv6 has no compatibility with IPv4 in the aspect of interconnection, it is assumed that a situation where IPv4 and IPv6 coexist should last for a fairly long period of time. Accordingly, if the device of the correspondent node (CN) is able to work merely with IPv4, it is necessary to communicate with IPv4. Thus, realization of a mobility technology in the IPv4 network is considered to be quite meaningful.

In the case of IPv4 network, it is difficult to assign global IP addresses to all mobile nodes (MNs) owing to the problem of IPv4 address exhaustion, and therefore, it is necessary to actively utilize private IP addresses. For that reason, MNs may move between a network where global IP address are used (hereinafter called "global network") and a network where private IP address are used (hereinafter called "private network"). In this kind of move, there needs to exist a Network Address Translator (NAT) on the path of communication before or after the move. As a result, the new IP address reported by the move notification does not match with the IP address used in the actual communication. Therefore, the relationship between the IP addresses before and after the move cannot be maintained correctly.

In Mobile IP [2], various countermeasures [3]–[5] have been studied so as to solve this problem, encapsulating the move notification with UDP or adding a special function to NAT. However, these methods cannot solve the problem in an efficient manner, as the transmission efficiency tends to degrade due to the header-overhead caused by the encapsulation or a special type of NAT is required.

We have proposed Mobile Peer-to-Peer Communication (Mobile PPC) [6] that can realize mobility in the IPv4 network with only end nodes. In the case of Mobile PPC, an MN starts communication with a CN after resolving its IP address by way of Dynamic DNS (DDNS) [7]. At the time when the IP address of MN changes during communication, MN directly notifies CN of the relationship of IP addresses before and after the move, and both nodes save it in their IP layers. Thereafter, they translate the IP addresses of all TCP and UDP packets in the IP layer, and thus, IP address changes are concealed from the upper layer and communications can be continued.

We have already proposed a NAT traversal method for Mobile PPC, whereby existing NATs can be used without any modification and yet MN can move freely between a global network and a private network [8]. In our proposed method, the principle of hole punching, which is known as a representative NAT traversal technology [9] is introduced as a means to cope with the address translation in NAT.

MN located in a private network makes NAT generate mapping information by binding negotiation (which is equivalent to the hole punching) to CN, and then obtains the IP address and the port number mapped to the exterior side of the NAT. Thereafter, by sending the acquired information in the move notification to CN, CN can appropriately translate the IP address and the port number of received packets from MN.

In this paper, we describe the fundamental mechanism of Mobile PPC and our proposed method in Section II and the implementation of our proposed method in Section III. Then Section IV describes the results of the demonstration of

978–1–4244–4547–9/09/$26.00 ©

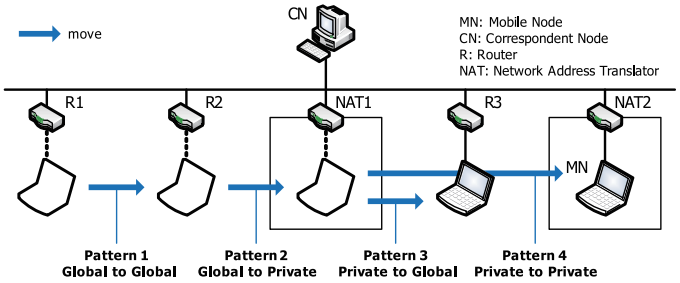


Fig. 1. Mobility Patterns in IPv4 Environment

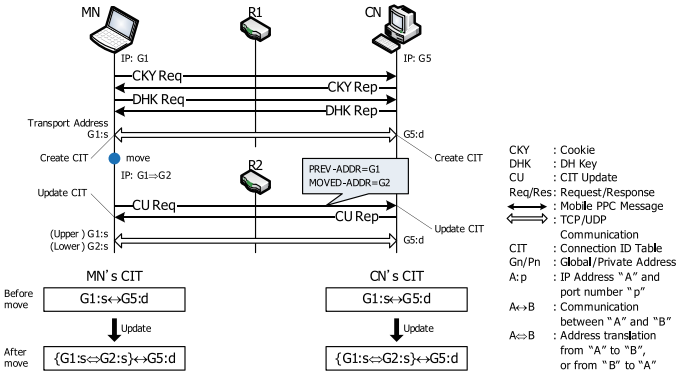


Fig. 2. Basic Sequence of Mobile PPC and Created CITs

the trial system together with some remaining problems, and finally Section V summarizes this paper.

II. Mechanism of Our Proposed Method

A. Definition of Mobility Patterns and Notation

Mobility patters of MN in the IPv4 network are shown in Fig. 1. In this paper, it is assumed that CN exists in a global network and that MN can start communication without fail. There exist the following four patterns.

- Pattern 1: Move from a global network to another global network.
- Pattern 2: Move from a global network to a private network.
- Pattern 3: Move from a private network to a global network.
- Pattern 4: Move from a private network to another private network.

While the conventional Mobile PPC corresponds to Pattern 1 only, our proposed method offers function to correspond to Pattern 2 to Pattern 4. Symbols used in this paper are defined as follows:

- Gi; Global IP address.
- Pi; Private IP address.
- $A : p$; IP address A and port number p.
- $proto$; Protocol type. (TCP or UDP)
- $S \rightarrow D$, $D \leftarrow S$; Communication from S to D.
- $S \leftrightarrow D$; Communication between S and D.
- $S \Leftrightarrow D$; Address translation from S to D, or from D to S.

B. Mobile PPC

Fig. 2 shows a sequence from the start of communication to the continuation of communication after the move in Mobile PPC. The IP addresses of MN and CN are indicated as "$G1$" and "$G5$". MN shares an authentication key with CN prior to the start of communication. This negotiation consists of two round trips of Cookie exchange and Diffie-Hellman (DH) key exchange. The authentication key is used for the verification at the time of move notification process.

After the above key sharing negotiation, MN creates a Connection ID Table (CIT) as shown in (1) by using the Connection ID (CID)[1] of the first TCP/UDP packet and then starts communication.

$$MN/CN: \quad G1 : s \leftrightarrow G5 : d \quad [proto] \tag{1}$$

At this stage, no address translation of the packet is yet made. CN also creates the same CIT just like MN when it receives the first TCP/UDP packet and memorizes the session information.

When MN moves to another network during communication and obtains a new IP address "$G2$", it executes a move notification process with CN. In the CIT Update (CU) request message to be sent from MN to CN, the IP address before move (PREV-ADDR) and the IP address after move (MOVED-ADDR) are described and a digital signature using the authentication key shared at the time of starting communication is added.

CN, upon receipt of CU request message, completes its verification and updates its own CIT that shows the IP address of MN becomes "$G2$" as shown in (2), and sends CU response message to MN.

$$MN/CN: \quad \{G1 : s \Leftrightarrow G2 : s\} \leftrightarrow G5 : d \quad [proto] \tag{2}$$

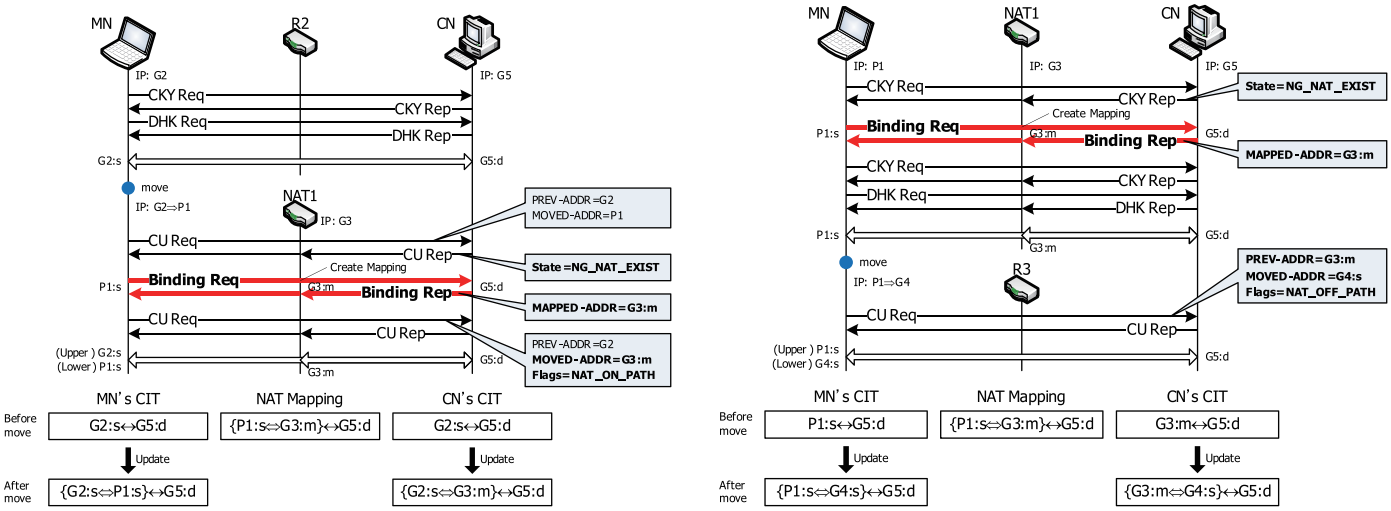
MN, upon receipt of CU response message, updates CIT in the same way as CN and completes its move notification process.

Thereafter, address translation is executed according to the updated CIT in the IP layer of both end nodes as follows. MN translates the source IP address of the packet passed from the upper layer from "$G1$" before the move of MN to "$G2$" after the move of MN, and sends it to CN. CN reversely translates the source IP address of the received packet from "$G2$" to "$G1$", and passes it to the upper layer. By performing the above sequence, the change in the IP address is concealed from the upper layer and the routing is performed correctly and thus, communication can be continued.

C. NAT Traversal Method

If a NAT exists on the communication path between MN and CN (e.g., MN locates in a private network and CN locates in a global network), the private IP address recognized as its own address by MN does not match the IP address recognized

[1]CID is information to identify a TCP connection or a UDP stream and is composed of five elements; namely, source and destination IP addresses/port numbers and protocol type.



(a) Case of Mobility Pattern 2

(b) Case of Mobility Pattern 3

Fig. 3. Communication Sequence

by CN as MN's IP address (namely, NAT's global IP address). As a result, CIT cannot be correctly updated with the move notification by ordinary Mobile PPC and thus, communication cannot be continued.

In order to solve the above problem, what is required of MN is to get the transport address of MN recognized by CN, namely the global IP address and port number mapped by NAT. Accordingly, we apply here the principle of hole punching to Mobile PPC. Hole punching is widely known as a NAT traversal technology. In Mobile PPC, a binding message is defined anew and the function is realized.

1) Communication Sequence (Mobility Pattern 2): Fig. 3a shows the communication sequence in the case of the move of MN from a global network to a private network. As no NAT exists on the communication path at the time of starting communication, the same process as that of ordinary Mobile PPC is performed.

When MN moves to the private network and obtains a new private IP address "$P1$", it executes a move notification process with CN. In the CU request message to be sent to CN, PREV-ADDR "$G1$" and MOVED-ADDR "$P1$" are written. When CN detects inconsistency between the MOVED-ADDR notified by CU request message and the source IP address of the message (i.e., NAT1's global IP address "$G3$") , it judges that there exists a NAT on the communication path and sends back to MN a CU response message with a status flag "NG_NAT_EXIST" without updating CIT.

MN, upon receipt of CU response message with the above flag, sends a binding request message to CN. The binding request message is composed of the same IP header and the transport header as those of the TCP/UDP packet to be actually sent by MN after its move and a Mobile PPC header[2]. In the case of Fig. 3a, the source address of the binding request message is "$P1 : s$" and the destination address is "$G5 : d$".

When NAT forwards the binding request message , the source address of the packet is translated from "$P1 : s$" to "$G3 : m$", and mapping information is created as shown in (3).

$$NAT1: \quad \{P1 : s \Leftrightarrow G3 : m\} \leftrightarrow G5 : d \quad [proto] \qquad (3)$$

CN obtains the mapped address "$G3 : m$" from the IP and the transport headers of the received binding request message and sends back a binding response message by describing them as "MAPPED-ADDR". MN modifies MAPPED-ADDR to MOVED-ADDR, and executes the move notification process again. In the CU request/response messages at this time, a status flag "NAT_ON_PATH" is set. Through this procedure, MN and CN can correctly update CIT, taking the address translation of NAT into account.

$$MN: \quad \{G1 : s \Leftrightarrow P1 : s\} \leftrightarrow G5 : d \quad [proto] \qquad (4)$$

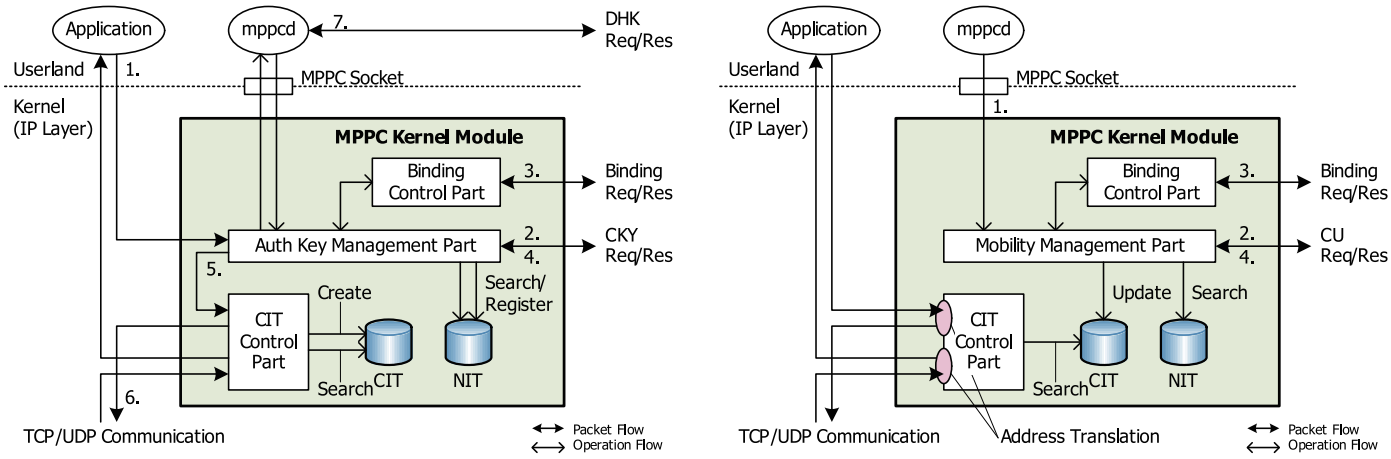
$$CN: \quad \{G1 : s \Leftrightarrow G3 : m\} \leftrightarrow G5 : d \quad [proto] \qquad (5)$$

2) Communication Sequence (Mobility Pattern 3): Fig. 3b shows the communication sequence for the case that MN moves from a private network to a global network. In this case, because a NAT exists on the communication path at the time of starting communication, a binding process is performed during the authentication key sharing process by the same mechanism as the move notification process of Mobility Pattern 2.

In NAT1, at the time of forwarding a binding request message, the same mapping information as (3) is created. By the creation of this information, MN can know the transport address "$G3 : m$" of MN recognized by CN, namely MAPPED-ADDR, in advance.

When MN moves to a global network and obtains a new global IP address "$G4$", it performs a move notification process with CN. Here, the PREV-ADDR to be described in the CU

[2]In the case of TCP communication, it is actually a TCP header, and in the case of UDP communication, it is a UDP header. The Mobile PPC header is described to judge whether it is a binding message or not. As for the details, please refer to Subsection III-B.



(a) At the time when MN starts communication

(b) After the move of MN

Fig. 4. Diagram Indicating the Relationship between Modules and Communication Sequence

request message is made as MAPPED-ADDR "$G3 : m$", and a status flag "NAT_OFF_PATH" is set. Through this process, MN and CN can correctly update CIT, taking the fact into account that the address translation of NAT is no more in place.

$$MN: \quad \{P1 : s \Leftrightarrow G4 : s\} \leftrightarrow G5 : d \quad [proto] \tag{6}$$

$$CN: \quad \{G3 : m \Leftrightarrow G4 : s\} \leftrightarrow G5 : d \quad [proto] \tag{7}$$

3) Communication Sequence (Mobility Pattern 4): In the case where MN moves from a private network to a different private network, NATs should exist on both communication paths before and after the move. In this case, the process before move shown in Fig. 3b and the process after move shown in Fig. 3a are to be combined. With this combination, two MAPPED-ADDRs obtained by binding processes conducted before move and after move can be set as PREV-ADDR and MAPPED-ADDR respectively.

III. Implementation

A. Module Composition and Process Flow

Fig. 4 shows diagrams indicating the module composition of Mobile PPC and its corresponding relationship with the communication sequence before move (Fig. 4a) and after move (Fig. 4b). Mobile PPC is composed of a kernel module to be implemented in the IP layer and a daemon "`mppcd`" that works in the userland.

The kernel module consists of several components; a CIT control part, an authentication-key management part, a mobility management part , and a binding control part which was newly implemented this time. The binding control part takes care of the creation of binding messages and the handling of sending and receiving messages, and passes obtained MAPPED-ADDR to the authentication-key management part as well as to the mobility management part.

When an application starts communication, TCP/UDP packets are firstly passed to the authentication-key management part. Here, Node Information Table (NIT) that manages the authentication key is checked and the authentication key of the correspondent node is searched. If no authentication key exists, the first TCP/UDP packet is temporarily stored in the kernel memory, and it calls the function to start an authentication key sharing negotiation. Here, if a status flag "NG_NAT_EXIST" is set in the received cookie response message, then the operation of the binding control part is executed.

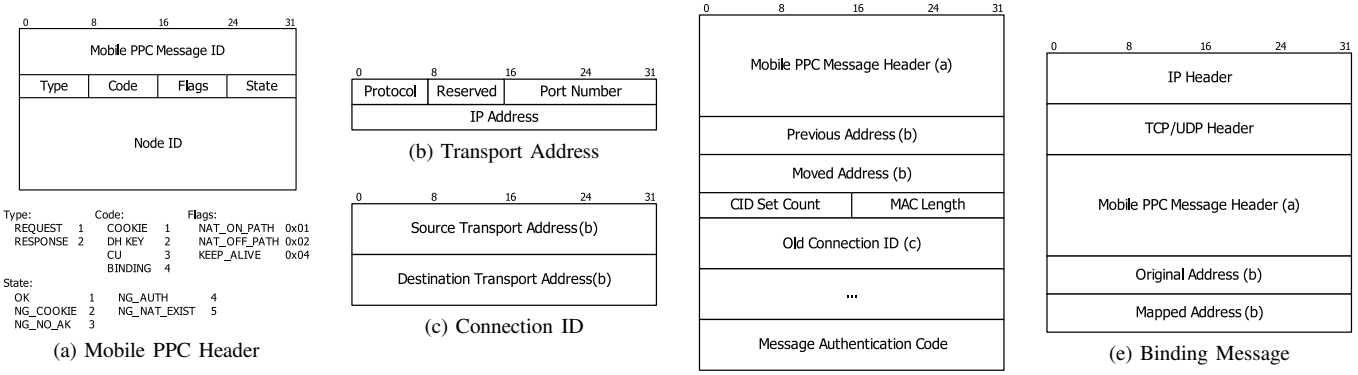
After completing the first round of the authentication key sharing negotiation, the stored TCP/UDP packet is restored, and after the creation of CIT, communication is to start. Thereafter, the authentication-key management part makes `mppcd` daemon execute the second round of DH key exchange through an MPPC socket. With this mechanism, it becomes possible to have the DH key exchange and the authentication-key generation process be executed in the back-end of actual TCP/UDP communication and to reduce the delay at the time of starting communication.

In the meantime, if `mppcd` daemon detects any move, it instructs the mobility management part to perform move notification process. In the same way as that by the authentication-key management part, the mobility management part executes the operation by the binding control part, if the status flag "NG_NAT_EXIST" is set in the received CU response message. After the updating of CIT, address/port translation of TCP/UDP packets is performed in the CIT management part and passed to either the upper layer or the lower layer.

B. Message Format

Fig. 5 shows the message format of Mobile PPC. Mobile PPC control message is based on ICMP Echo/Echo Reply and it is so structured that Mobile PPC header (Fig. 5a) follows ICMP header.

In our proposed method, 16 Octets field called Node ID is added anew. Node ID is an ID to assign a single meaning to each node, and even if a complicated IP address change occurs, end nodes can easily identify from which node each message was sent out. This value is created from Fully Qualified Domain Name (FQDN) of each node by a hash function by



(a) Mobile PPC Header

(b) Transport Address

(c) Connection ID

(d) CU Message

(e) Binding Message

Fig. 5. Mobile PPC Message Formats

using Universally Unique Identifier (UUID) [10].

In Mobile PPC, the initial IP address of the corresponding node is resolved by using DDNS server. Therefore, it is not necessary to make a new setting to the node when creating Node ID, because all the nodes using Mobile PPC necessarily possess FQDN.

In addition, we expanded the fields of PREV-ADDR and MOVED-ADDR in the CU messages. Though it was required to notify only IP addresses before and after move in the case of conventional Mobile PPC, it is also necessary to notify the port numbers in addition to IP addresses in the proposed method. Accordingly, we defined anew a format for the transport address (Fig. 5b). This format shows a set of data indicating a transport address (pair of an IP address and a port number) which various nodes use for communication, which is used for Connection ID (Fig. 5c), in addition to PREV-ADDR and MOVED-ADDR (Fig. 5d).

With respect to the binding message, Mobile PPC header follows TCP/UDP header as indicated in Fig. 5e. The source transport address of a binding request message in MN is described in the original address field, and MAPPED-ADDR mapped in NAT in the Mapped Address field.

C. Definition of Binding Message List

In introducing the binding process, Binding Message List (BML) is defined anew. Fig. 6 shows the structure of BML. The main objectives of BML are to reposit binding message information and also to store a triggered TCP/UDP message. Upon the creation of BML, monitoring of binding messages is initiated and upon the deletion of BML, monitoring is terminated. BML has a connection list structure and is created for each correspondent node, and furthermore, it has a sub-connection list structure for each established session.

For instance, when MN has established two sessions with the same CN, only one CU request message is required and carries the transport addresses of the two sessions (see Fig. 5d). However, although CU message has all information, the binding process cannot be integrated as it needs to be performed for each session.

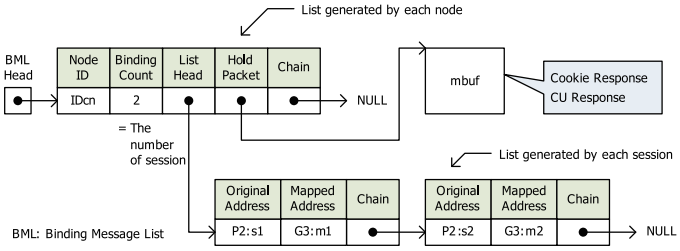


Fig. 6. Structure of Binding Message List

Thus, each node establishes the number of session information described in CU message as a binding count value and decrease the value each time it receives a binding message from the correspondent node. When the value has become zero, CN deletes BML and stops the monitoring of binding messages. On the side of MN, it records the two transport addresses described in the biding response message in the sub-connection list of BML. If the binding count value has not yet reached zero, MN should be able to receive other binding response messages and therefore, monitoring is continued. When the binding count has become zero, MN deletes BML after passing recorded transport addresses to the authentication-key management part or the mobility management part.

IV. Results and Considerations

A. Results of Demonstration of Trial System

This time, we have implemented the binding control part and improved the mobility management part of the previously-implemented Mobile PPC module in FreeBSD 7.0-RELEASE. We have checked the operation of the system when MN moves to a private network after MN and CN started communication in the global network using UDP. As a result, we have confirmed that the system works properly and the communication continues even if MN moves from a global network to a private network.

B. Remaining Problems

1) Move from Private Network to Global Network: While we are to obtain MAPPED-ADDR by binding process at the

time of authentication key sharing negotiation, it is needed to hold this transport address until a move notification process is performed. Thus, the MAPPED-ADDRESS obtained before move is stored in NIT entry. Because it is necessary for MN to obtain multiple MAPPED-ADDRESS with multiple CNs, it is required to expand the capacity of NIT by adding a sub-connection listing structure like BML.

2) Obstruction of Firewalls: When studying the issue of NAT traversal, we cannot ignore the existence of firewalls. As Stateful Packet Inspection (SPI) function is implemented in many of the NAT routers in recent years, firewalls check the consistency between outbound packets and inbound packets. If the TCP/UDP sessions are not based on correct procedures, their packets are discarded. In TCP communication in particular, it is thought to be necessary to make the binding sequence a 1.5 round trip sequence like a TCP three-way handshake.

3) Keep-Alive Operation: The NAT mapping information created by binding process is deleted after a certain limited period of time has elapsed without any communication. While in the case of TCP sessions the created mapping information is maintained for a fairly long time once connection is established, the mapping information is maintained only for a relatively short time in the case of UDP.[3] Therefore, it is necessary for MN to perform Keep-Alive operation for CN periodically when communicating with UDP.

V. Conclusion

In this paper, we have shown the method of applying the principle of hole punching to binding process of Mobile PPC and then shown the result of implementation of part of the function. We have confirmed that MN can continue communication even if it moves from a global network to a private network.

Hereafter, we will study the remaining problems, and evaluate the effects to the time period of communication interruption and the load due to Keep-Alive operation. We have already proposed another NAT traversal method in [13] that can realize both NAT traversal and mobility in the case where an MN starts communication from the external of NAT to a CN in a private network. However, since some functions are added to NAT in that method, it cannot cope with the situation where MN moves to an unspecified private network as shown in this paper. We also plan to study a method which enables receipt of communication from the external of NAT by effectively utilizing the binding function.

Acknowledgment

This research was partially supported by Japan Society for the Promotion of Science (JSPS), Grant-in-Aid for JSPS Fellows, 20-1069, 2008.

[3]For instance, for FreeBSD NAT, while mapping information is maintained for 24 hours in the case of TCP, it is deleted after 60 seconds in the case of UDP [11]. For CISCO NAT, while mapping information is also maintained for 24 hours in the case of TCP, it is deleted after five minutes in the case of UDP [12].

References

[1] F. Teraoka, "Node mobility protocols in the Internet," *Electronics and Communications in Japan (Part III: Fundamental Electronic Science)*, vol. 88, no. 6, pp. 39–59, Jun. 2005.
[2] C. Perkins, "IP mobility support for IPv4," RFC 3220, Jan. 2002.
[3] H. Levkowetz and S. Vaarala, "Mobile IP traversal of network address translation (NAT) devices," RFC 3519, Apr. 2003.
[4] G. Montenegro, "Reverse tunneling for mobile IP, revised," RFC 3024, Jan. 2001.
[5] A. Idoue, H. Yokota, and T. Kato, "Proposal of hierarchical mobile IP supporting private addresses utilizing NAT function and its implementation on UNIX operating system," *IEICE Transactions on Communications*, vol. E84-B, no. 12, pp. 3155–3165, Dec. 2001.
[6] M. Takeuchi, H. Suzuki, and A. Watanabe, "A proposal of Mobile PPC that realizes end-to-end mobility and its implementations," *Transactions of Information Processing Society of Japan*, vol. 47, no. 12, pp. 3244–3257, Dec. 2006, (in Japanese).
[7] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, "Dynamic updates in the domain name system (DNS UPDATE)," RFC 2136, Apr. 1997.
[8] H. Suzuki and A. Watanabe, "Design of NAT traversal for Mobile PPC applying hole punching technology," in *Proc. IEEE Region 10 Conference (TENCON'08)*, Hyderabad, India, Nov. 2008, pp. 1–6.
[9] B. Ford, P. Srisuresh, and D. Kegel, "Peer-to-peer communication across network address translators," in *Proc. USENIX Annual Technical Conference*, Anaheim, CA, Apr. 2005, pp. 179–192.
[10] P. Leach, M. Mealling, and R. Salz, "A universally unique identifier (UUID) URN namespace," RFC 4122, Jul. 2005.
[11] (2007) FreeBSD/Linux kernel cross reference. [Online]. Available: http://fxr.watson.org/fxr/source/netinet/libalias/alias_db.c?v=FREEBSD70#L213
[12] (2003) Rate limiting NAT translation. [Online]. Available: http://www.cisco.com/en/US/docs/ios/12_3t/12_3t4/feature/guide/gt_natrl.html#wp1027269
[13] H. Suzuki and A. Watanabe, "A realization method of mobility in case that a correspondent node is in a private network," *IEICE Transactions on Communications (Japanese Edition)*, vol. J92-B, no. 1, pp. 109–121, Jan. 2009, (in Japanese).

Singapore, Nov. 23-26, 2009
Session: Wired and Wireless Networks
TUE3.4.6 P0819

Implementation of NAT Traversal for Mobile PPC with the Principle of Hole Punching

Hidekazu SUZUKI†‡ **Keiji TERAZAWA**† **Akira WATANABE**†

† Meijo University, JAPAN

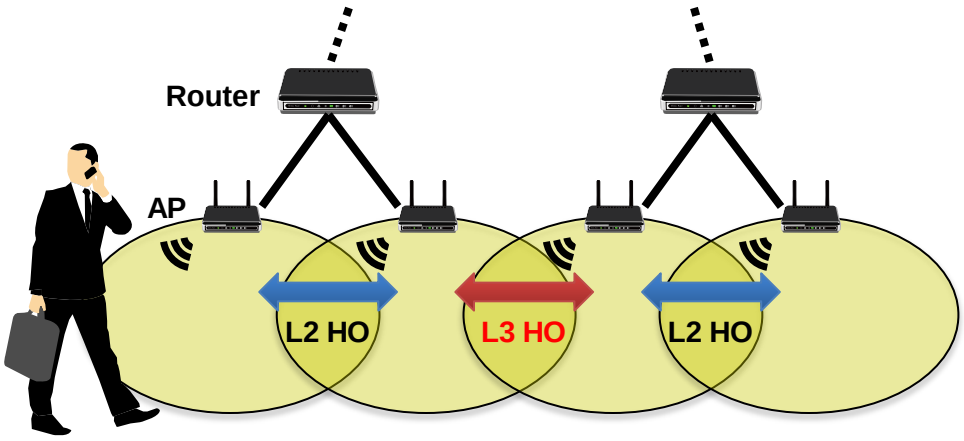
‡ Research Fellow of the Japan Society for the Promotion of Science

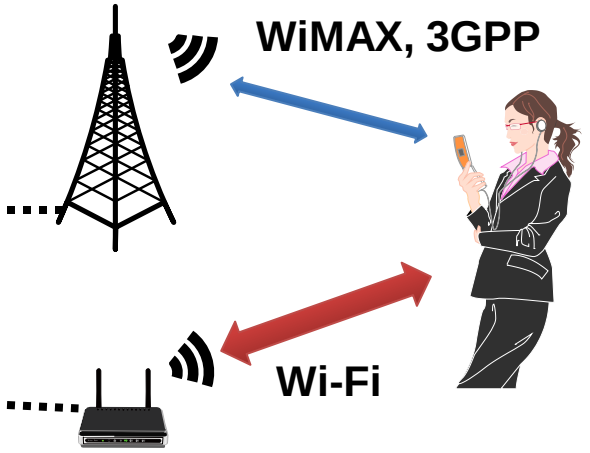
Research Background

MN: Mobile Node
Mobile PPC: Mobile Peer-to-Peer Communication

- **IP Mobility**
  - maintains transport session if an IP address of MN is changed
    ex) Mobile IP, Mobile PPC (Our original technology)

- **Change of IP address occurs due to;**
  A) Horizontal handoff (change an access point with the same device)
  B) Vertical handoff (switch between different types of devices)

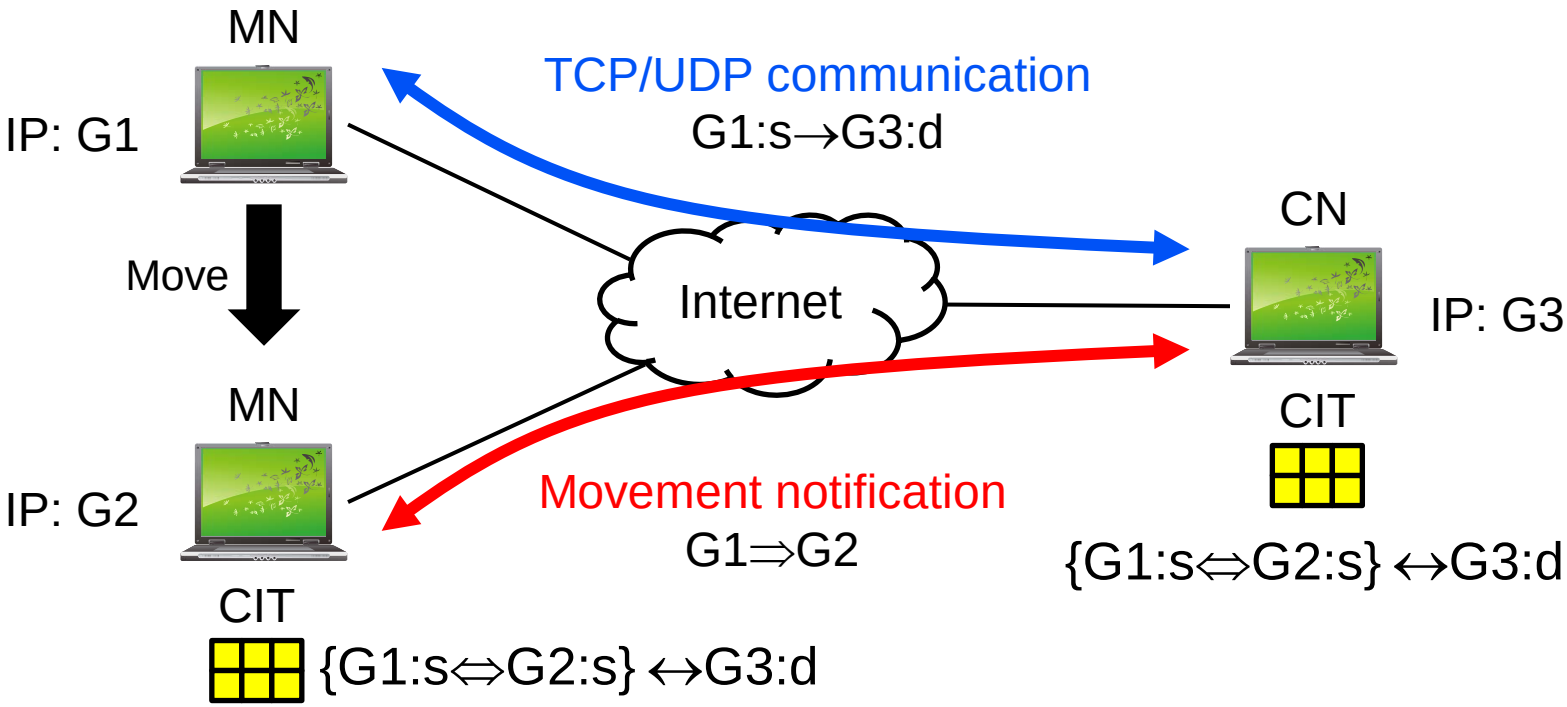




Introduction of Mobile PPC (1/2)

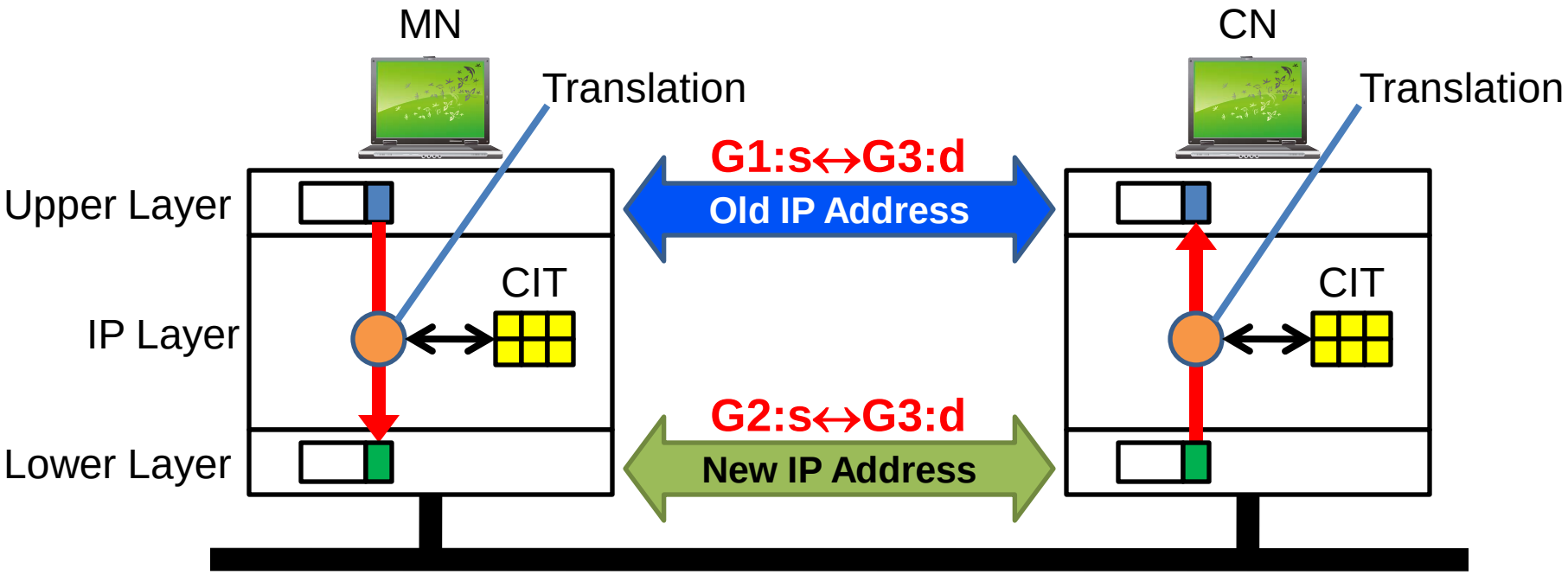
CN: Correspondent Node
CIT: Connection ID Table

- **MN and CN create CIT at the start of communication**
- **MN notifies CN of its migration directly after the move**
  - They update own CIT
    - records the relationship between the old and new addresses



Introduction of Mobile PPC (2/2)

- **They translate the IP address of TCP/UDP packet according to updated CIT in the IP layer**
  - When sending: Old address “G1” → New address “G2”
  - When receiving: New address “G2” → Old address “G1”



Target of This Research

NAT: Network Address Translator

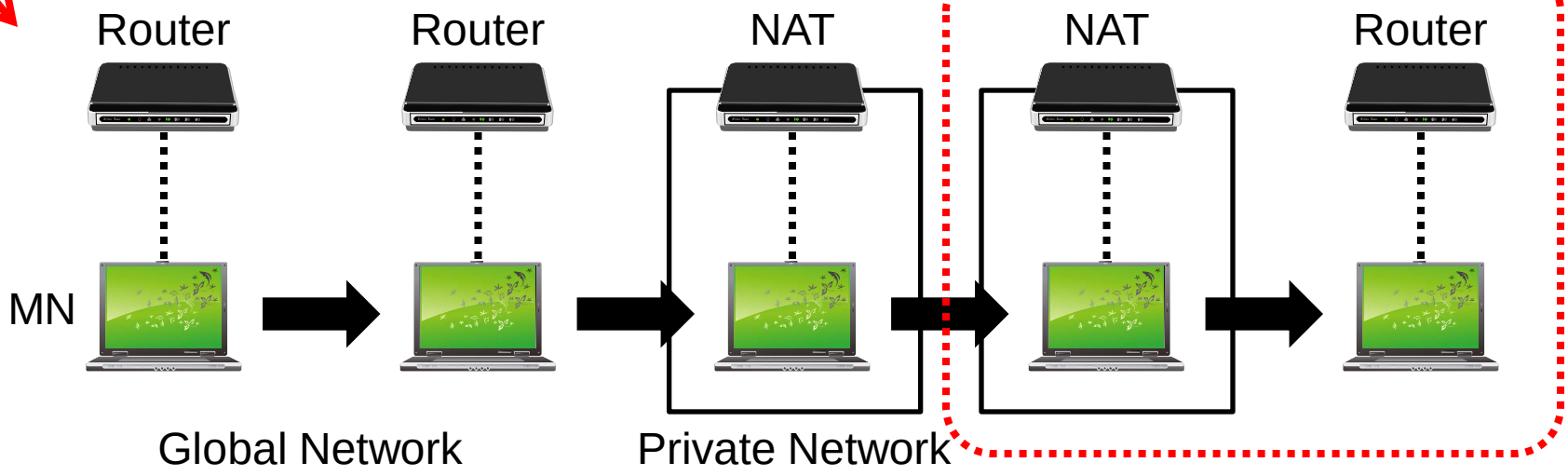
Mobility Patterns

From	To
IPv4 Network	IPv4 Network
IPv6 Network	IPv6 Network
IPv4 Network	IPv6 Network
IPv6 Network	IPv4 Network

Our group researches all cases

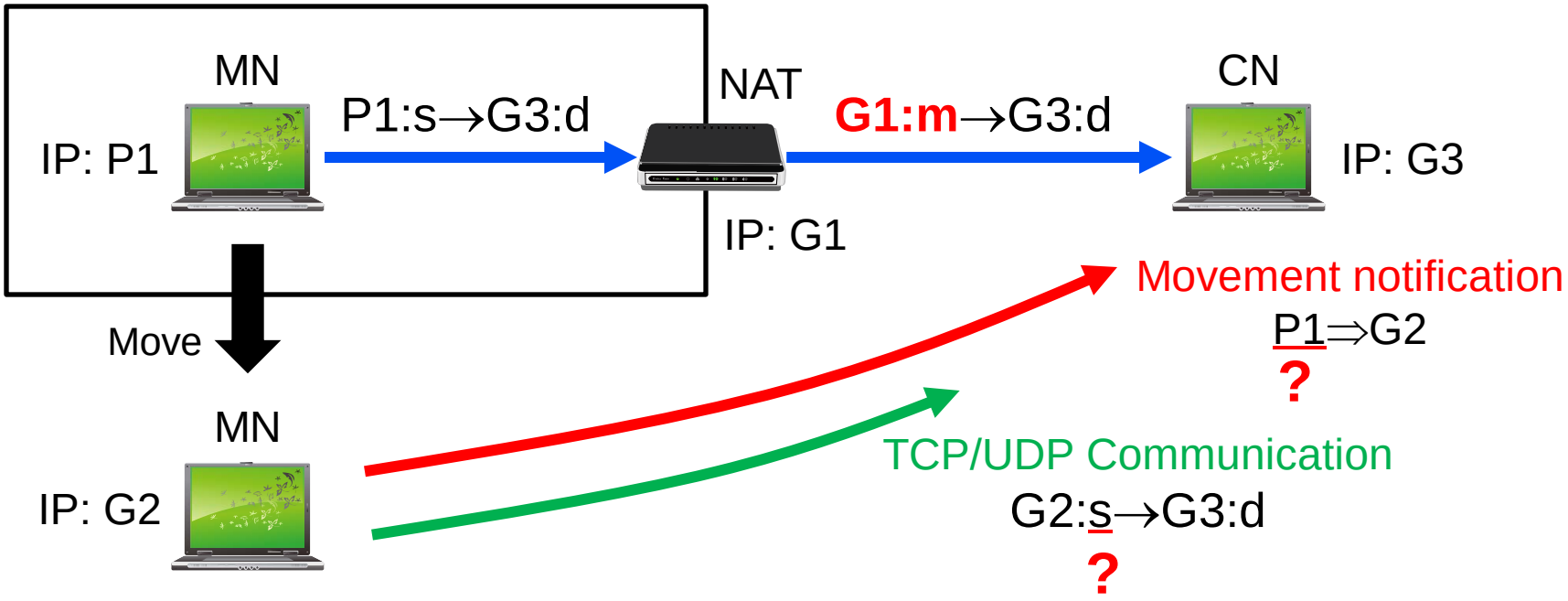
Target of this research

Focus my talk on this case



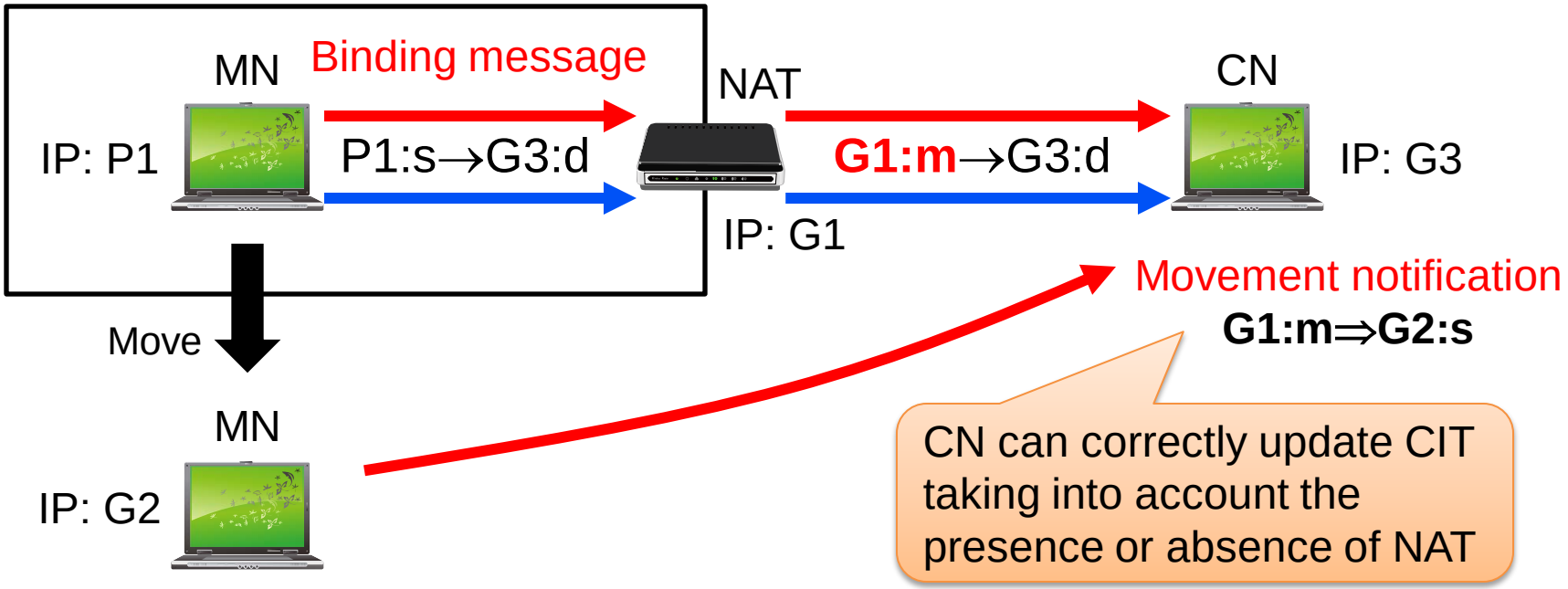
IP Mobility Issues in IPv4

- **NAT translates an IP address and a port number of TCP/UDP packet**
  - Connection information that MN and CN recognize is different
    - CN does not recognize the private IP address notified by MN and the original source port number
      ➔ CN's CIT record cannot be correctly updated

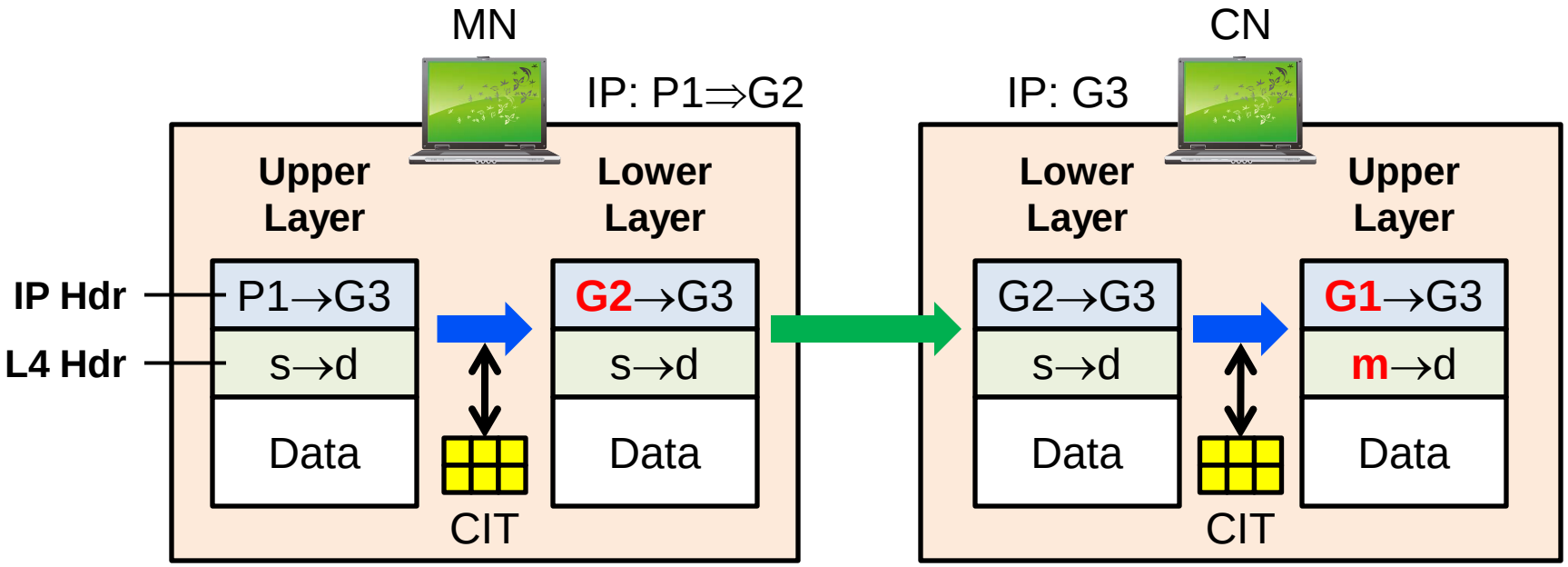


Proposed Method

- **Applying the principle of "Hole Punching" to Mobile PPC**
  - Hole punching is a NAT traversal technology
  - Binding message makes NAT mapping information
    - MN can obtain the mapped address "G1:m" by the response
  - Notification includes the mapped address and the port number



Address Transition Process

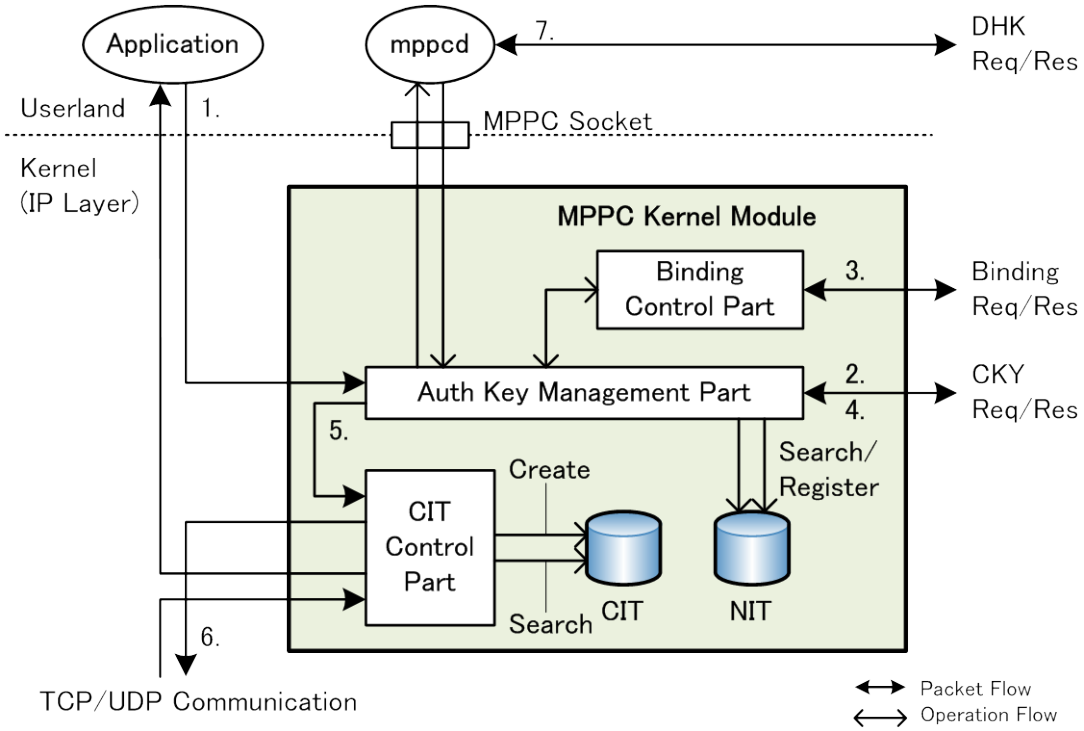


- **Change in the IP address and the port number are concealed from the upper layer**
  - Communication can be continued even if MN moves over NAT

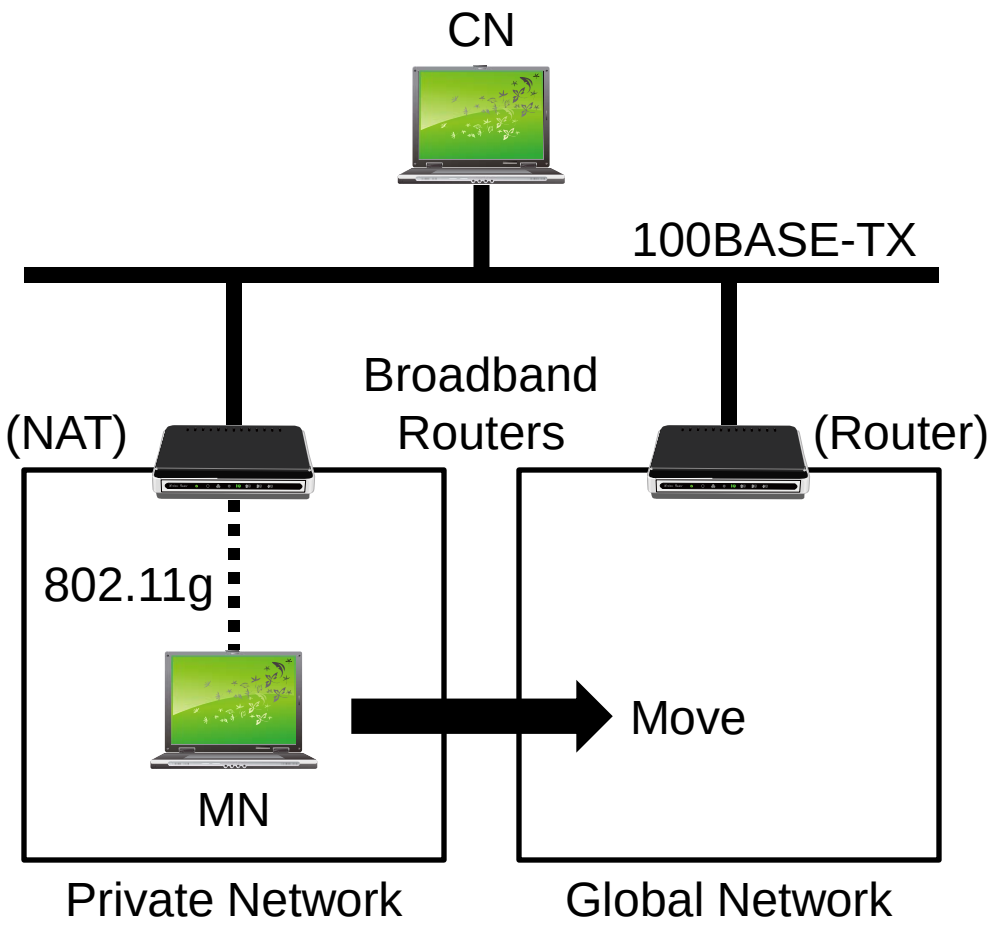
Implementation

CKY: Cookie exchange
DHK: Diffie-Hellman Key exchange
CU: CIT Update
NIT: Node Information Table

- **IP layer of FreeBSD 7.0-RELEASE**
  - Kernel module
    - CIT control
    - Authentication-Key management
    - Mobility management
    - **Binding control** (newly implemented)
- **Userland**
  - "mppcd" daemon (remains unchanged)
    - Move detection
    - DH key exchange
    - Authentication-Key generation



Trial System Configuration



Specifications

MN (ASUS Eee PC S101)
CPU : Atom N270 (1.6GHz)
Memory : 1GB
NIC : IEEE802.11g

CN
CPU : Geode LX800 500MHz
Memory : 256MB
NIC : 100BASE-TX

Broadband Router
BUFFALO WZR-G144NH
NAT Feature: Enable/Disable

* Commercially available Wi-Fi broadband router

Results of Demonstration

From	To	UDP	TCP
Global	Global	○	○
Global	Private	○	✕
Private	Private	○	✕
Private	Global	○	○

- **Case of UDP:**
  - Communication can be continued with all mobility patterns
- **Case of TCP:**
  - Communication can be continued with just mobility patterns that MN moves to a global network

Remaining Problem

- **Firewall intercepts TCP packets**
  - Many of NAT routers implements a strong firewall, **Stateful Packet Inspection (SPI)**
    - Checks the consistency between outbound and inbound packets (Bad IP addresses, incorrect TCP flags, erroneous sequence number, etc…)
    - Their packets are discarded if the session is not based on correct procedures

  ➔In TCP communication, it is thought to be necessary to make binding sequence **a TCP three-way handshake**
    - TCP sequence numbers of binding messages must be corresponded with TCP communication after MN moves to a private network

Conclusion

- **IP mobility issues in IPv4 network**
  - IP address mismatching occurs because of translation by NAT

  ➔We apply the principle of hole punching for Mobile PPC

- **Implementation and the results of trial system**
  - MN can completely move from a private network to a global network
  - MN cannot continue TCP communication after moving to a private network because of SPI firewall on NAT

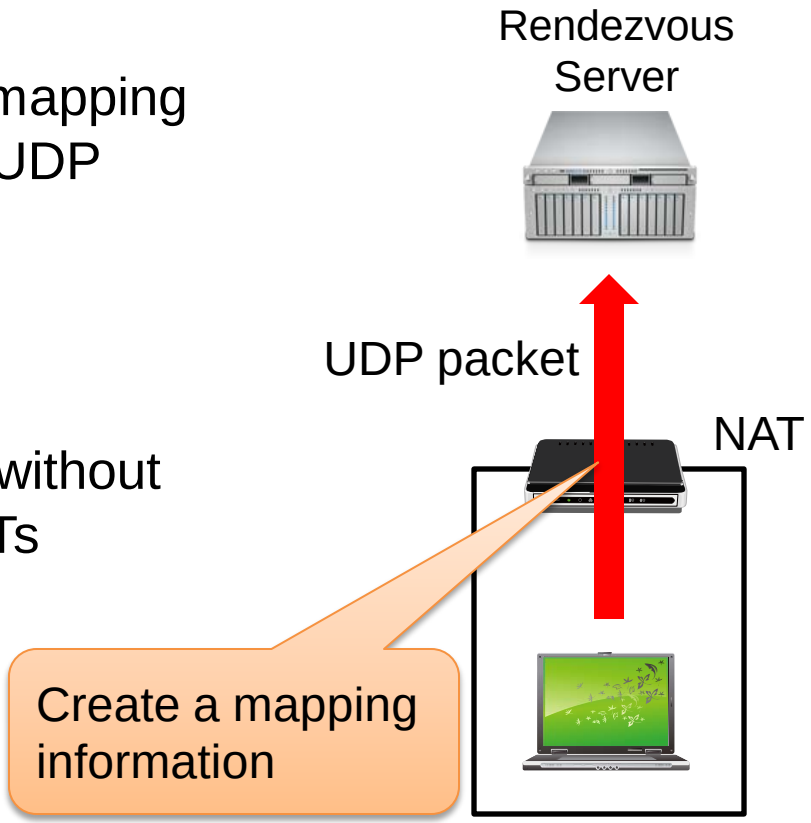
- **Future works**
  - Implementation of new TCP binding sequence for SPI

Appendix

Hole Punching

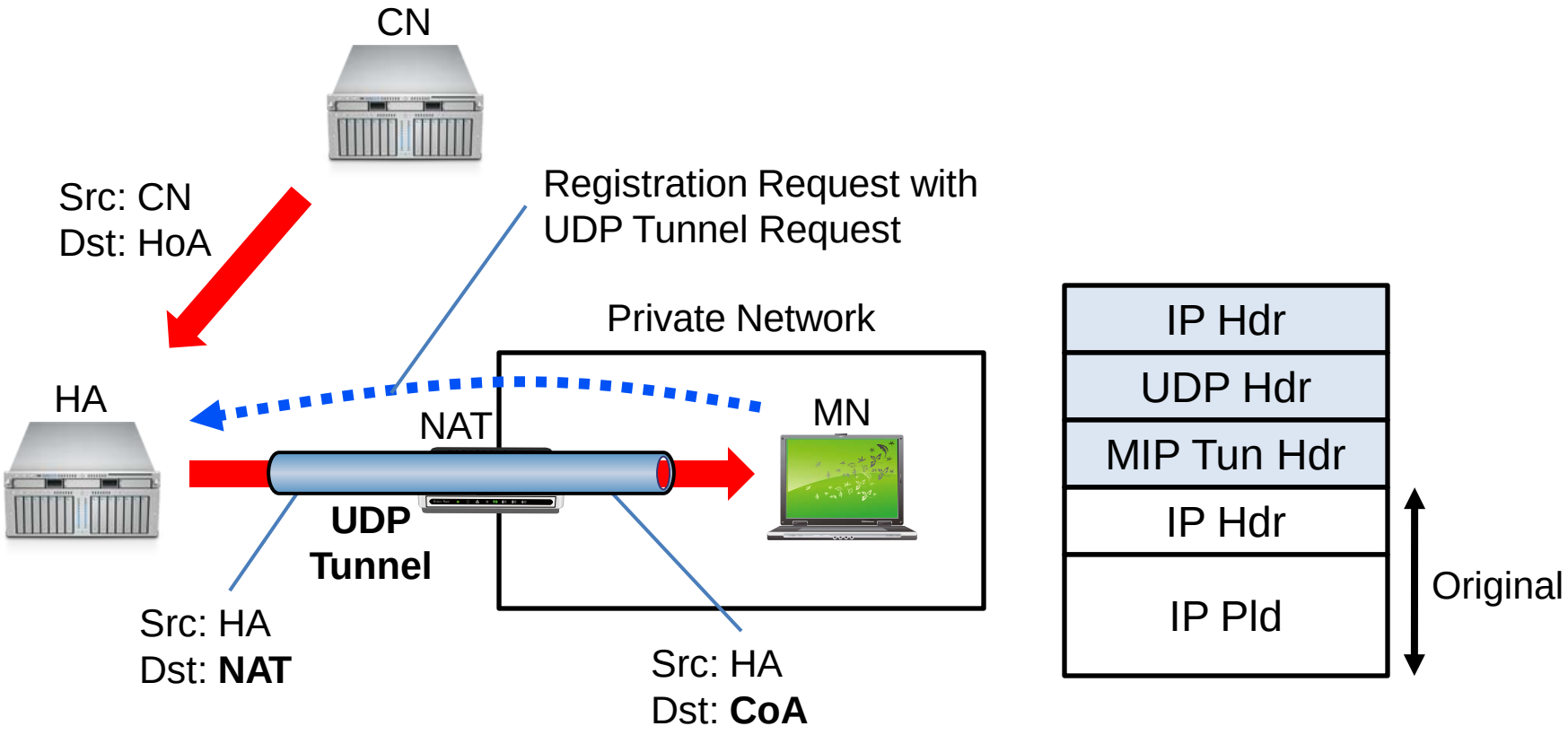
- **NAT traversal technology**
  - It has been applied to some conventional technologies
    - Simple Traversal of UDP Through NATs (STUN)
    - Teredo
  - An external node sends to the mapping address in order to establish a UDP session with an internal node

- **Advantages**
  - Solving NAT traversal problem without any modification to existing NATs
  - Good throughput performance



NAT Traversal for Mobile IP

- **NAT traversal by using UDP tunneling** (RFC3519)



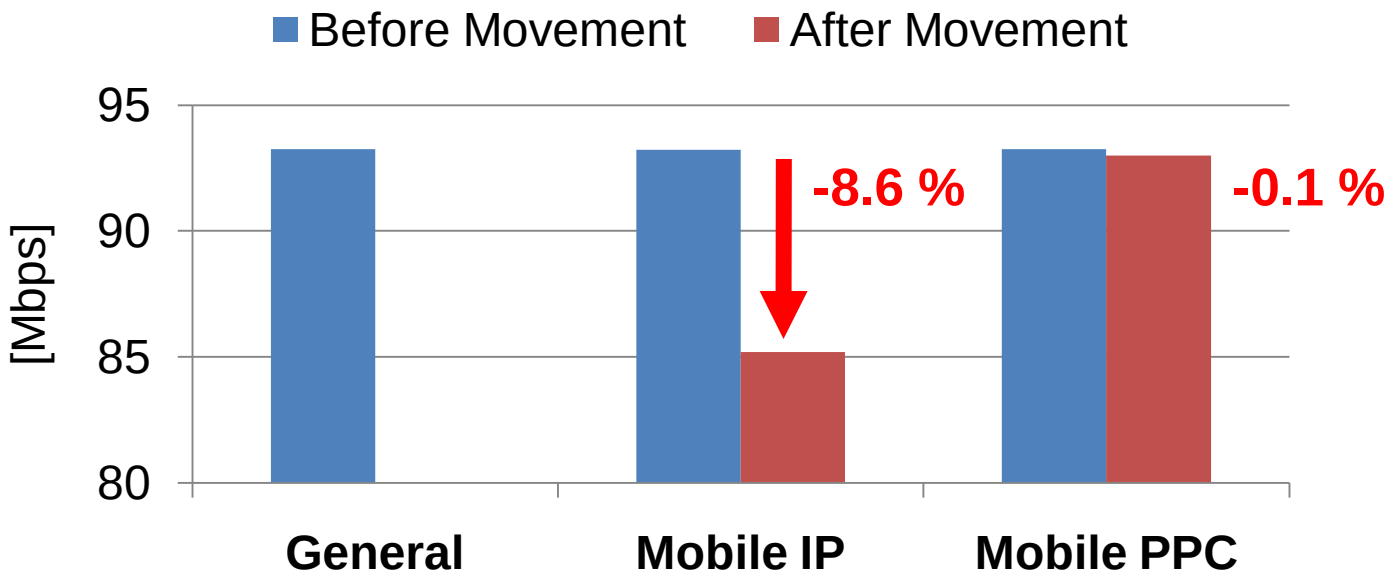
HA : Home Agent
HoA : Home Address
CoA : Care-of Address

Benefits of the proposed method

- **End-to-End “Hole Punching”**
  - No rendezvous server (like a STUN/TURN server)
    - No communication delay
    - No single point of failure
  - Supporting all types of NATs
    - Cone NAT/Symmetric NAT
- **Only address translation**
  - No encapsulation
    - High throughput

Comparison of Throughput

- “Original” Mobile IP & Mobile PPC
  - In case of Mobile IP, MN moves between global networks (no NAT traversal)
  - In case of Mobile PPC, MN moves from a private network to a global network



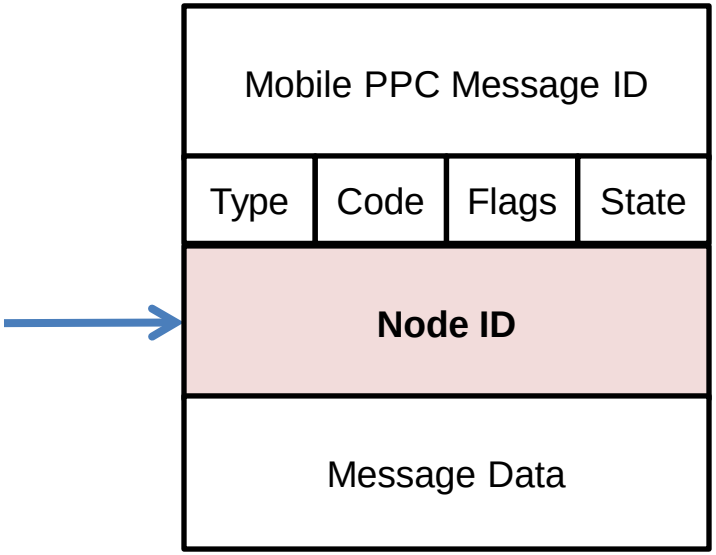
Message Format Extension

- **Adding “Node ID”**
  - Adopt UUID (Universally Unique Identifier) [RFC4122]
    - Is created from FQDN of each node by a hash function
  - All nodes have a unique identification number
    - They are completely unaffected by change of an IP address

MN

FQDN: alice.example.com

Node ID:
550e8400-e29b-41d4-a716-446655440000



Performance

CKY: Cookie Exchange
DHK: Diffie-Hellman Key Exchange

Initial delay

i. 1st Cookie Exchange = 3.55 msec
ii. Binding Negotiation = 1.72 msec
iii. 2nd Cookie Exchange = 0.68 msec

TOTAL: 7.43 msec

(Generic system: 2.81 msec)

iv. DH Key Exchange = 54.5 msec
v. Authentication-Key Generation
- MN = 5.4 msec
- CN = 38.9 msec

Last two processes execute in parallel with the started communication