

ID-LOC 분리 기반 인터넷 구조에서 분산형 매핑 시스템의 구현 및 평가

강형우*, 최낙중*, 김지인**, 정희영***, 고석주°

Implementation and Evaluation of Dynamic and Distributed Mapping System in ID-LOC Separation Internet Architecture

Hyung-Woo Kang*, Nak-Jung Choi*, Ji-In Kim**, Heeyoung Jung***, Seck-Joo Koh°

요 약

본 논문에서는 모바일 중심의 미래 인터넷을 위한 Mobile Oriented Future Internet (MOFI) 구조를 소개하고, 호스트 Identifier(ID)와 Locator(LOC)의 분리를 토대로 설계된 MOFI 구조에서 ID와 LOC의 분산형 매핑 제어 기법을 제안한다. 제안하는 Dynamic and Distributed Mapping System (DDMS) 기법에서는 ID와 LOC의 매핑 기능을 각 도메인의 Access Router(AR)에 분산시킴으로써 기존의 Proxy Mobile IP (PMIP) 프로토콜의 중앙집중형 방식의 문제점을 개선하였다. 성능 분석을 위해 리눅스 플랫폼에서 MOFI 구조와 제안하는 매핑 제어 기능을 구현하였고, 테스트베드 실험을 통해 기존 PMIP에 비해 제안 기법이 데이터 전송 측면에서 더 효율적임을 확인하였다.

Key Words : MOFI, ID-LOC Separation, DDMS, Test-bed, Performance Analysis

ABSTRACT

In this paper, we discuss a new architecture of future Internet for mobile oriented environments, named Mobile Oriented Future Internet (MOFI). The MOFI architecture is designed with Host Identifier and Local Locator (HILL) for identifier and locator separation. Based on the HILL separation architecture, we propose a Dynamic and Distributed Mapping System (DDMS) for identifier locator mapping control. In DDMS, the mapping control function is distributed onto each access router in the domain, which is different from the centralized approach using a central anchor. For validation of the proposed MOFI-DDMS architecture, we implemented the data delivery and mapping control functions using Linux platform. From the testbed experimental results, we see that the DDMS architecture can give better performance than the existing Proxy Mobile IP (PMIP) protocol in terms of data transmission throughput.

I. 서 론

최근 스마트폰 사용이 증가하면서 고정 단말 위

주의 인터넷 환경이 이동 단말 위주의 인터넷 환경으로 변동되고 있다. 향후에는 이러한 추세가 더욱 심화될 것으로 전망됨에 따라 미래 인터넷 구조 설

※ 본 연구는 한국연구재단 기초연구사업(NRF-2011-0026529)과 2012학년도 경북대학교 학술연구비 및 방송통신위원회의 산업원천기술 개발사업(KCA-10913-05004)의 연구결과로 수행되었습니다.

• 주저자 : 경북대학교 컴퓨터학부 석사과정, hwkang0621@gmail.com, 준회원

° 교신저자 : 경북대학교 컴퓨터학부 교수, sjkoh@knu.ac.kr, 종신회원

* 경북대학교 컴퓨터학부 석사과정, peaceful7007@gmail.com

** 경북대학교 컴퓨터학부 박사과정, jiin16@gmail.com

*** 한국전자통신연구원 책임연구원, hyjung@etri.re.kr

논문번호 : KICS2012-08-353, 접수일자 : 2012년 8월 7일, 최종논문접수일자 : 2012년 10월 29일

계 시에 이동환경을 중요하게 고려할 필요가 있다.

현재 인터넷은 고정 환경 위주로 설계됨에 따라 이동환경을 지원하기 위해 Mobile IP (MIP) [1] 및 Proxy MIP (PMIP) [2] 등의 추가적인 프로토콜이 개발되었다. 그러나, 이러한 방식은 이동단말 위주의 미래 인터넷에 대한 근본적인 해결책을 제시하지는 못하고 있다.

상기 문제점을 해결하기 위해서 전 세계적으로 eMobility [3], MobilityFirst [4], AKARI [5] 등의 미래 인터넷 연구가 활발히 진행되고 있으며, 한국에서도 이동 환경 위주의 미래 인터넷 연구를 위해 Mobile Oriented Future Internet (MOFI) 구조 연구가 진행 중에 있다^{6,7,8)}.

MOFI 구조에서는 Identifier (ID)와 Locator (LOC)의 분리를 위해 Host ID and Local Locator (HILL) 원칙을 채택하고 있다. 즉, Host ID (HID)와 LOC 기능을 분리하여, 통신 기능은 HID를 토대로 수행하고, 라우터의 IP 주소를 LOC로 활용하여 네트워크에서의 패킷 전달 기능을 수행한다. 여기서 LOC는 로컬 IP 주소로서 각 네트워크에서만 유일하면 된다.

MOFI 구조에서는 HID-LOC 분리를 토대로 데이터 전송을 위한 LOC 조회(Query) 절차를 수행하여 최적화된 경로를 찾도록 하고, HID와 LOC 매핑 정보는 ‘규모성(scalability)’을 고려하여 분산형(distributed) 방식으로 관리한다. HID-LOC를 분리하여 사용하므로, 현재 IP 주소가 가지고 있는 “의미 중복(overloaded semantics)” 문제를 해결할 수 있고, Query 기능을 사용하여 데이터 전송에서 최적화된 경로를 사용할 수 있다. 또한, HID-LOC 매핑 정보를 분산형으로 관리하여, 중앙 서버에 제어 트래픽이 집중되는 문제를 해결할 수 있다.

본 논문에서는 HID-LOC 분리 구조에서 “해쉬(hash) 함수를 사용하는 Dynamic and Distributed Mapping System (DDMS) 기법”을 제안한다. 제안 기법의 성능 검증을 위해, 리눅스 플랫폼에서 제안 기법을 구현하고 테스트베드 구축을 통해 실험하였다.

본 논문은 다음과 같이 구성된다. 먼저 2절에서는 MOFI의 기본적인 구조 및 개요에 대해서 설명하고, 3절에서는 제안하는 해쉬 기반 분산형 HID-LOC 매핑 제어 기법에 대하여 기술한다. 4절에서는 리눅스 플랫폼에서의 제안 기법 구현 및 기존 기법과의 성능 비교분석을 기술한다. 마지막으로 5절에서 본 논문을 결론짓는다.

II. MOFI 구조 개요

본 논문에서 사용되는 용어를 표 1에 정리한다.

표 1. 약어 정리
Table 1. abbreviation

AR	Access Router
ADP	Access Delivery Protocol
BDP	Backbone Delivery Protocol
DDMS	Dynamic and Distributed Mapping System
HBR	HID Binding Request
HBA	HID Binding ACK
HID	Host Identifier
HILL	Host ID and Local LOC
HLR	HID-LOC Register
LMC	Local Mapping Controller
LOC	Locator
LQR	LOC Query Request
LQA	LOC Query ACK
MOFI	Mobile-Oriented Future Internet
QFDD	Query-First Data Delivery

그림 1은 MOFI의 3가지 기능 블록을 보여준다. 먼저, HILL 구조에서 각 단말은 고유한 HID를 가지며, IPv6 응용과의 호환성을 위해서 128 bit 형식을 취한다. LOC는 네트워크에서 호스트의 위치를 나타내고 데이터 패킷의 라우팅에 사용된다. LOC는 로컬(local) 속성을 지니며 글로벌 망에서 유일할 필요는 없다. 이는 글로벌 인터넷 망에서 라우팅 규모성 문제를 감소시키는 데에 유리하다. 통상적으로 LOC는 라우터 혹은 게이트웨이의 IP 주소를 사용한다. 단말이 이동하는 경우에 HID는 변하지 않고 LOC는 이동할 때마다 변경되며, HID와 LOC 간의 매핑 정보는 본 논문에서 제안하는 해쉬 기반 분산형 HID-LOC 매핑 제어 기법을 통해 관리된다.

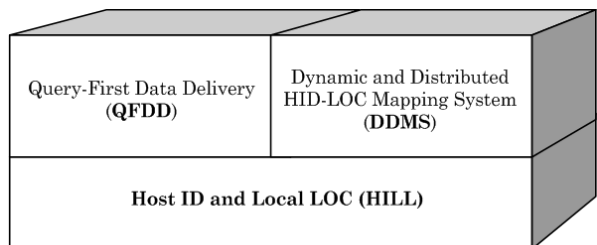


그림 1. MOFI의 3가지 기능 블록 [6]
Fig. 1. Main functional blocks of MOFI [6]

다음으로, QFDD 기능을 통해 송신 단말은 HID-LOC Query 절차를 통해 수신 단말의 위치 및 최적화 경로를 찾는다. 이를 위해 본 논문에서 제안하는 HID-LOC 관리 기법이 사용된다. 한편, 기존의 MIP 및 PMIP 프로토콜에서는 LOC에 대한 Query 기능 없이 송신 단말의 모든 데이터가 중앙 에이전트(HA/LMA)로 전달되어 비최적화된 경로를 사용하는 문제점이 있다.

림 2는 MOFI 구조에서의 HID 기반 종단간 통신 기능과 LOC 기반의 라우팅 절차를 보여준다.

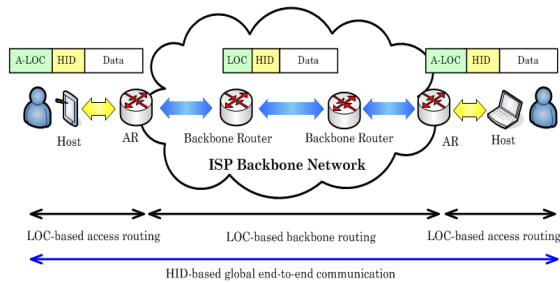


그림 2. HID 기반 통신과 LOC 기반의 Local 라우팅
Fig. 2. HID-based Communication and Local LOC-based Routing

그림에서 종단 호스트간 통신은 HID를 사용하고, LOC는 패킷 전송에 사용된다. 패킷 전송을 위한 LOC는 접속망에서의 A-LOC와 백본망의 LOC로 구분된다. 각 데이터 패킷 전송은 접속망과 백본망에서 독립적으로 이루어지고, Access Router(AR)에서 A-LOC와 LOC 사이의 변환 작업이 수행된다.

그림 3은 데이터 전송을 위한 프로토콜 모델을 보여준다. 통신 과정에서 HID 정보는 수송 계층과 네트워크 계층 사이에 MOFI shim 헤더 형태로 삽입된다. 패킷 전달은 Access Delivery Protocol (ADP)와 Backbone Delivery Protocol (BDP)을 통해 이루어지며, 현재 IP 프로토콜이 사용될 수 있다.

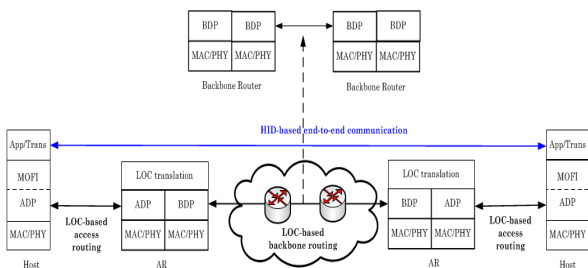


그림 3. 데이터 전송 프로토콜 모델
Fig. 3. Protocol model for data delivery

표 2는 지금까지 ID-LOC 분리 기법으로 제안된 Host Identity Protocol (HIP) [9], Locator Identifier Separation Protocol (LISP) [10], Identifier Locator Network Protocol (ILNP) [11] 프로토콜과 제안하는 MOFI 기법의 구조적인 특징을 비교하고 있다.

표 2. ID-LOC 분리 기법 비교
Table 2. ID-LOC Separation Technique compares

구분	MIP/PMIP	HIP	LISP	ILNP	MOFI
ID	HoA	HIT	EID	IPv6 address	HID
LOC	CoA	IP address	RLOC	IPv6 address	IP address
LOC Range	Global	Global	Global	Global	Local
Mapping Manger	HA, LMA	RVS	MS	D-DNS	LMC
Mapping Control	Centralized	Centralized	Centralized	Centralized	Distributed
Data delivery	Data-First	Query-First	Query-First	Query-First	Query-First

먼저, ID-LOC 관점에서 MIP/PMIP은 IP 주소를 HoA(ID) 및 CoA(LOC)로 사용하는 반면에, HIP는 ID로서 128비트의 Host Identity Tag(HIT)를 사용하고 LOC는 IP 주소를 사용한다. LISP의 경우, IP 주소를 Endpoint ID(EID) 및 Routing Locator(RLOC)로 사용한다. ILNP의 경우, IPv6 주소의 하위 64비트를 ID로 사용하고, 상위 64비트를 LOC로 사용한다. MOFI에서는 HIP와 유사하게 128비트의 HID를 사용하고 LOC로서 AR의 IP 주소를 사용한다. LOC 범위 관점에서, 기존 방식은 모두 글로벌(global) LOC를 사용하는 반면에, MOFI는 로컬(local) LOC를 사용하여, 라우팅 테이블 관리에서 규모성을 제공할 수 있다.

ID-LOC 매핑 관리자(agent) 및 매핑 제어 기법 측면에서, 기존 방식들은 HA/LMA, Rendezvous Server(RVS), Map Server (MS), Dynamic DNS (D-DNS) 등의 중앙집중형 서버를 사용하는 반면에, MOFI에서는 각 라우터에 Local Mapping Controller(LMC)를 사용하여 분산형으로 ID-LOC 매핑 정보를 관리한다.

데이터 전송 측면에서, MIP/PMIP 프로토콜은 데이터를 먼저(data-first) HA/LMA에 보내는 반면에, 다른 기법들은 먼저 매핑 관리자에 Query를 전송하여 상대방 노드의 LOC 정보를 파악한 다음 최적화된 경로로 데이터를 전송한다.

III. 분산형 HID-LOC 매핑 제어 기법

MOFI 구조에서 HID LOC 매핑 정보의 관리는 DDMS 기법을 통해 이루어지며, 본 논문에서는 도메인(ISP) 안에서 적용되는 “해쉬 기반 DDMS” 기법을 제안한다.

3.1. 분산형 매핑 제어 모델

분산형 HID-LOC 매핑 제어를 위해 각 AR은 Local Mapping Controller (LMC)를 가지며, 단말에 대한 HID-LOC 매핑 정보를 관리한다. 각 단말을 관리하는 LMC는 단말의 HID에 해쉬 함수를 적용하여 결정한다. 즉, HID를 토대로 해당 단말의 HID-LOC 정보를 관리하는 특정한 LMC가 결정된다. 그림 4는 해쉬 기반 분산형 매핑 제어 모델을 보여준다. 그림에서 각 AR별로 LMC가 위치하고 LMC간에 HID-LOC 매핑 관리를 수행한다.

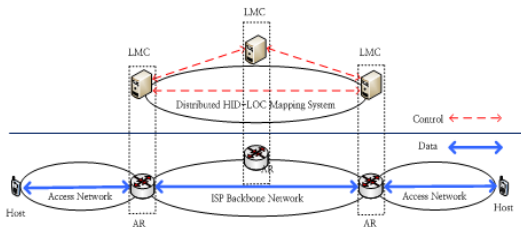


그림 4. 분산형 HID-LOC 매핑 제어 모델
Fig. 4. Distributed HID-LOC mapping control model

특정 단말에 대한 매핑 제어를 관리하는 LMC를 결정할 때, 도메인 내에서는 동일한 해쉬 함수를 사용한다고 가정한다. 단말이 망에 접속하는 경우, AR은 해당 단말의 HID에 해쉬 함수를 적용하여 어느 LMC가 단말의 HID-LOC 매핑 정보를 관리할 지 파악한다. 해쉬 함수를 통해서 결정된 LMC는 해당 단말의 HID-LOC 매핑 정보를 자신의 HID-LOC Register (HLR)에 기록 및 갱신한다. 이러한 방법으로 HID-LOC 매핑 정보는 각 AR의 LMC에서 분산되어 관리된다.

3.2. HID-LOC Binding 절차

그림 5는 HID-LOC Binding 절차를 보여준다.

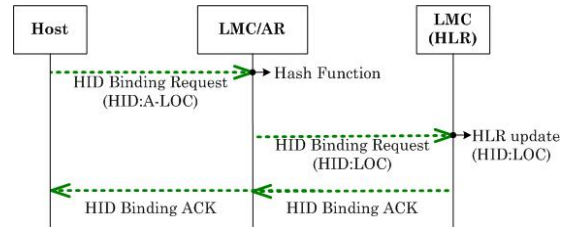


그림 5. 해쉬 기반 HID-LOC Binding 절차
Fig. 5. Hash-based HID-LOC Binding Operation

단말이 AR에 접속하는 경우, 단말은 AR에 위치한 LMC에 HID Binding Request (HBR) 메시지를 전송하고, 이어서 LMC/AR은 단말의 HID에 해쉬 함수를 적용하여 매핑 정보를 관리할 LMC를 결정한 후 HBR 메시지를 전달한다. HBR을 수신한 LMC는 HID-LOC 정보로 HLR을 갱신하고(LOC는 LMC/AR의 IP 주소를 의미함), HID Binding ACK (HBA) 메시지로 응답한다.

3.3. 데이터 전송을 위한 LOC Query 절차

그림 6은 데이터 전송을 위한 LOC Query 절차를 보여준다. 그림에서 송신단말(SH: Sending Host)과 수신단말(RH: Receiving Host)은 각각 LMC/AR-1과 LMC/AR-3에 접속한 상태이고, RH의 HID-LOC 정보는 해쉬 함수를 통해 LMC-2에 등록된 것으로 가정한다.

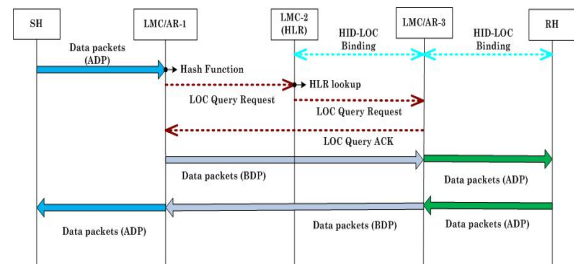


그림 6. 데이터 전송을 위한 LOC Query 절차
Fig. 6. LOC Query Operation for Date Delivery

SH의 데이터가 LMC/AR-1에 도착하면, LMC-1은 해쉬 함수를 통해 파악된 LMC-2에 LOC Query Request(LQR) 메시지를 전송한다. LMC-2는 HLR 검색을 통해 RH의 위치를 파악하고 LMC/AR-3에 LQR 메시지를 전달한다. LMC-3는 SH가 접속해 있는 LMC/AR-1에 LOC Query ACK(LQA) 응답 메시지를 전송한다. 그리고 나서, AR-1은 최적화된 경로로 AR-3 및 RH에게 데이터를 전송한다. 이후 데이터 전송은 추가적인 LOC Query 절차 없이 SH와 RH간에 최적화 경로를 이용하여 수행된다.

IV. 구현 및 성능 평가

제안된 기법의 성능 분석 및 검증을 위해 단말과 LMC/AR 기능을 리눅스 플랫폼에 구현하였다.

4.1. 구현 세부 사항

4.1.1. 호스트(단말)

먼저, 단말의 구현을 위해 Ubuntu 10.04 버전을 사용하였고, 리눅스 커널 버전은 2.6.32.16 버전을 사용하였다. 또한 6-to-4 터널링 방법을 사용하여서 HID를 구현하였다[12]. 6-to-4 터널링에서 IPv6 주소를 HID로 사용하였고, IPv4 주소를 LOC 주소로 사용하였다. 128-bit의 HID는 그림 7과 같이, 6-to-4 터널링을 위해서 2-byte prefix(2002::)와 4-byte의 Domain ID, 그리고 10-byte의 사용자 ID로 구성된다.



그림 7. HID 구성
Fig. 7. HID format

4.1.2 AR

데이터 전송을 위한 AR구현을 위해 호스트와 같이, Ubuntu 10.04 버전과 리눅스 커널 2.6.32.16 버전을 사용하였다. 그리고, 데이터 패킷이 호스트에서 AR에 도착하였을 때, AR은 ADP 헤더(header)를 BDP 헤더로 변환해야 하는데, 이러한 헤더 변환을 구현하기 위해서 netfilter[13]와 iptables[14]를 사용하였다.

그림 8은 AR간의 데이터 전송 프로토콜 모델을 보여준다. 각 AR은 LOC와 BDP를 이용하여 AR간에 패킷 전송을 수행한다. 각 AR은 상대방 AR에 데이터를 전송하기 이전에 LOC Query 절차를 수행하여 상대방 단말의 LOC를 알아내고, 패킷 캡슐화를 통해 데이터 패킷을 전송한다. 이를 “Query-and-Encapsulation” 절차라 한다.

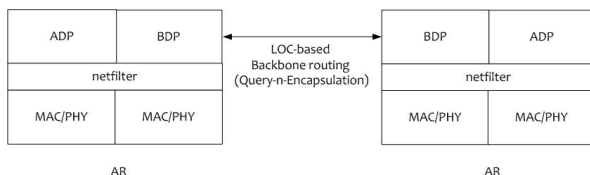


그림 8. AR에서 데이터 전송 프로토콜 모델
Fig. 8. Protocol model at AR

그림 9는 AR에서 netfilter를 통한 헤더 변환절차에 사용되는 함수들을 보여준다.

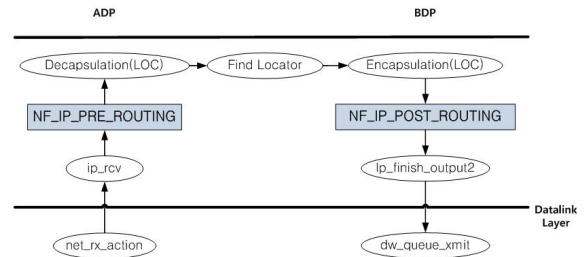


그림 9. AR에서 사용되는 netfilter 모듈
Fig. 9. netfilter module modified for AR

패킷이 AR에 도착했을 때, ip_rcv 함수가 네트워크 계층에서 수행된다. 그리고 netfilter의 NF_IP_PRE_ROUTING 함수를 사용하여 데이터 패킷을 hooking한다. hooking 후에 데이터의 헤더는 ADP에서 BDP로(또는 BDP에서 ADP로) 변환된다. BDP 헤더를 구성하기 위해서 해당 AR의 LMC는 LOC Query를 수행하여 이동 단말의 LOC 정보를 파악한다. 프로토콜 변환이 완료되면, NF_IP_POST_ROUTING 함수를 통해 데이터 패킷을 ip_finish_output 함수로 전달한다. 그리고 iptables은 데이터 패킷을 단말에서 AR로(또는 AR에서 단말로) 전달하기 위해 사용된다.

4.1.3 LMC

LMC는 AR에서 구현되었으며, 분산형 매핑 제어를 위한 해쉬 함수로 modulo(%) 연산자를 사용하였다. 즉, HID에 대하여 “HID % 도메인에 있는 LMC의 수” 계산을 통해서 해당 호스트의 매핑 정보를 관리하는 LMC를 결정한다.

단말과 LMC/AR 혹은 LMC/AR간의 제어 메시지는 모두 UDP를 사용하여 구현하였다.

4.2 성능 비교 분석

4.2.1 실험망 구성

그림 10은 성능실험을 위한 실험망을 보여준다. 실험망은 2개의 호스트(SH, RH)와 3개의 LMC/AR로 구성되었다. 실험망에서 SH는 패킷을 보내는 역할을 하고, RH는 패킷을 받는 역할을 한다.

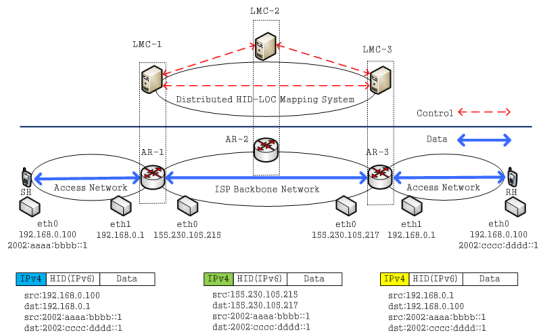


그림 10. 성능 분석을 위한 실험망 구성도
Fig. 10. Test-bed Network configuration for performance analysis

SH의 HID로 2002:aaaa:bbbb::1를 사용했고, RH의 HID로 2002:cccc:dddd::1을 사용했다. 그리고 A-LOC 주소로 사설 IPv4 주소(192.168.x.y)를 사용했고, LOC 주소로는 공중 IPv4 주소(155.230.x.y)를 사용하였다.

4.2.2. 데이터 전송 실험

데이터 전송에 대한 실험은 다음과 같이 수행하였다. 첫 번째로, SH와 RH를 AR-1과 AR-2에 연결을 하여 SH에서 RH로 데이터 패킷을 전송하였다. 이 실험에서 LOC Query 과정은 이미 수행되었다고 가정하여 실험을 하였다. 각 실험에 대하여 Wireshark[15]를 이용하여 교환되는 패킷을 캡처하였다.

그림 11는 AR-1에서 AR-2로의 패킷 전송 결과를 캡처한 결과를 보여주고, 그림 12는 AR-2에서 RH로 데이터가 전송된 패킷을 캡처한 결과를 보여준다. 각 그림에서, 전송된 경로의 LOC 정보를 확인할 수 있다. 그림 11에서 AR-1과 AR-2는 155.230.105.215와 155.230.105.217의 LOC를 사용하고 있다. 그림 12에서 RH의 A-LOC로 192.168.0.100가 쓰이고 있다. 아울러, SH와 RH의 HID가 데이터가 전송되는 동안 변하지 않음을 확인할 수 있다.

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 (08:00:0c:03:64:c0)
 LOC, Src: 155.230.105.215 (155.230.105.215), Dst: 155.230.105.217 (155.230.105.217)
 HID, Src: 2002:aaaa:bbbb::1 (2002:aaaa:bbbb::1), Dst: 2002:cccc:dddd::1 (2002:cccc:dddd::1)
 User Datagram Protocol, Src Port: 41906 (41906), Dst Port: avt-profile-1 (5004)
 Data (1400 bytes)

그림 11. AR-1에서 AR-2로 전송되는 패킷
Fig. 11. Packet Capture from AR-1 to AR-2

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: (08:00:0c:00:b7:)
 LOC, Src: 155.230.105.215 (155.230.105.215), Dst: 192.168.0.100 (192.168.0.100)
 HID, Src: 2002:aaaa:bbbb::1 (2002:aaaa:bbbb::1), Dst: 2002:cccc:dddd::1 (2002:cccc:dddd::1)
 User Datagram Protocol, Src Port: 42713 (42713), Dst Port: avt-profile-1 (5004)
 Data (429 bytes)

그림 12. AR-2에서 RH로 전송되는 패킷
Fig. 12. Packet Captuer from AR-2 to RH

4.2.3 분산형 매핑 제어 기법의 성능 분석

DDMS 매핑 기법의 성능 분석을 위해 3가지의 다른 시나리오를 적용하였다. 각 경우에서 HID와 LOC의 매핑 정보를 제어하는 LMC는 해쉬 함수를 사용하여 결정된다.

첫 번째 경우는, 그림 13과 같이 SH는 AR-1에 연결되어 있고, RH는 AR-3에 연결되어 있다. 또한 RH의 HID가 LMC-1에 등록되어 있다. 데이터가 AR-1에 도착했을 때, LMC-1에서 RH의 HID-LOC 매핑 정보를 이미 LMC-1이 관리하기 때문에, LOC Query가 수행될 필요가 없다. 따라서 데이터 패킷은 바로 AR-1에서 AR-3로 전송된다.

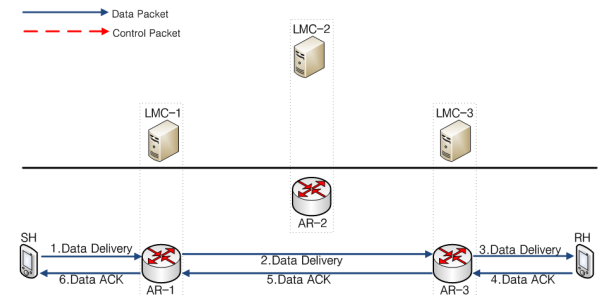


그림 13. Case 1 - LOC Query가 필요 없는 경우
Fig. 13. Case 1 - No need of LOC Query

두 번째 경우는 그림 14와 같이 SH와 RH는 AR-1과 AR-3에 연결되어 있지만, RH의 HID가 LMC-2에 등록되어 있는 경우이다. 따라서 데이터 패킷이 AR-1에 도착했을 때, LMC-1이 RH의 HID-LOC 매핑 정보를 관리하고 있지 않기 때문에, LMC-1은 HID-LOC 정보를 관리하는 LMC-2에게 LQR 메시지를 보내게 된다. 그리고 LMC-2는 RH가 연결되어 있는 LMC-3에게 LQA 메시지를 전달하고, LMC-3가 LMC-1에게 LQA 메시지를 보내게 된다. 그 후에 AR-1이 AR-3에게 데이터를 전송할 수 있다.

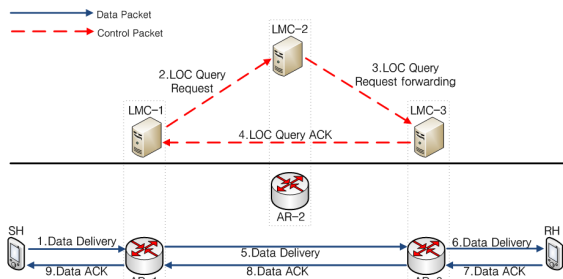


그림 14. Case 2 - LOC Query가 LMC 2로 수행되는 경우
Fig. 14. Case 2 - LOC Query to LMC-2 and to LMC-3

세 번째 경우는 그림 15과 같이, SH와 RH는 AR-1과 AR-3에 연결되어 있고, RH의 HID가 LMC-3에 등록되어 있는 경우이다. 데이터 패킷이 AR-1에 도착하면, LMC-1은 LMC-3에게 LQR 메시지를 보내게 되고, LMC-3는 응답으로 LQA 메시지를 LMC-1에게 보내게 된다. 그 후에 AR-1이 데이터를 AR-3로 보내게 된다.

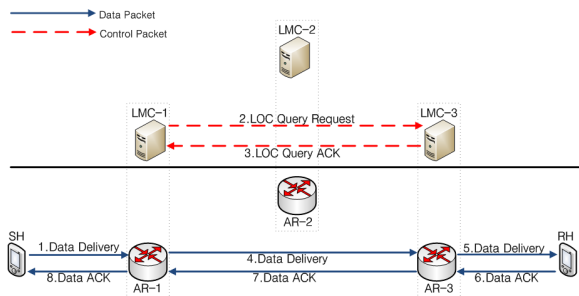


그림 15. Case 3 - LOC Query가 LMC-3로 수행되는 경우
Fig. 15. Case 3 - LOC Query to LOM-3

DDMS와의 성능 비교를 위해서 PMIP[2]에서의 데이터 전송 실험을 수행하였다. PMIP 전송에서는 LMC-2가 PMIP의 LMA 기능을 수행하고, AR-1의 SH에서 AR-3의 RH에게 보낸 모든 패킷은 LMA를 거쳐서 전송이 되며, LOC Query 과정은 수행되지 않는다.

SH에서 RH로 10개의 데이터 패킷을 보내는 실험을 위 3가지 경우의 DDMS와 PMIP에 적용시켜 보았다. 그리고 LOC Query를 포함한 데이터 전송 시간을 측정해 보았다.

그림 16은 DDMS의 3가지 경우와 PMIP의 데이터 전송 시간을 비교하고 있다. 그림에서 보듯이 PMIP는 DDMS의 3가지 경우보다 더 좋지 않은 성능을 보였다. 이러한 이유는 PMIP에서는 최적화 경로를 사용하지 않기 때문이다. DDMS의 3가지 경우를 서로 비교하면, Case 1이 가장 좋은 성능을 보였다. 이는 Case 1은 LOC Query가 수행되지 않

기 때문이다. 한편, Case 2는 Case 1이나 Case 3보다 좋지 못한 성능을 보였는데, 그 이유는 LOC Query 메시지를 LMC-2를 거쳐서 보내야 하기 때문이다.

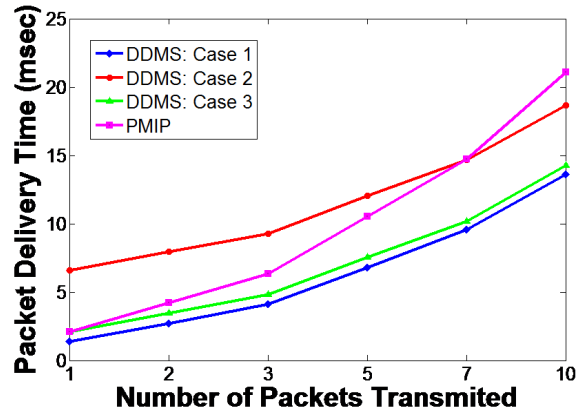


그림 16. 제안 기법과 PMIP와의 성능 비교
Fig. 16. Comparison of DDMS and PMIP

그림 17는 단말의 수를 증가시키면서 DDMS와 PMIP의 데이터 전송 속도를 비교한 것이다. DDMS는 3가지 경우 중에서 Case 2를 선택하여 실험을 수행하였는데, 그 이유는 Case 2가 제안 기법 중 가장 나쁜 성능을 보였기 때문이다. 그럼에도 불구하고 DDMS Case 2가 PMIP보다 더 좋은 성능을 보이고 있음을 확인할 수 있다. 특히, 단말의 수가 늘어날수록 PMIP와 DDMS의 성능의 차이가 훨씬 더 커짐을 알 수 있다.

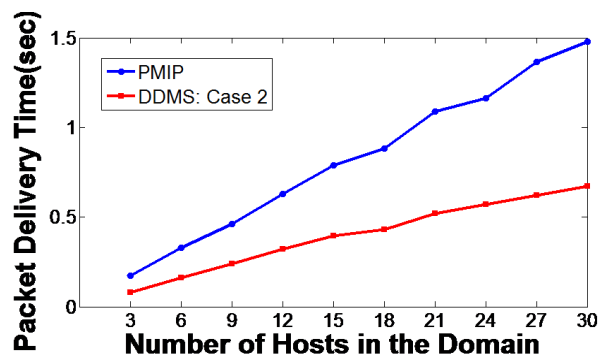


그림 17. 제안 기법(Case2)와 PMIP와의 성능 비교
Fig. 17. Comparison of DDMS (Case2) and PMIP

V. 결론

본 논문에서는 이동환경 중심의 인터넷을 위한 MOFI 구조를 소개하고, MOFI의 ID-LOC 분리 구

조에 적합한 해쉬 기반의 분산형 HID LOC 매핑 제어 기법을 제안하였다. 제안하는 매핑 제어 기법에서는 기존의 중앙집중형 방식과는 달리, HID-LOC 매핑 기능을 각 도메인의 AR에 분산하여 수행함으로써, 기존 PMIP의 중앙집중식 방식에 비해 규모성 및 전송효율 측면에서 장점을 제공한다.

한편, 제안 기법의 성능 분석 및 검증을 위해 MOFI 구조의 성능을 검증하기 위해, 리눅스 플랫폼에서 제안하는 매핑 기법을 구현하고 테스트베드 실험을 통해 기존 PMIP 기법과 비교하였다. 테스트베드 실험 결과, 제안하는 분산형 매핑 기법이 PMIP 등의 기존 중앙집중형 방식에 비해 데이터 전송 측면에서 더 나은 성능을 보임을 확인하였다.

References

[1] D. Johnson, et al., *Mobility Support in IPv6*, IETF RFC 3775, 2004

[2] S. Gundavelli, *Proxy Mobility IPv6*, IETF RFC 5213, 2008

[3] W. Mohr, eMobility Project, Retrieved October 30, 2012, from www.emobility.eu.org

[4] D. Raychaudhuri, *Mobility First: Future Internet Architecture*, Retrieved October 30, 2012, from mobilityfirst.winlab.rutgers.edu

[5] H. Harai, AKARI Project, Retrieved October 30, 2012, from akari-project.nict.go.jp

[6] H. Jung, *Mobile Oriented Future Internet*, Retrieved October 30, 2012, from <http://www.mofi.re.kr>

[7] Y. H. Han, et al., "Overview of Mobile Network Architecture and Research in Future Internet", *Korean Institute of Communications and Information Sciences*, 29(3), pp. 41 ~ 48, 2012

[8] J. W. Park, et al., "Performance Enhancement of Proxy Mobile IPv6 using Binding Query", *Journal of Korean Institute of Communications and Information Sciences*, 36(11), pp. 1269 ~ 1276, 2011

[9] R. Moskowitz, et al., *Host Identity Protocol*, IETF RFC 5201, Apr. 2008

[10] D. Farinacci, et al., *Locator/ID Separation Protocol*, IETF Internet-Draft, draft-ietf-lisp-23,

May 2012

[11] R. Atkinson and S. Bhatti, *Identifier-Locator Network Protocol*, IETF Internet-Draft, draft-irtf-rrg-ilnp-arch-06, July 2012

[12] Wikipedia, 6to4 tunneling, Retrieved October 30, 2012, from en.wikipedia.org/wiki/6to4

[13] P. Ayuso, Netfilter HOWTO, Retrieved October 30, 2012, from netfilter.org/

[14] Ubuntu, IPtables HOWTO, Retrieved October 30, 2012, from help.ubuntu.com/community/IptablesHowTo

[15] Wireshark, Wireshark Homepage, Retrieved October 30, 2012, from wireshark.org

강 형 우 (Hyung-Woo Kang)



2011년 2월 경북대학교
컴퓨터학부 공학사
2011년 3월~현재 경북대학교
컴퓨터학부 석사과정
<관심분야> 미래인터넷,
이동성 관리

최 낙 중 (Nak-Jung Choi)



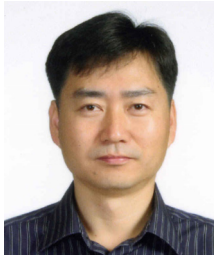
2011년 2월 경북대학교
컴퓨터학부 공학사
2011년 3월~현재 경북대학교
컴퓨터학부 석사과정
<관심분야> 미래인터넷,
이동성 관리

김 지 인 (Ji-In Kim)



2008년 2월 경북대학교
컴퓨터학부 공학사
2010년 2월 경북대학교
컴퓨터학부 공학석사
2011년 3월~현재 경북대학교
컴퓨터학부 박사과정
<관심분야> 미래인터넷

정 희 영 (Heeyoung Jung)



2004 충남대학교
정보통신공학 공학박사
1991~현재 ETRI 책임연구원
<관심분야> 미래인터넷,
이동성 관리

고 석 주 (Seok-Joo Koh)



1992년 KAIST 공학사
1994년 KAIST 공학석사
1998년 KAIST 공학박사
1998년 9월~2004년 3월
ETRI 표준연구센터
2004년 3월~현재 경북대학교
컴퓨터학부 교수

<관심분야> 미래인터넷, IP Mobility, 멀티캐스트