icact 2013

http://www.icact.org/

IEEE

THE 15 th INTERNATIONAL CONFERENCE ON ADVANCED COMMUNICATION TECHNOLOGY

Smart Services with Internet of Things!

Phoenix Park, Pyeongchang. Republic of Korea
Jan. 27~30, 2013

IEEE Catalog Number : CFP13561-CDR
ISSN 1738-9445
ISBN 978-89-968650-0-1

MAIN
MESSAGE
OBJECTIVES
COMMITTEE
SESSION OVERVIEW
TUTORIAL Ⅰ
PLENARY SESSION
TUTORIAL II
TECHNICAL SESSIONS
AUTHOR INDEX
REVIEWER LIST
TRANSPORTATION
FLOOR PLAN FOR ICACT2013
OTHER INFORMATION

Organizers
GIRI

Sponsors
IEEE COMMUNICATIONS SOCIETY
NIA
ETRI
KICS
IEEK ComSoc
Korean Institute of Information Scientists and Engineers
OSIA
Korea Institute of Information Security & Cryptology

Session 3F: NGN and Future Network 2 **Room : Phoenix Ⅲ**

Chair : Dr. Sun-Moo KANG, National Information Society Agency, Korea

03F-01 0107	Application-aware Optimization of Networking Protocols for Data Centers
	SeungCheol Kim*, Sae Hyong Park**, Yongwoon Song*, Woocheol Cho*, Hyuk-Jun Lee*
	*Sogang University, Korea, **Future Internet Platform Research Team ETRI, Korea
03F-02 0381	Evolving Bio Plausible Design With Heterogeneous Noc
	Nithya Narayanamurthy*, J. K. Periasamy *
	*Sri Sairam Engineering College, India
03F-03 0110	Design of a Near-Minimal Dynamic Perfect Hash Function on Embedded Device
	Derek Pao, Xing Wang and Ziyan Lu
	City University of Hong Kong, Hong Kong
03F-04 0061	Host Identifier and Local Locator for Mobile Oriented Future Internet: Implementation Perspective
	Nak Jung Choi*, Ji In Kim**, Seok Joo Koh*
	*Kyungpook National University

Day 3 : January 29 (Tuesday)

Time **10:00 ~ 11:30**

Session 4A: Mobile Communication 1 **Room : Ruby**

Chair : Prof. Sangho Choe, The Catholic Univ of Korea, Korea

04A-01 0286	Scenario-Oriented Parameters Optimization and Configuration Methods and Processes in TD-LTE
	Cuibo Yu, XueRong Gou, Cong Zheng, Wen Ye
	Beijing University of Posts and Telecommunications, China
04A-02 0468	Comparative Study on Radio Wave Propagation Models for 4G Network
	Rosilah Hassan, Faisal Mohd Amin
	Universiti Kebangsaan Malaysia, Malaysia
04A-03 0134	Assessment of Quasi-Permanently Vacant Channels in Mobile Communication Bands for Cognitive Radio
	Neeta Nathani 1, G.C. Manna 2, S.B.Mule 1
	1 G.H. Raisoni College of Engineering, India, 2 Indian Telecommunication Service, BSNL, India,
04A-04 0473	Gateway Service for Integration of Heterogeneous Networks using Different Interworking Solutions
	Hyunho Park*, Hyeong Ho Lee*, H. Anthony Chan**
	*Electronics and Telecommunications Research Institute (ETRI), Korea, ** Huawei U.S.A . North American Headquarters, U.S.A
	Verbal Autopsies, Mobile Data Collection for Monitoring and Warning Causes of Deaths

Host Identifier and Local Locator for Mobile Oriented Future Internet: Implementation Perspective

Nak Jung Choi*, Ji In Kim**, Seok Joo Koh*

* School of Computer Science and Engineering, Kyungpook National University, Korea

peaceful7007@gmail.com, jiin16@gmail.com, sjkoh@knu.ac.kr

Abstract—**It is envisioned that the future Internet will be evolved to mobile oriented network environment, and the mobility support is a key issue in the design of future Internet. This paper proposes an architecture of Host Identifier and Local Locator (HILL) for the future mobile-oriented Internet. The proposed HILL architecture is implemented over Linux platform by using netfilter, iptables, and 6to4 tunneling. In addition, the handover performance is analyzed over the experimental testbed. In the experimental result, we can see that the handover operation is completed within one second.**

Keywords— **Host ID; Local LOC; ID-LOC separation; implementation**

I. INTRODUCTION

With an explosive growth of the number of subscribers of mobile cellular systems and also other wireless data systems such as WiFi and WiMAX, the mobile networks snow become the key driver toward the future Internet. In addition, a variety of wireless access networks, such as adhoc network and sensor network, are emerging, and they will be the major access means to future Internet.

However, it is noted that the current Internet was historically designed for fixed network environment, rather than for the mobile network environment. This has enforced Internet to add a lot extensional features to satisfy the mobility requirements, as shown in the example of Mobile IP (MIP) [1, 2]. However, this patch-on approach seems to be just a temporal heuristic to the problems in the mobile environment, rather than an optimization, a variety of research activities have been made to design the future Internet for mobile environment, which include eMobility [3], 4WARD [4], FIND [5], MobilityFirst [6], GENI [7], and AKARI [8]. It is also noted that many challenging works are in progress with the identifier-locator separation principle, as shown in Host Identity Protocol (HIP) [9], Locator-Identifier Separation Protocol (LISP) [10], and Identifier-Locator Network Protocol (ILNP) [11].

With these observations, we design the architecture of Mobile Oriented Future Internet (MOFI) [12] to support the mobile environment of future Internet. The MOFI is designed with the following functional blocks: Host Identifier and Local Locator (HILL), Query-First Data Delivery (QFDD), and Dynamic and Distributed Mapping System (DDMS). The details of the MOFI architecture are described in [12].

This paper discusses the HILL architecture for ID-LOC separation over the MOFI. Based on the HILL architecture, the QFDD in the data plane and DDMS in the control plane can be supported. In this paper, we focus on the implementations of HILL architecture with some experimental results.

This paper is organized as follows. Section 2 briefly summarizes the HILL architecture. In Section 3, we describe the Linux-based implementation of HILL architecture. Section 4 discusses the experimentations over testbed. Section 5 concludes this paper.

II. HILL ARCHITECTURE

A. *Identifier And locator*

In MOFI, Host Identifier (HID) is separated from Locator (LOC). HID is used to identify a host in Internet. We assume that a HID contains the domain information of an ISP that a host belongs to. LOC represents the location of a host in the network and it is used for delivery of data packets in the network. In addition, a HID shall be globally unique in Internet, whereas a LOC is a locally routable address that has only to be locally unique in the concerned network. In MOFI, the mappings between HIDs and LOCs will be managed in the dynamic and distributed manner.

The ID-LOC separation structure of MOFI is different from the other ID-LOC separation schemes such as host Identity Protocol (HIP) [9], Locator Identifier Separation Protocol (LISP) [10], and Identifier Locator Network Protocol (ILNP) [11]. First, a HID is allocated to a host itself, not its interface. Next, a LOC is given to the network (rather than a host) that a host is attached to. Moreover, LOC is ‘local’, rather than ‘global’

1) *Host identifier (HID)*

In MOFI, it is assumed that a host should be identified by HID and given by an ISP in a static and secure manner. At present, we consider the format of 128-bit (16-byte) HID, which is designed for compatibility of IPv6 applications

2) *Locator (LOC)*

LOC is used for delivery of data packets between objects in the network. In MOFI, the IPv4/IPv6 addresses of the Access Router (AR) and Gateway (GW) are used as LOCs. There IP address may be private in the network. The LOC is used for end-to-end communication between the two hosts through one or more networks. In mobile environments, a host with a single HID may change its LOCs by movement

B. Functional Entities

1) Host identifier (HID)

HID. The end-to-end communication is governed by HID. Each data packet has the MOFI header which includes the HIDs of source and destination hosts.

For data delivery, a host sends or receives data packets to or from AR by using the Access Delivery Protocol (ADP). AR receives the data packets from its local host and delivers them to the network, and it receives data packets from the network and forwards the packets to the local hosts by using ADP. The host may have two or more Access LOCs (A-LOCs) in the case of host multi-homing.

Each ISP domain may use its own routing scheme in its backbone network, which is called Backbone Delivery Protocol (BDP). BDP is responsible for packet routing between ARs, possibly via one or more backbone routers. In MOFI, the current IPv4/IPv6 protocol is used as BDP. MOFI uses the LOC-based local delivery. The LOC used in the BDP may be local (private). Each LOC has only to be unique within the ISP domain.

2) Access Router (AR)

Each AR keeps its local binding cache that maintains the mapping information of HID and A-LOC for the local hosts. AR also has its data forwarding cache with the list of HID-LOC for data packet forwarding to active hosts, which will be updated in the signaling operations for LOC query.

C. Data Delivery Model

In HILL, each host has a globally unique HID, by which global communication is accomplished. In the meantime, one or more LOCs are used for packet routing in the network. Each LOC may be used locally in the transit networks, without any assumption on global uniqueness of LOC. A-LOC is used for forwarding of data packet between hosts and AR in the access network. The format of A-LOC is specific to the underlying access network.

Figure 1 shows the packet delivery operations with Global HID-based Communication and Local LOC-based Routing in the HILL architecture.

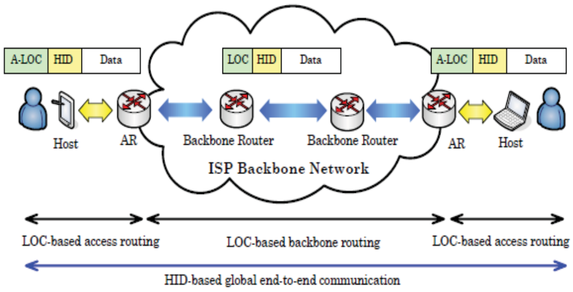


Figure 1. HID-based communication and LOC-based routing

III.IMPLEMENTATIONS

In this section we discuss implementations of HILL architecture over Linux platform. The HILL architecture is implemented on Host and AR.

A. Host

Figure 2 shows the network model for implementation of HILL. Each host has the protocol stack of MOFI/ADP, rather than the legacy IP, in which the MOFI layer is responsible for HID-based global communications between two end hosts and the ADP layer performs the packet delivery between host and AR. The packet delivery between AR and BR or between BRs is governed by BDP, as shown in the figure.

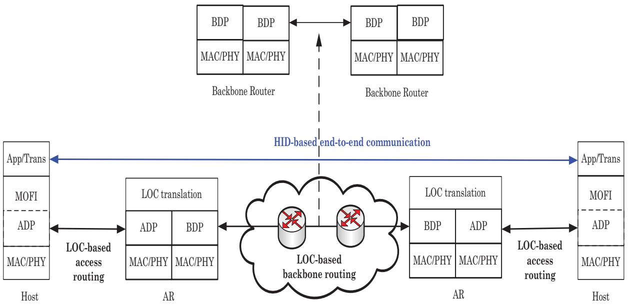


Figure 2. Network mdoel for implementations

For implementation of host, we use the Linux Ubuntu 10.04 version. The HID is implemented by using the well-known 6to4 tunneling scheme. This is because most of the Linux platforms support the 6to4 tunneling scheme [13].

The conventional method of making the IPv6 6to4 tunnel address from an IPv4 address is shown in Figure 3. In the figure, an IPv4 address is translated into hexadecimal digits and inserted into 3~6 byte positions of the IPv6 address.

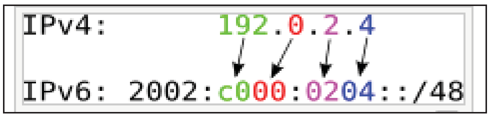


Figure 3. Typical way of making a 6to4 tunnel address

In HILL, an IPv6 address for the 6to4 tunneling is used as HID, whereas an IPv4 address is used as LOC. The 128-bit HID includes 2-byte prefix for 6to4 tunneling, 4-byte AS number, and subscriber ID (10 bytes), as shown in Figure 4.

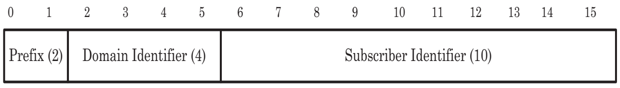
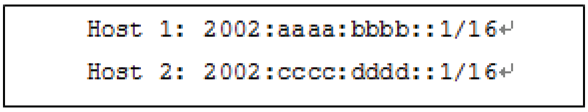


Figure 4. 128-bit HID format

In implementations, each HID has the following format, as shown in Figure 5. In the figure, a HID includes the 4-byte AS number in the 3-6 byte positions. Note that aaaa:bbbb and cccc:dddd represent the 4-byte AS numbers and ‘1’ is a subscriber ID.



```
Host 1: 2002:aaaa:bbbb::1/16↵
Host 2: 2002:cccc:dddd::1/16↵
```

Figure 5. 128-bit HID format used in HILL

The legacy Linux platform (at AR) does not support this type of 6to4 tunnel address. Accordingly, the associated AR modules in Linux need to be modified is required to support the HID, which will be described in the subsequent section.

B. Access Router (AR)

In this section we describe how to implement an Access Router (AR) over the HILL architecture. The implementation of AR is based on netfilter and iptables [14].

Figure 6 shows the protocol model of AR. It is assumed that each AR uses ADP (and A-LOC) for packet delivery between AR and host, and it uses BDP (and LOC) between ARs. It is noted that each AR performs the LOC query and update operation to find the location of the correspondent AR, before data transmission. This is called “Query-and-Encapsulation”, which is described in [12] more in details.

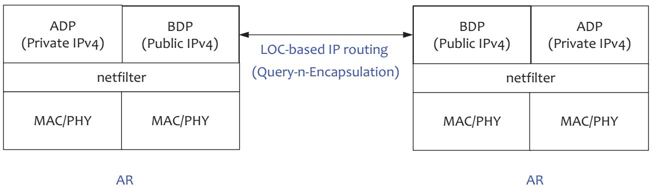


Figure 6. Protocol model of AR

When a data packet is delivered from a host, AR should translate the A-LOC (for ADP) to LOC (for BDP). In our implementation, a private IPv4 address is used as A-LOC, and a public IPv4 address is used as LOC. For this LOC translation, we use the netfilter and iptables over the Linux 10.04 version

1) Host and Network

The netfilter is a well-known tool used in Linux, which is used for LOC (and header) translation in the implementation.

Figure 7 shows the netfilter functions given in Linux. In the figure, the shaded boxes (functions) are used to support the LOC translation at AR. Those functions are called the netfilter hooking points. When a data packet arrives from the host, the AR performs the LOC translation (and packet encapsulation) from A-LOC (for APD) to LOC (for BDP). The encapsulated data packet will be delivered to the correspondent AR, and the AR will perform the LOC translation from LOC to A-LOC with the netfilter functions

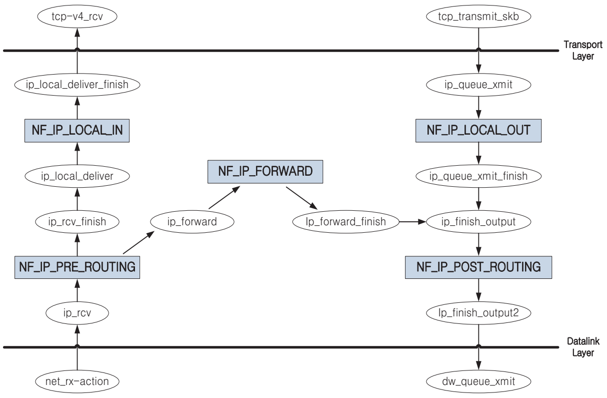
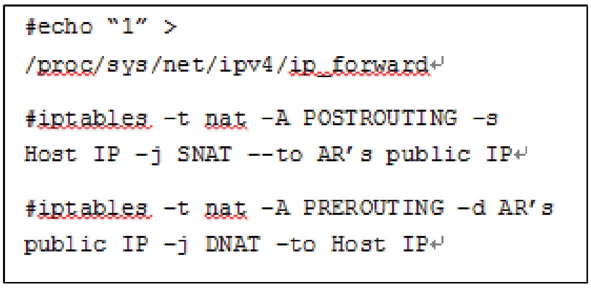


Figure 7. Netfilter hooking points at AR

For implementation of HILL, the iptables is used together with netfilter. Typically, iptables are used for packet filtering, such as packet drop and packet forward. In the HILL architecture, the iptables are used for packet forwarding between AR and host.

2) Iptables

Figure 8 shows the use of iptables at AR to perform the packet forwarding to host. In the figure, the iptables are configured to support the packet delivery based on the HILL architecture at AR.



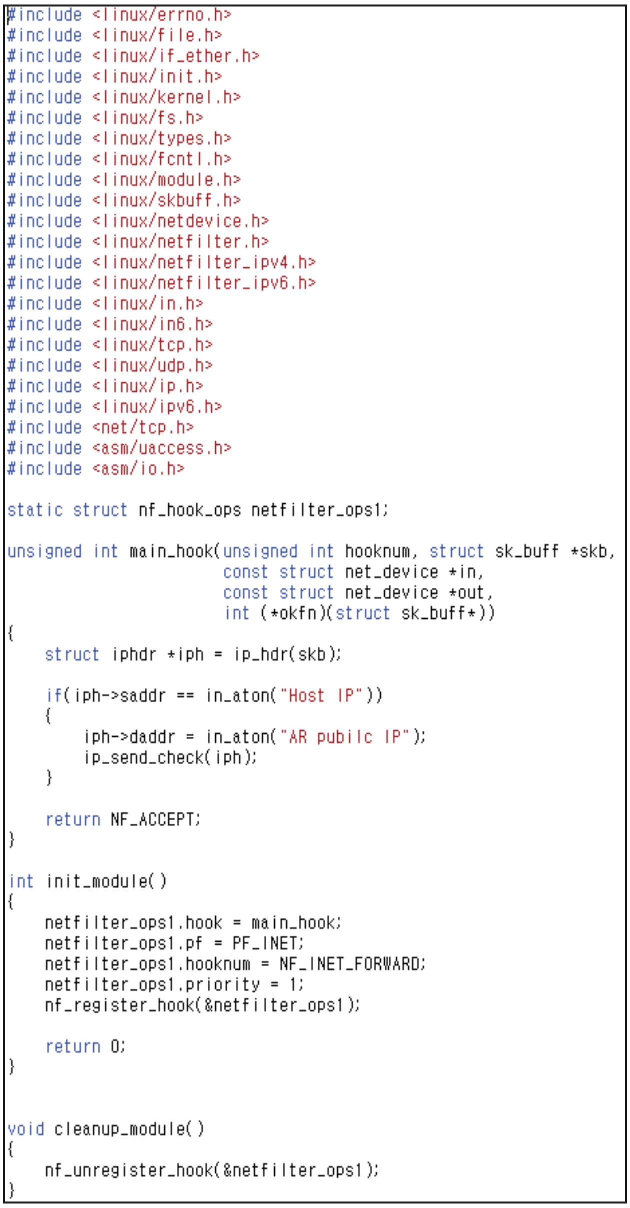
```
#echo "1" >
/proc/sys/net/ipv4/ip_forward↵

#iptables -t nat -A POSTROUTING -s
Host IP -j SNAT --to AR's public IP↵

#iptables -t nat -A PREROUTING -d AR's
public IP -j DNAT -to Host IP↵
```

Figure 8. Configuration of iptables at AR

B.2. Kernel Compile

With the configured netfilter and iptables, the Linux kernel needs to be re-compiled. In the Kernel compile, only the concerned (modified) modules will be compiled by using the ‘micro-kernel’ compilation process, rather than the entire compilation. Note that this micro-compilation can reduce the phenomenon of kernel panic.

Figure 9 shows the kernel modules associated with the LOC (IP address) change at AR. Note that this code is a part of the entire source code for IP address change. In the figure, the A-LOC (and the concerned IP header) of the host is translated to the LOC (and the concerned IP header)



```
#include <linux/errno.h>
#include <linux/file.h>
#include <linux/if_ether.h>
#include <linux/init.h>
#include <linux/kernel.h>
#include <linux/fs.h>
#include <linux/types.h>
#include <linux/fcntl.h>
#include <linux/module.h>
#include <linux/skbuff.h>
#include <linux/netdevice.h>
#include <linux/netfilter.h>
#include <linux/netfilter_ipv4.h>
#include <linux/netfilter_ipv6.h>
#include <linux/in.h>
#include <linux/in6.h>
#include <linux/tcp.h>
#include <linux/udp.h>
#include <linux/ip.h>
#include <linux/ipv6.h>
#include <net/tcp.h>
#include <asm/uaccess.h>
#include <asm/io.h>

static struct nf_hook_ops netfilter_ops1;

unsigned int main_hook(unsigned int hooknum, struct sk_buff *skb,
                       const struct net_device *in,
                       const struct net_device *out,
                       int (*okfn)(struct sk_buff*))
{
    struct iphdr *iph = ip_hdr(skb);

    if(iph->saddr == in_aton("Host IP"))
    {
        iph->daddr = in_aton("AR pubilc IP");
        ip_send_check(iph);
    }

    return NF_ACCEPT;
}

int init_module()
{
    netfilter_ops1.hook = main_hook;
    netfilter_ops1.pf = PF_INET;
    netfilter_ops1.hooknum = NF_INET_FORWARD;
    netfilter_ops1.priority = 1;
    nf_register_hook(&netfilter_ops1);

    return 0;
}


void cleanup_module()
{
    nf_unregister_hook(&netfilter_ops1);
}
```

Figure 9. Kernel module for IP address change

IV. EXPERIMENTATIONS

This section discusses how to make the testbed and how to test the applications on the HILL architecture. In testbed experimentation, we employ the VLC video player as the upper layer application on top of UDP.

A. Identifier And locator

Figure 10 shows the testbed network configuration, in which the two hosts (Host 1 and Host 2) and three routers (AR1, AR2, and AR3) are used.

In the testbed, Host1 acts as the video server (sender) and Host2 plays a role of the video client (receiver). In the figure, the HID binding and LOC update represent the control operations of MOFI (please refer to [12] for more details).

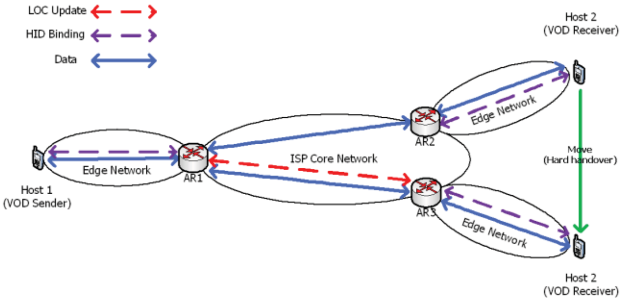


Figure 10. Testbed network

During experimentation, Host2 will move from AR2 to AR3 by handover. By performance analysis, the handover delay will be measured.

Figure 11 shows the snapshot of the real machines associated with in the testbed of Figure 10.

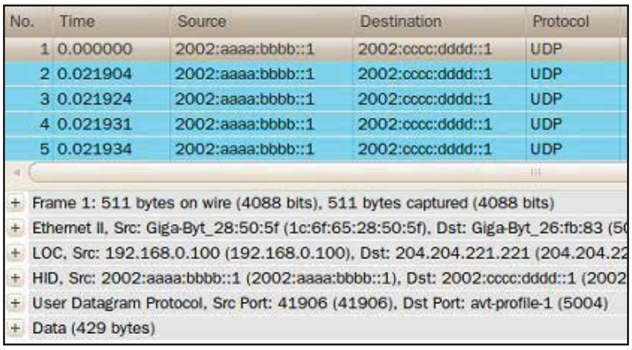
Figure 11. Testbed snapshot

In experimentation, the test scenario can be summarized as follows. First, Host1 is connected to AR1, and Host2 is attached to AR2. For VLC video streaming services, Host2 receives the video data traffics from Host1 by using UDP. During the video transmission, Host2 moves from AR2 to AR3, and the handover delay is measured.

B. Experimental Result

Figure 12 show the packet capturing results for the packets from Host1 to AR1, by using the Wireshark [15]. Similarly, Figure 13 and 14 show the packet capturing results from AR1 to AR2, and from AR2 to Host2, respectively. In the figures, we can see that Host1 uses the HID of 2002:aaaa:bbbb::1 and Host2 uses the HID of 2002:cccc:dddd::1.

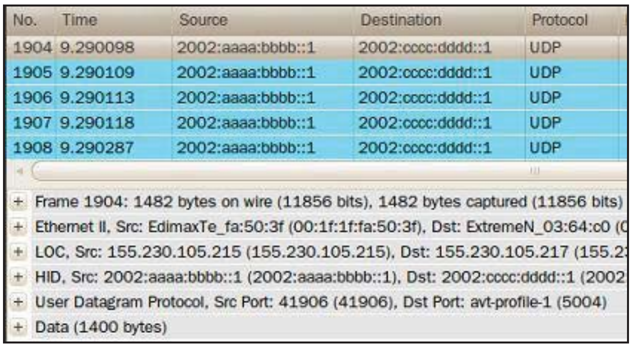
On the other hand, LOCs are changed in each network region. For example, in Figure 12, Host1 uses the A-LOC of 192.168.0.100. In Figure 13, AR1 uses LOC of 155.230.105.215 (public IP address), and AR2 uses LOC of 155.230.105.217 (public IP address). It is noted that LOCs are changed by network region, whereas HIDs of Host1 and Host2 are not changed.



No.	Time	Source	Destination	Protocol
1	0.000000	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
2	0.021904	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
3	0.021924	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
4	0.021931	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
5	0.021934	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP

```
+ Frame 1: 511 bytes on wire (4088 bits), 511 bytes captured (4088 bits)
+ Ethernet II, Src: Giga-Byt_28:50:5f (1c:6f:65:28:50:5f), Dst: Giga-Byt_26:fb:83 (5
+ LOC, Src: 192.168.0.100 (192.168.0.100), Dst: 204.204.221.221 (204.204.22
+ HID, Src: 2002:aaaa:bbbb::1 (2002:aaaa:bbbb::1), Dst: 2002:cccc:dddd::1 (2002
+ User Datagram Protocol, Src Port: 41906 (41906), Dst Port: avt-profile-1 (5004)
+ Data (429 bytes)
```

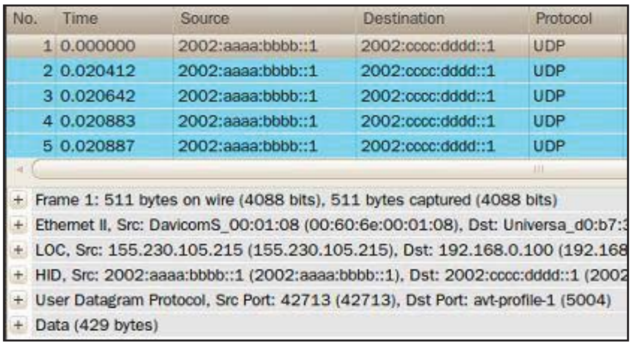
Figure 12. Packet capture (Host1 to AR1)



No.	Time	Source	Destination	Protocol
1904	9.290098	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
1905	9.290109	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
1906	9.290113	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
1907	9.290118	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
1908	9.290287	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP

```
+ Frame 1904: 1482 bytes on wire (11856 bits), 1482 bytes captured (11856 bits)
+ Ethernet II, Src: EdimaxTe_fa:50:3f (00:1f:1f:fa:50:3f), Dst: ExtremeN_03:64:c0 (
+ LOC, Src: 155.230.105.215 (155.230.105.215), Dst: 155.230.105.217 (155.2
+ HID, Src: 2002:aaaa:bbbb::1 (2002:aaaa:bbbb::1), Dst: 2002:cccc:dddd::1 (2002
+ User Datagram Protocol, Src Port: 41906 (41906), Dst Port: avt-profile-1 (5004)
+ Data (1400 bytes)
```

Figure 13. Packet capture (AR1 to AR2)

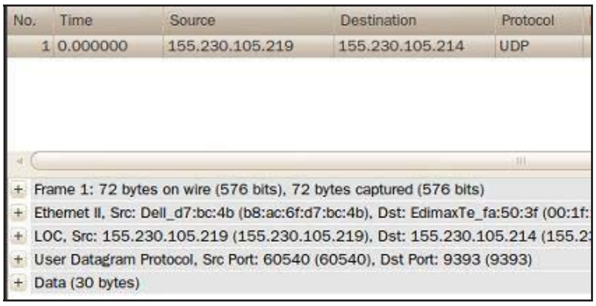


No.	Time	Source	Destination	Protocol
1	0.000000	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
2	0.020412	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
3	0.020642	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
4	0.020883	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
5	0.020887	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP

```
+ Frame 1: 511 bytes on wire (4088 bits), 511 bytes captured (4088 bits)
+ Ethernet II, Src: DavicomS_00:01:08 (00:60:6e:00:01:08), Dst: Universa_d0:b7:3
+ LOC, Src: 155.230.105.215 (155.230.105.215), Dst: 192.168.0.100 (192.168
+ HID, Src: 2002:aaaa:bbbb::1 (2002:aaaa:bbbb::1), Dst: 2002:cccc:dddd::1 (2002
+ User Datagram Protocol, Src Port: 42713 (42713), Dst Port: avt-profile-1 (5004)
+ Data (429 bytes)
```

Figure 14. Packet capture (AR2 to Host)

Figure 15 shows the packet captures for LOC update operation by handover, in which AR3 sends a LOC update message to AR1 to update the LOC of Host2 after handover. AR3 uses the IP address of 155.230.105.219 as LOC, and AR1 uses another IP address to handle the LOC update message (155.230.105.214).



No.	Time	Source	Destination	Protocol
1	0.000000	155.230.105.219	155.230.105.214	UDP

```
+ Frame 1: 72 bytes on wire (576 bits), 72 bytes captured (576 bits)
+ Ethernet II, Src: Dell_d7:bc:4b (b8:ac:6f:d7:bc:4b), Dst: EdimaxTe_fa:50:3f (00:1f:
+ LOC, Src: 155.230.105.219 (155.230.105.219), Dst: 155.230.105.214 (155.2
+ User Datagram Protocol, Src Port: 60540 (60540), Dst Port: 9393 (9393)
+ Data (30 bytes)
```

Figure 15. LOC update message (AR3 to AR1) by handover

Figure 16 shows the packet capturing results for traffics from AR1 to AR3 after handover. The concerned LOCs are changed from 155.230.105.215 to 155.230.105.219, whereas HIDs of Host1 and Host2 are not changed.

No.	Time	Source	Destination	Protocol
1	0.000000	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
2	0.000116	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
3	0.000239	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
4	0.000362	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP
5	0.000485	2002:aaaa:bbbb::1	2002:cccc:dddd::1	UDP

```
+ Frame 1: 1482 bytes on wire (11856 bits), 1482 bytes captured (11856 bits)
+ Ethernet II, Src: ExtremeN_03:64:c0 (00:04:96:03:64:c0), Dst: Dell_d7:bc:4b (b8:a
+ LOC, Src: 155.230.105.215 (155.230.105.215), Dst: 155.230.105.219 (155.23
+ HID, Src: 2002:aaaa:bbbb::1 (2002:aaaa:bbbb::1), Dst: 2002:cccc:dddd::1 (2002:c
+ User Datagram Protocol, Src Port: 45998 (45998), Dst Port: avt-profile-1 (5004)
+ Data (1400 bytes)
```

Figure 16. Packet capture(AR1 to AR3) after handover

Finally, we measured the handover delays for 10 handover trials, as shown in Figure 17. For each trial, the handover delay is calculated by difference between the reception time of data packet at AR3 and the reception time of data packet at AR2. In the figure, we can see that the handover is performed within one second for all the handover trials, and the average handover delay is 0.540 second for 10 trials.

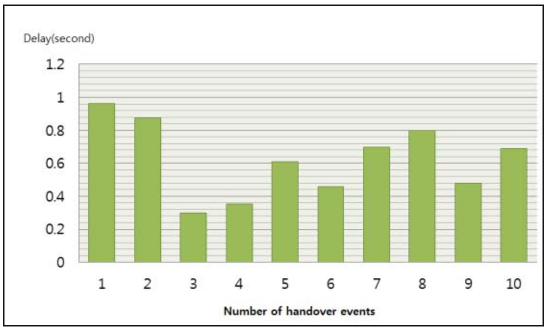


Figure 17. Handover delay

V. CONCLUSIONS

In this paper, we discuss the HILL architecture for ID-LOC separation for mobile-oriented future Internet. The details of implementation for the HILL architecture are described at the host and AR sided. For implementation, we configure the HIDs with the AS number from the 6to4 tunneling address. Each AR is implemented on the Linux platform by using netfilter and iptables so as to support the LOC and header translations between A-LOC and LOC. For experimentation, the testbed is configured and we measured the packet capturing results for the traffic between Host and AR, and between ARs. From the handover analysis, we can see that the handover delay is completed within one second.

For further works, the implementations and experimentations need to be extended by considering the real wide-area networks. In addition, the HILL architectural concept is integrated into the overall MOFI architecture more elaborately.

ACKNOWLEDGMENT

This research was supported by Basic Science Research Program of NRF(2011-0026529), ITRC support program of NIPA(NIPA-H0301-12-2004), ICT Standardization program of MKE(The Ministry of Knowledge Economy), and IT R&D support program of KCA(KCA-10913-05004).

REFERENCES

[1] IETF RFC 5944, IP Mobility Support for IPv4: Revised, November 2010
[2] IETF RFC 3775, Mobility Support in IPv6, June 2004
[3] eMobility Project, http://www.emobility.eu.org/
[4] 4WARD Project, http://www.4ward-project.eu/
[5] Future Internet Design (FIND), http://www.nets-find.net/
[6] Mobility First: Future Internet Architecture, http://mobilityfirst.winlab.rutgers.edu/
[7] Global Environment for Network Innovations (GENI), http://www.geni.net/
[8] AKARI, http://akari-project.nict.go.jp/eng/
[9] Host Identity Protocol (HIP), http://www.ietf.org/html.charters/hip-charter.html
[10] Locator Identifier Separation Protocol (LISP), http://www.ietf.org/html.charters/lisp-charter.html
[11] Identifier Locator Network Protocol (ILNP), http://ilnp.cs.st-andrews.ac.uk/
[12] Homepage of Mobile Oriented Future Internet (MOFI), http://www.mofi.re.kr
[13] 6to4 tunneling, http://en.wikipedia.org/wiki/6to4
[14] Linux netfilter hacking HOWTO, July 2002.
[15] Wireshark Homepage, http://www.wireshark.org

Nak Jung Choi received B.S. degree in Computer Science from Kyungpook National University in 2011. He is now a Master course in School of Electrical Engineering and Computer Science from Kyungpook National University. His current research interests include Internet Mobility, and Future Internet

Ji In Kim received B.S. degree in Electrical Engineering and Computer Science from Kyungpook National University in 2008. He is now a Master course in School of Electrical Engineering and Computer Science from Kyungpook National University. His current research interests include Internet Mobility, and Future Internet.

Seok Joo Koh received B.S. and M.S. degrees in Management Science from KAIST in 1992 and 1994, respectively. He also received Ph.D. degree in Industrial Engineering from KAIST in 1998. From August 1998 to February 2004, he worked for Protocol Engineering Center in ETRI. Since March 2004, he has been with the school of Electrical Engineering and Computer Science in the Kyungpook National University as an Associate Professor. He has published over 25 international journal papers with IEEE, Elsevier, and Springer-Verlag. His current research interests include mobility control in Future Internet, mobile SCTP, and mobile multicasting. He has also participated in the International Standardization as an editor in the ITU-T SG13 and ISO/IEC JTC1/SC6.