

객체지향개념을 도입한 Mobile IP의 설계 및 구현(2)

손동우*, 채동현, 한규호, 마영식, 안순진

고려대학교 전자공학과 컴퓨터네트워크 연구실

{bunny, hsunhwa, garget, mys, sunshin}@dsys.korea.ac.kr

A Design and Implementation of Mobile IP Architecture using Object-Oriented Concept(2)

Dongwoo Sohn*, Donghyun Chae, Gyuho Han, Youngsik Ma, Sunshin An

Computer Network Lab., Dept. Of Electronic Eng., Korea University

요약

Mobile IP는 OSI 7계층 중 네트워크 계층에서의 이동성 지원을 위한 Protocol이다. Mobile IP의 세 주체 중 Home Agent와 Foreign Agent는 Mobile Node에게 이런 이동성을 제공하는 서비스 제공자로서의 역할을 수행한다.

본 논문에서는 Mobile IP Protocol의 주체 중 Mobile Agent를 객체지향 개념을 도입하여 설계·구현한다. Linux Kernel 2.2.14의 환경에서 객체지향 언어인 C++을 이용, 기능적인 측면에서의 객체를 구성하고, IP 계층에서의 이동성을 제공하기 위해 Home Agent와 Foreign Agent의 주된 기능인 터널링(Tunneling)을 Linux Kernel의 모듈(Module)을 이용, 설계·구현한다.

1. 서론

무선 인터넷은 통신 서비스의 사용자가 이동하는 환경과 고정되어 있는 환경의 경계를 무너뜨리는 방향으로 발전하고 있다. 다시 말해 사용자가 이동하는 환경에서도 고정된 환경과 똑같은 서비스를 제공하기 위한 기술 개발 연구가 차세대 통신망의 방향을 이끌어 가고 있다. 이와 더불어 차세대 통신망이 Packet Data 통신을 기반으로 하는 All IP 망으로의 통일을 지향하는 것으로 요구되어 지고 있는 상황에서 분명 네트워크 계층에서의 이런 이동성 지원을 목적으로 하는 Mobile IP Protocol[1]은 차세대 인터넷의 핵심기술임에 틀림이 없다.

이에 본 논문에서는 IETF의 Mobile IP Protocol[1]을 설계 및 구현을 그 목적으로, 논문 1에의 Mobile Node 중심의 설계 및 구현에 이어 두 가지 Mobile Agent인 Home Agent와 Foreign Agent의 객체지향 개념의 설계 및 구현을 IP Tunneling의 기능 구현을 중심으로 전개하겠다. 구현은 Linux Kernel 2.2.14의 환경에서 C++ 언어를 통해 이루어졌다.

우선 2장에서는 관련 연구에 대해 고찰하고, 3장에서는 Mobile Agent를 기능적인 객체들로 구성·설계하고, IP Tunneling 구현에 대해 언급한 후, 4장에서 테스트 베드 구축과 구현의 실행 결과를 제시하고, 마지막으로 5장에서는 이에 대한 결론과 향후 연구방향에 대해 논하겠다.

2. 관련연구

이미 Mobile IP Protocol은 여러 연구 단체에 의해 설계·구현되어 왔다. Stanford University의 MosquitoNet Mobile IP[2]는 현재 Linux kernel 2.2.x에서 동작하는 2.0.2beta version까지 개발되었다. MosquitoNet은 [1]에서보다 Foreign Agent의 기능이 취약한 대신에 co-located 상태를 보강한 것이 특징이다. National University of Singapore의 Mobile IP[3]는 kernel 레벨에서의 구현으로서 현재 Linux 환경에서 동작하는 Mobile IPv4와 Mobile IPv6 등의 Linux Version과 Win95 환경에서 동작하는 Window Version이 있다.

3. Objects Design

3.1 Design of Mobile Agents

기능적인 면에서의 객체 설계를 기본으로 Mobile Agent는 크게 고유의 기능을 수행하는 세가지 객체들로 구성된다. 이는 우선 Agent Advertisement Management 기능을 수행하는 Agent Advertisement Manager, Mobile Node Registration Management를 담당하는 Registration Manager, 마지막으로 Tunneling을 맡고 있는 Tunnel Maker로 열거된다. 이에 Mobile Agent가 저장하고 변경할 데이터들의 집합을 각각 테이블로 구성, 그림 1과 같이 하나의 Mobile Agent를 형성하게 된다.

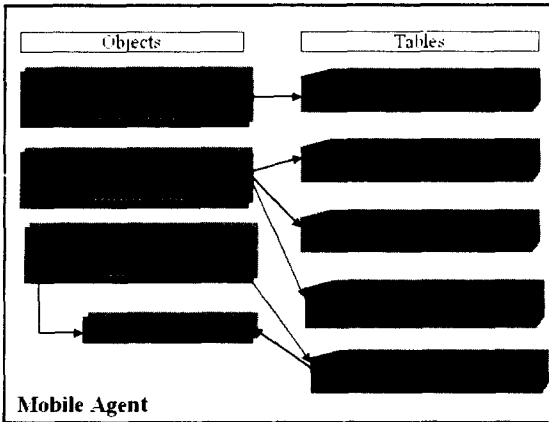


그림 1. Mobile Agents System Architecture

3.2 Agent Advertisement Manager

주기적인 ICMP Agent Advertisement Making/Broadcasting 기능과 ICMP Agent Solicitation Receiving/Processing 기능을 담당하는 객체로 그림 2 의 (a) 부분에 해당된다. 이는 Mobile Agent 가 지원하는 모든 인터페이스 (Interface)에 대해 동시에 수행되며, ICMP Agent Solicitation 패킷의 분석 후 주어진 Mobile Node 로 만의 Unicast 에 의한 ICMP Agent Advertisement Sending 이 아니라 Broadcast 에 의한 Sending 기능이 수행됨이 특징이다.

3.3 Registration Manager

Mobile Node 로부터의 Registration Request Receiving/Processing 기능과 Registration Reply Making/Sending 기능을 담당하며 그림 2 의 (b)에 해당된다. Registration Request 패킷에 따라 Mobile Node 를 Home Agent 에 경우 Binding-Table 에 추가하고, Foreign Agent 에 경우 Visitor-Table 에 추가한다. 또한 Mobile Node 가 등록될 때마다 Tunnel Maker 를 생성시켜 Tunneling 기능을 관장하게 된다.

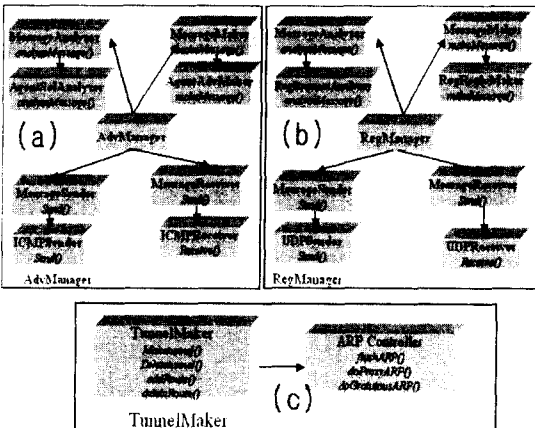


그림 2. Object Specifications in Mobile Agents

```

Home Agent
[bunny@purple bunny]$
[bunny@purple bunny]$
[bunny@purple bunny]$ /sbin/ip tunnel show
tunl0: ip/ip remote any local any ttl inherit nopmtudisc
gre0: gre/ip remote any local any ttl inherit nopmtudisc
tunl1: ip/ip remote 163.152.27.62 local 163.152.33.127 ttl 100
[bunny@purple bunny]$
[bunny@purple bunny]$ /sbin/ip route show
163.152.33.127 dev eth0 scope link
163.152.33.126 via 163.152.27.62 dev tunl1 onlink
163.152.33.0/24 dev eth0 proto kernel scope link src 163.152.33.127
127.0.0.0/8 dev lo scope link
default via 163.152.33.1 dev eth0
[bunny@purple bunny]$ /sbin/arp -a
? (163.152.33.1) at 00:20:DA:86:45:73 [ether] on eth0
? (163.152.33.126) at * PERM PUP on eth0
[bunny@purple bunny]$
    
```

그림 3. Virtual Interface(top), Routing Table(middle) and ARP cache(bottom) Statuses in Home Agent(163.152.33.127) when Mobile Node(163.152.33.126) is at Foreign Link(163.152.27)

Foreign Agent 의 경우는 Mobile Node 로부터 온 Registration Request 패킷에 대한 분석뿐만 아니라, Home Agents 로부터 온 Registration Reply 패킷에 대한 분석 또한 필요하다.

3.4 Tunnel Maker

Registration Manager 에 의해 생성되어 주어진 Mobile Node 에 대한 가상 인터페이스(Virtual Interface) 추가/삭제와 라우팅 테이블(Routing Table)의 추가/삭제를 담당하고 ARP Controller 객체를 생성, Proxy ARP 와 Gratuitous ARP 도 관장한다.

4. Tunneling Implementation

터널링(Tunneling)은 IP Encapsulation/Decapsulation 기능을 이용, 터널(tunnel)의 start point 에 도착한 일정 노드에 향하는 패킷을 터널의 end point 로 라우팅(Routing)하는 것을 의미한다. 이는 IP 패킷을 포워딩(forwarding)할 수 있음은 물론 가상 인터페이스를 생성할 수 있어야 실행 가능하다. Linux kernel 2.2.14 에서 제공하는 IP Forwarding 옵션과 IP Tunnel 모듈을 통해 이를 실현하였다.

```

Foreign Agent
[bunny@netdragon bunny]$
[bunny@netdragon bunny]$
[bunny@netdragon bunny]$ /sbin/ip tunnel show
tunl0: ip/ip remote any local any ttl inherit nopmtudisc
gre0: gre/ip remote any local any ttl inherit nopmtudisc
tunl1: ip/ip remote 163.152.33.127 local 163.152.27.62 ttl 100
[bunny@netdragon bunny]$
[bunny@netdragon bunny]$ /sbin/ip route show
163.152.33.126 dev eth0 scope link
163.152.27.62 dev eth0 scope link
163.152.27.0/24 dev eth0 proto kernel scope link src 163.152.27.62
127.0.0.0/8 dev lo scope link
default via 163.152.27.1 dev eth0
[bunny@netdragon bunny]$
    
```

그림 4. Virtual Interface(top) and Routing Table(bottom) Statuses in Foreign Agent (163.152.27.69) when Mobile Node(163.152.33.126) is at Foreign Link(163.152.27)

곧이어 단순한 가상 인터페이스 생성뿐만 아니라, 적절한 라우트(Route) 항목의 추가, ARP 기능의 설정 등이 순차적으로 실행된다. Home Agent 에서는 터널링 될 패킷을 터널의 end point 로의 라우팅을 위하여 Mobile Node 로 향하는 패킷에 대한 라우트 항목을 가상 인터페이스를 통하도록 라우팅 테이블에 추가하고, Mobile Node 로 향하는 패킷을 가로채기 위해서 Proxy ARP 에 대한 기능 또한 수행한다.

마찬가지로 Foreign Agent 에서도 Home Agent 와 대칭적인 가상 인터페이스를 생성하고, Mobile Node 가 직접 Foreign Agent 의 링크에 연결되어 있으므로 Mobile Node 로 향하는 라우트에 관해 직접 연결된 형태의 항목으로 라우팅 테이블에 추가하였다. 그림 3 은 Mobile Node(163.152.33.126)가 Foreign 링크 (163.152.27.)에 있을 때, Home Agent(163.152.33.127)에서 생성된 라우팅 테이블과 가상 인터페이스, 그리고 Proxy ARP cache 의 상황, 그림 4 는 Foreign Agent (163.152.27.69)에서 생성된 라우팅 테이블과 가상 인터페이스의 상황을 나타낸다.

반대로 Mobile Node 가 Home 링크로 되돌아올 경우에는 Home/Foreign Agent 에서 생성된 가상 인터페이스가 삭제되고, Mobile Node 에 대한 라우트 항목도 라우팅 테이블에서 제거된다. 이에 Home Agent 는 이웃 노드들의 ARP cache 를 갱신하도록 Gratuitous ARP 를 통해 Proxy ARP 기능의 종료와 동시에 ARP 기능을 제자리로 되돌리는 기능을 수행한다.

5. Implementation Result

Test Bed 를 구축하기 위하여 별개의 링크를 사용하는 두개의 망(163.152.33, 163.152.27)에 각각 Mobile Agent 인 Home Agent(163.152.33.127)과 Foreign Agent (163.152.27.69)를 설치하였다. Mobile Node (163.152.33.126)에는 WaveLan 카드를, 각각의 링크에는 Access Point 를 장착, 무선 환경을 지원하는 형태로 Test Bed 를 구축하였다. 그림 5 는 실행되고 있는 Mobile Agent 의 상황을 모니터링한 모습이고, 그림 6 은 Mobile IP Protocol 이 실행되어 어느 한 노드(Comme3.korea.ac.kr)에서 Traceroute 프로그램을 통해 Foreign 링크에 있는 Mobile Node 로 향하는 경로를 보여주는 것으로 터널링되는 상황을 잘 나타내 주고 있다. Mobile Node 를 찾기 위해 우선 Mobile Node 의 홈 네트워크 링크(163.152.33)로 라우트 한 다음 Home Agent(163.152.33.127)가 Mobile Node 로 향하는 패킷을 가로챘 후 이를 Foreign Agent (163.152.27.69)로 터널링을 수행하는 모습이다.

6. 결론 및 향후 연구과제

지금까지 Mobile IPv4 의 Mobile Agent 를 객체지향적으로 설계 및 구현하고, IP 계층에서의 이동성 지원을 IP Tunneling 의 구현과 실행 결과로 증명하였다. 보다 더 적절한 핸드오프(Handoff)를 지원하고, 잦은 이동성을 부여하기 위해서는 단순한 IP 계층만의 이동성지원이 아니라 데이터 링크 계층에서의 이동성 지원 기술과 연동되어야 할 것이다. 또한 네트워크의 효율적인 사용과 IP 기능의 확장성을 위해서는 반드시 Mobile IPv6 으로의 전환이 필요하다.

7. 참고 문헌

[1] C. Perkins, IP Mobility support, Internet RFC2002, October 1996

[2] M.G Baker, X.H.Zhao, S.Cheshire, J.Stone, Supporting mobility in MosquitoNet, Proceeding of the 1996 USENIX, San Diego, CA, January 1996, pp.1 - 13
 [3] K.C.Chua, Y.Z.Li, C.C.Foo, On a Linux implementation of mobile IP and its effect on TCP performance

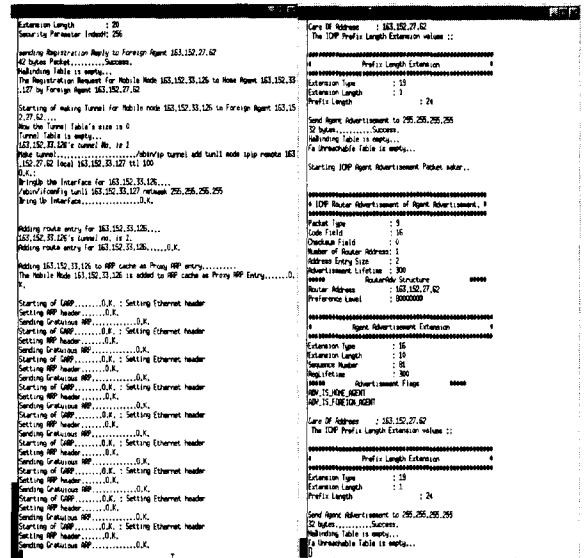


그림 5. One Shot of Monitoring Mobile Agents(Home Agents(left), Foreign Agent(right))

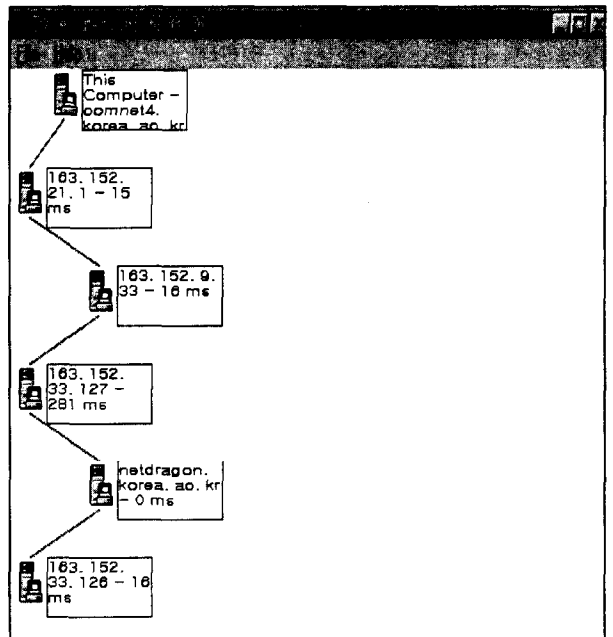


그림 6. Running Traceroute for Mobile Node when Mobile Node(163.52.33.126) is at Foreign Link(163.152.27.)