

논문 2007-44TC-12-6

다중 인터페이스 MIPv6 환경에서의 Fast Handover 방안 및 성능 분석

(Fast Handover Mechanism for Multi-Interface MIPv6 Environments
and Performance Evaluation)

박 만 규*, 황 안 규*, 이 재 용**, 김 병 철**

(Man Kyu Park, An Kyu Hwang, Jae Yong Lee, and Byung Chul Kim)

요 약

최근에 이동 단말들의 급격한 증가와 함께 이동 노드에 다양한 무선 접속 기술들의 사용이 가능해졌다. 또한 IPv6 기술이 네트워크에 도입되면 하나의 단말이나 인터페이스에 여러 개의 공중 IP주소를 가지는 멀티호밍(multi-homing) 단말이 일반화 될 것이다. 이에 맞추어 다중 인터페이스 멀티호밍 단말에 대한 이동성 관리기술 연구가 무선 인터넷 분야에서 활발히 진행 중이다. 본 논문에서는 다중 인터페이스를 지원하는 FMIPv6의 FBU 메시지를 대신하여, "tunnel destination" 이동성 옵션과 이 옵션을 표시하는 "T" flag를 이용하는 MFBU 메시지를 새로 정의하여, NAR이 아닌 특정 터널링 목적지로 핸드오버 동안 패킷을 터널링 시키는 다중 인터페이스 fast handover Mobile IPv6 절차의 확장을 제안한다. 이는 기존의 FMIPv6를 이용하여 핸드오버절차를 수행하는 동안 TCP 플로우의 패킷 도착순서가 뒤바뀌어 세 개의 중복 ACK에 의한 불필요한 혼잡 제어로 성능이 저하 되는 것을 완화하여, 핸드오버 성능을 향상 시키게 된다. 본 논문에서는 그 성능을 검증하기 위해 기존의 단일 인터페이스 MIPv6 NS-2 시뮬레이터를 확장하여 다중 인터페이스 FMIPv6 시뮬레이터를 구현하였으며, TCP 트래픽을 이용한 시뮬레이션을 통해 핸드오버 성능 향상을 확인하였다.

Abstract

Recently, in addition to the sharp increase of mobile nodes, various kinds of wireless technologies are available for mobile nodes. If IPv6 technology is applied to the network, multi-homing terminals which have several public IP addresses on one interface will be common. Accordingly, there are many research activities on mobility management for multi-interface, multi-homing nodes. In this paper we propose an extended fast handover mechanism for multi-interface MIPv6 environments that uses multi-interface FBU (MFBU) message instead of the existing FBU message. The MFBU message has the "tunnel destination" mobility option that points a specific tunnel destination other than NAR, and "T" flag that indicates the existence of tunnel destination option. The proposed mechanism can improve the TCP performance by mitigating packet reordering during FMIPv6 handover that can cause unnecessary congestion control due to 3 duplicate ACKs. In this paper, we implemented a multi-interface MIPv6 simulator by extending a single-interface MIPv6 simulator in NS-2, and showed that the performance of TCP traffic is improved by using the proposed multi-interface fast MIPv6.

Keywords : Fast Handover, Multi-Interface, Wireless LAN, FMIPv6, Multi-interface FMIPv6

I. 서 론

최근 들어 무선 액세스 망 기술이 발달되면서 다양한 액세스 망 접속 기술이 개발 되어 서비스 중에 있으며,

이동 단말기들은 두 개 이상의 무선 인터페이스들을 동시에 사용하여 부하 분산 또는 대역폭 집중 효과를 얻는데 사용되거나, 끊임 없는 핸드오버 서비스를 위한 백업 인터페이스로 사용되려고 하고 있다. 이와 같이 다양한 목적을 위해 이동 노드의 다중 무선 인터페이스들은 각각의 인터페이스 특성에 맞게 다양한 응용에 활용될 수 있다.^[1]

한편, IP 이동성 지원을 위해 제안된 기존의 Mobile

* 학생회원, ** 중신회원, 충남대학교 정보통신공학과
(Department of Information and Communications

Engineering, Chungnam National University)

접수일자: 2007년6월25일, 수정완료일: 2007년12월12일

IPv6^[2]는 하나의 AR(Access Router)에서 다른 AR로의 핸드오버 동안 세션 연결을 유지하기 위한 이동 노드의 프로토콜 동작을 제공하고 있지만, 핸드오버 지연으로 때문에 발생하는 지연이나 처리율에 민감한 인터넷 응용들의 품질을 떨어뜨리는 문제점을 가진다. 이에, 핸드오버 지연을 감소하기 위하여 FMIPv6(Fast Handover MIPv6) 프로토콜^[3]이 제안되었는데, 이 프로토콜은 이동 노드가 PAR(Previous AR)에서 이탈하기 전에 미리 Mobile IP관련 시그널링을 처리함으로써 이동 감지에 걸리는 지연과 새로운 CoA(Care of Address)를 설정하는데 걸리는 지연을 감소시킬 수 있도록 하였다. 그리고 핸드오버 절차동안 발생하는 지연에 의해 발생하는 패킷 손실을 방지하기 위하여 PAR로부터 NAR(New AR)로의 터널을 생성하고 이동 노드에게 전송될 패킷들을 터널링하는 방법을 이용한다.

다중 인터페이스 MIPv6^[4] 프로토콜에서도 지연과 처리율에 민감한 인터넷 응용들을 위해 핸드오버 지연과 패킷의 손실을 줄여 핸드오버 성능을 증가시킬 수 있는 고속 핸드오버 절차가 필요한데, 하나의 인터페이스를 가진 이동 단말과 달리 다중 인터페이스 이동단말이 두 개의 인터페이스를 모두 사용하는 경우, 어느 한 인터페이스만 이용하여 고속 핸드오버를 수행하면 전체적으로 심각한 패킷 순서 오류가 발생하여 트래픽 성능이 저하될 가능성이 매우 높다. 그러나 다중 인터페이스를 갖는 이동 노드는 핸드오버에 관여하는 인터페이스 외에 사용될 수 있는 또 다른 인터페이스를 가지므로 고속 핸드오버 시에 이 인터페이스를 잘 활용 하면 패킷 순서 오류를 개선하고 핸드오버 성능을 높일 수 있다. 이에 본 논문에서는 고속 핸드오버 시에 NAR이 아닌 특정 터널링 목적지로 핸드오버가 수행되는 동안 패킷을 전송 시키는 방법을 다중 인터페이스 고속 핸드오버 Mobile IPv6 절차의 확장으로 제안한다.

본 논문의 구성은 제안하는 방안에서 사용한 핸드오버 시나리오와 용어를 II장에서 설명하고, III장에서 다중인터페이스 고속 핸드오버 MIPv6 방안을 제안한다. 그리고 IV장에서는 NS-2 시뮬레이터에서 기존의 단일 인터페이스 FMIPv6 시뮬레이터를 확장하여 구현한 다중 인터페이스 FMIPv6 시뮬레이터를 구현을 소개하고, V장에서 구현된 시뮬레이터를 이용하여 제안한 핸드오버 방안에 대한 성능평가 결과를 설명하고, 마지막 VI장에서 본 논문의 결론을 맺는다.

II. 핸드오버 시나리오 및 용어 설명

본 장에서는 제안하는 핸드오버 방안에서 정의된 용어에 대해서 설명한다. 다른 용어와 약어들은 고속 핸드오버 문서인 RFC 4068^[3]에 기술된 것을 사용하였으며, 핸드오버 참조 시나리오는 그림 1과 같다. 이 참조 시나리오에서는 간단하게 이동 노드가 무선 인터페이스 두 개를 가지는 것만을 고려하였다. 이동 노드의 인터페이스 1에 연결된 AR1은 이 경우에는 핸드오버에 관여하지 않는 것으로 한다. 이동 노드의 인터페이스 2는 핸드오버에 관여하고 AR2와 AR3 사이를 이동한다고 가정한다. 인터페이스 2에서 AR2는 PAR(Previous Access Router)이고 AR3는 NAR(New Access Router)이다.

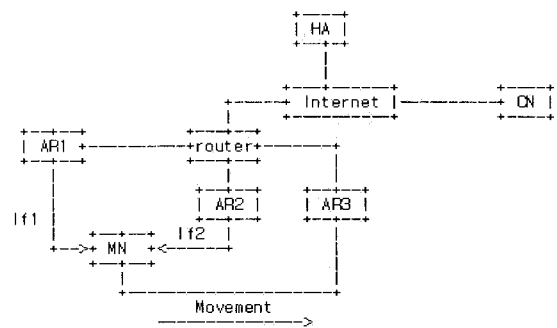


그림 1. 다중 인터페이스 고속핸드오버를 위한 참조 시나리오

Fig. 1. Reference scenario for multi-Interface fast handover.

본 논문에서는 제안하는 고속 핸드오버를 기술하기 위해 다음과 같은 용어를 사용한다.

① 핸드오버 인터페이스

현재 핸드오버 절차에 관여하는 다중 인터페이스 이동 노드의 무선 인터페이스를 말한다.

② 안정된 인터페이스

현재 안정적으로 AR에 연결되어 있고, 그것을 통해 안정적인 통신을 하고 있는 다중 인터페이스 이동 노드의 무선 인터페이스로서 그림 1에서 AR1을 통한 인터페이스이다. 현재 핸드오버에 관여하지 않는 인터페이스를 말한다.

③ 터널 목적지 옵션

다중 인터페이스 이동 노드가 고속 핸드오버를 수행할 때, 핸드오버 인터페이스를 통해 이동 노드에게 전송되던 트래픽을 재 전달하기 위해 지정하는 터널 목적

지 포인트를 나타내며 본 논문에서 소개하는 이동성 옵션 중 하나이다. 보통 이 옵션은 이동 노드의 안정적인 인터페이스 CoA를 사용한다.

④ 다중 인터페이스 고속 핸드오버 MIPv6 프로토콜

다중 CoA 등록 절차를 사용하는 다중 인터페이스 이동 노드가 고속 핸드오버를 수행할 때 핸드오버 성능을 증가시키기 위해 본 논문에서 정의한 프로토콜 절차를 말한다.

III. 다중 인터페이스 고속 핸드오버 MIPv6 프로토콜 설계

본 장에서는 제안하는 다중 인터페이스 고속 핸드오버 MIPv6 프로토콜에 대해서 절차와, 각 구성 요소의 동작 그리고 FBU(Fast Binding Update) 메시지를 확장하여 새로 정의한 MFBU(Multi-interface Fast Binding Update) 메시지에 대해서 설명한다.

1. 프로토콜 절차

제안하는 프로토콜의 핸드오버 절차는 앞서 살펴본 그림 1과 같은 구조에서 하나의 인터페이스가 이동하면서 발생하는 핸드오버 처리를 위해 제안 되었으며 핸드오버 인터페이스를 통해 RtSolPr(Router Solicitation for Proxy) 메시지를 보내면서 시작한다. 이에 AR은 하나 이상의 액세스 포인트 아이디와 정보를 담은 PrRtAdv(Proxy Router Advertisement) 메시지를 이동노드에게 보낸다. 이동 노드는 임의의 link-specific 이벤트 (Trigger)의 응답으로 또는 라우터 discovery를 수행한 후에 RtSolPr 메시지를 보낼 수도 있다. 이 메시지들의 형식은 FMIPv6^[3]에 정의 된 것과 같다.

기존 단일인터페이스 MIPv6의 경우 PrRtAdv 메시지에 담긴 정보를 가지고 이동 노드는 예상되는 NCoA(New CoA)를 생성하고, FBU메시지를 보낸다. 단일 인터페이스 FMIPv6 에서 FBU 의 목적은 PCoA (Previous CoA)와 NCoA 의 바인딩을 PAR 에게 알려서 들어오는 패킷들을 이동 노드의 새로운 위치로 터널링될 수 있도록 한다. 그런데 다중인터페이스를 사용하는 노드가 로드 분산기능을 사용하는 경우, 어느 한 인터페이스가 핸드오버를 진행하면 PAR에서 NAR로 터널링 되어 나중에 전달되는 트래픽과 핸드오버하고 있지 않은 인터페이스로 전송되는 트래픽 간에 많은 패킷 순서오류가 발생하여 TCP와 같은 응용의 성능을 저하시킬 수 있다. 따라서 제안하는 방안에서는 NCoA가

아니고, 동일한 이동 노드의 다른 인터페이스에 등록된 다른 CoA를 "tunnel destination" 이동성 옵션으로 설정할 수 있도록 제안한다. 이 메시지를 기존 FMIPv6 프로토콜의 FBU 메시지와 구분하기 위하여 MFBU 메시지로 정의 한다. 이동 노드는 MFBU 메시지를 구성할 때 제일 먼저 다중 인터페이스에 등록된 CoA들의 수를 점검하고 나서, 현재 핸드오버에 관여하지 않는 적절한 CoA들을 터널 목적지로 선택한다.^[4] 그 후 이동노드는 선택된 CoA들 중, 각 인터페이스가 터널링 될 트래픽에 대해 적절한 처리 특성을 가지는지, 아닌지를 점검한다. 또한 이동 노드는 선택된 인터페이스가 트래픽을 수용할 수 있는지 해당 인터페이스의 가용 대역폭을 검사한다. 최종적으로 선택된 인터페이스의 CoA를 MFBU 메시지의 "tunnel destination" 이동성 옵션에 넣고 'T' 플래그를 설정하여 이 옵션의 존재를 표시한다.

PAR 은 MFBU 메시지를 받은 후에, PCoA로 도착하는 패킷들을 "tunnel destination" 즉, 이동 노드의 또 다른 안정된 인터페이스의 CoA로 터널링하기 시작한다. 터널은 이동 노드가 새로운 CoA를 HA(Home Agent) 또는 CN(Correspond Node) 에게 등록 완료할 때까지 유지된다. 역방향 트래픽에서는 이동 노드가 NCoA로 바인딩 업데이트를 완료할 때까지 'tunnel destination'의 인터페이스를 통하여 PAR 로 역 터널을 사용하여 패킷을 보내야만 한다. 왜냐하면 그런 역 터널은 송신지 IP 주소로서 PCoA를 포함한 패킷들이 ingress filtering에 의해 손실되는 것을 방지하기 때문이다.^[5] 그런 후에 HI (Handover Initiate), HACK (Handover Acknowledge), FBACK(Fast Binding Acknowledge) 그리고 FNA (Fast Network Attachment)메시지들은 기본 FMIPv6 프로토콜에서와

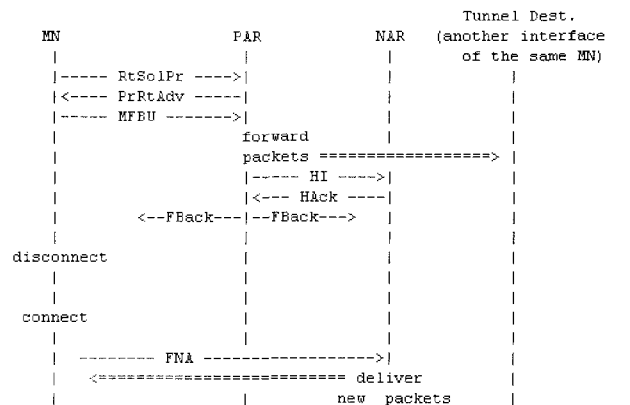


그림 2 다중 인터페이스 이동 노드를 위한 고속 핸드오버 절차

Fig. 2. Fast handover procedure for multi-Interface mobile node.

같은 방법으로 사용된다.^[3] 다중 인터페이스 고속 핸드 오버 FMIPv6 절차의 전체 시나리오를 그림 2에 나타 내었다.

2. 이동 노드 동작

이동 노드가 "L2 trigger"로 시그널 품질의 페이딩과 하나 이상의 근처 액세스 포인트들을 발견할 때, 이동 노드는 근처 액세스 포인트들의 서브넷 라우터 정보를 찾기 위해 RtSolPr 메시지를 보낸다. 이에 대한 응답으로 이동 노드는 현재 PAR에 연결된 액세스 포인트인지 또는 이동 노드가 이동해야 할 새로운 라우터에 연결된 액세스 포인트 인지를 알려주는 PrRtAdv 메시지를 받는다. 만약 근처의 액세스 포인트가 핸드오버를 위한 새로운 라우터에 연결되어 있고 다중 인터페이스 고속 핸드오버 절차를 지원한다면, 이동 노드는 PAR에 MFBU 메시지를 보낸다.

기본적으로 터널 목적지 인터페이스는 현재 다른 핸드오버절차에 관여되지 않아야 한다. 선택된 인터페이스가 터널링 될 트래픽 특성에 적합하고 그 인터페이스의 가용대역폭이 터널링 될 트래픽에 충분히 적합하다면, 이동 노드는 MFBU 메시지의 "tunnel destination" 이동성 옵션에 해당 인터페이스의 CoA 를 넣고 'T' 플래그를 1 로 설정한다. 이와 같이 적절한 터널링 목적지를 선택한 후 핸드오버 인터페이스는 데이터 수신에는 관여할 필요가 없고, 핸드오버 시그널링 절차에 집중할 수 있다. 왜냐하면 핸드오버 인터페이스로 오는 트래픽은 PAR로 부터 이동 노드의 또 다른 안정된 인터페이스로 터널링 되기 때문이다. 만약 이동 노드가 CN에 보낼 트래픽이 있다면 ingress filtering을 피하기 위해 핸드오버가 끝날 때까지 'tunnel destination' 옵션의 인터페이스를 사용하여 PAR로 역 터널을 통해 패킷을 전송하여야 한다.^[5]

3. 액세스 라우터 동작

RtSolPr 메시지는 새로운 라우터에 대한 정보와 PAR이 아닌 다른 새로운 라우터에 연결된 액세스 포인트들에 연결할 것인지에 대한 질문을 함께 보낸다. RtSolPr 메시지의 응답으로 PAR은 이동 노드에게 요청된 액세스 포인트들의 연결 상태를 알리기 위한 PrRtAdv 메시지를 보낸다. 핸드오버 시점이 임박했을 때, PAR은 이동 노드로부터 터널 목적지 옵션을 포함하고 'T' 플래그가 1로 설정된 MFBU 메시지를 받을 수 있다. 그것은 이동 노드가 일반적인 NAR이 아닌 터

널 목적지 옵션에 있는 특정 IP 주소로 해당 인터페이스의 트래픽을 터널링하기 원하는 것을 나타낸다. 이 시점부터 PAR 은 "tunnel destination", 즉 이동 노드의 또 다른 인터페이스의 CoA로 PCoA에 도착하는 패킷들을 터널링하기 시작한다. 이 터널은 이동 노드가 HA 나 CN에 바인딩 업데이트를 완료할 때까지 사용된다. MFBU의 응답으로서 PAR은 HI와 HAcK 메시지의 교환을 끝낸 후, PCoA와 NCoA 사이의 바인딩을 만들고 이동 노드에게 FBack 를 보낸다.^[3]

4. 메시지 포맷 변경

MFBU(Multi-Interface Fast Binding Update) 메시지는 FMIPv6 FBU(Fast Binding Update)^[11] 메시지의 확장이다. 즉, FBU 메시지에 두 가지 필드가 추가되어 만들어지는데, 하나는 "tunnel destination" 이동성 옵션이고 다른 하나는 'T' 플래그이다. MFBU 메시지의 형식은 그림 3과 같은데, 여기서 T 플래그는 FBU 메시지가 "tunnel destination" 이동성 옵션을 가지고 있는 것을 나타낸다. 만약 Tunnel destination option에 'T' 플래그가 1로 설정되면, "tunnel destination" 이동성 옵션이 포함되어야 한다.

Tunnel Destination option은 다른 인터페이스의 CoA로 이동 노드의 핸드오버 인터페이스를 대신해 트래픽을 재전달하기 위해 PAR에게 터널링 도착 주소를

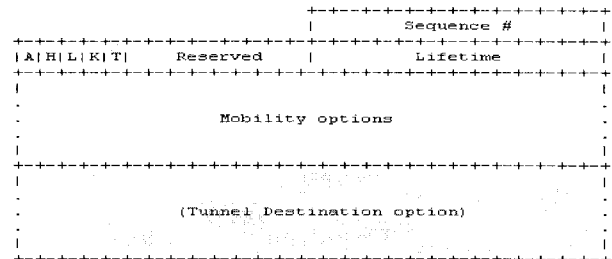


그림 3. 다중 인터페이스 MFBU 메시지
Fig. 3. Mult-Interface MFBU Message.

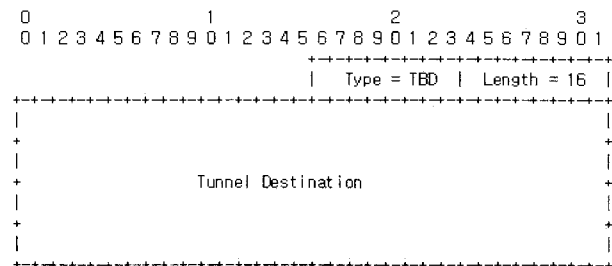


그림 4. Tunnel Destination 이동성 옵션
Fig. 4. Tunnel Destination mobility option.

알려주기 위하여 MFBU 메시지의 이동성 옵션에 포함된다. 이 옵션은 오직 다중 인터페이스 FBU 메시지에만 유효하다. Tunnel Destination option 의 형식은 그림 4와 같이 구성된다.

IV. 다중 인터페이스 MIPv6 시뮬레이터 구현

본 장에서는 기존에 구현된 FMIPv6 시뮬레이터와 본 논문에서 제안한 MFBU 메시지를 처리하는 다중 인터페이스를 지원하는 시뮬레이터를 구현한 내용에 대해서 설명한다.

1. 다중 인터페이스를 지원하는 이동 노드 구현

UC 버클리에서 개발되어 현재 네트워크 시뮬레이터로 널리 사용되고 있는 NS-2^[6]는 유선의 네트워크뿐만 아니라, 무선의 이동 노드를 지원하고 있다. NS-2에서의 이동 노드의 프로토콜 스택 구성도를 보면 그림 5와 같다. 하위 무선채널에 의해 수신된 데이터는 NetIF와 MAC 계층을 통해 LL 계층에 전송되며, 다시 상위의 addr demux와 port demux에 의해서 구별되어 응용 에이전트에 전송된다. 반대로 상위 에이전트에서 송신되는 데이터는 LL 계층에서 ARP 테이블을 참조하여 데이터를 전송하기 위한 경로를 파악한 후 IFq에 입력되며, MAC 계층과 무선 채널 계층을 통해 전송된다.

본 논문에서는 다중 인터페이스를 지원하는 이동 노드를 구현하기 위해서 기존의 프로토콜 스택을 확장하여 그림 6과 같이 점선으로 표시된 부분을 추가 구성하였다. 두 개의 인터페이스를 지원하기 위해서 NetIF,

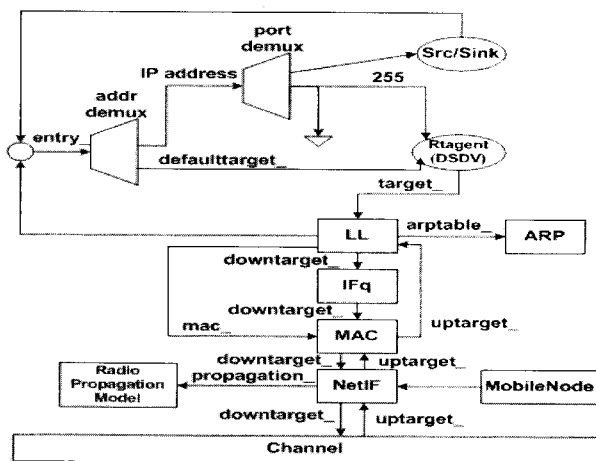


그림 5. 기존 NS-2에서 이동 노드에 대한 계층 블록도
Fig. 5. Layer block diagram of mobile node in existing NS-2.

MAC과 IFq를 추가하였으며, 서로 다른 채널 사용을 가능하게 하기 위해서 새로운 채널을 추가하였다. 이와 같은 구현을 통해 서로 다른 무선 인터페이스는 다른 채널을 간섭 없이 독립적으로 사용할 수 있다.

그림 7은 LinkDelay 에서 파생된 LL클래스에 대한 주요 멤버 변수를 보여준다. 2 라인의 ifq_는 하위 계층의 큐를 가리키는 포인터이며 3 라인과 4 라인의 mac_, arptable_은 각각 MAC 계층과 ARPTable을 가리키는 포인터를 나타낸다.

LL블록은 상위에서 데이터가 전송되었을 경우 arptable_을 통해 ARP(Address Resolution Protocol) 블록을 참조한 후, 하위 큐에 추가하기 위해서 ifq_를 사용한다. 또한 mac_을 통해 MAC 계층에 해당 이벤트를 전달하게 된다. 5 라인의 seqno_는 MAC 계층에서 데이터 송수신에 사용되는 패킷에 대한 순차적인 번호를 관리하기 위해서 사용되며, 6 라인의 macDA_에는 arptable_을 통해 패킷이 전송될 다음 홉에 대한 주소가

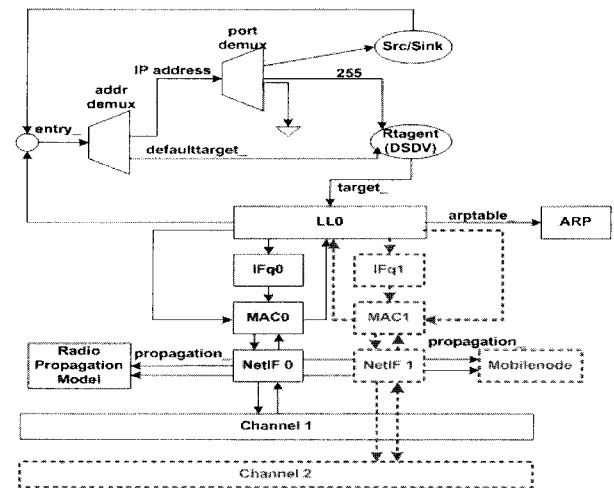


그림 6. 다중 인터페이스를 지원하는 NS-2 이동 노드에 대한 계층 블록도

Fig. 6. Layer block diagram of NS-2 mobile node for supporting multi-interface.

```

1 class LL : public LinkDelay {
2     Queue* ifq_;
3     Mac* mac_;
4     ARPTable* arptable_;
5     int seqno_;
6     int macDA_;
7     NsObject* downtarget_;
8     NsObject* uptarget_;
9     int M_seqno_[Max_interface];
10    Queue* M_ifq_[Max_interface];
11    Mac* M_mac_[Max_interface];
12    NsObject* M_downtarget[Max_interface];

```

그림 7. LL 계층 클래스의 주요 멤버 변수

Fig. 7. Primary member variable of the LL layer class.

```

1 class MobileNode : public Node {
2     ...
3     int base_stn_;
4     int coa_;
5     int MultiPathCount_;
6     int M_base_stn_[Max_interface];
7     int M_coa_[Max_interface];
8 }
    
```

그림 8. 이동 노드 계층 클래스의 주요 멤버 변수
Fig. 8. Primary member variable of mobile node layer class.

설정되며 MAC 헤더를 생성하기 위해서 사용된다. 8 라인의 `uptarget_`은 하위 MAC 계층에서 전달된 데이터를 `entry_`로 전송하기 위해서 사용된다. 9~12 라인에 추가된 멤버 변수들이 다중 인터페이스를 지원하기 위해서 필요한 부분으로써, 그림 6에 추가된 두 번째 프로토콜 스택의 정보를 관리하기 위해서 사용된다. LL 블록은 상위에서 전송되는 데이터를 두 개의 인터페이스 중에서 하나를 선택하여 보내기 위해서 10 라인 `M_ifq_[Max_interface]`와 11 라인 `M_mac_[Max_interface]`를 관리한다.

그림 8은 이동 노드를 정의한 `MobileNode` 클래스에서 사용되는 주요 멤버 변수를 나타낸다. 3 라인의 `base_stn_`은 자신과 연결된 base-station의 ID를 나타내며, 4 라인의 `coa_`는 base-station에서 할당 받은 CoA를 관리하기 위해서 사용된다. 5 라인의 `MultiPathCount_`는 이동노드가 사용하는 멀티인터페이스의 수를 나타내며 이동노드가 기본적으로 하나의 인터페이스만 사용할 경우 이 값은 0이 된다. 6라인과 7라인은 다중 인터페이스를 사용할 경우 해당 인터페이스와 연결된 base-station ID와 각 base-station에서 할당받은 CoA를 저장하기 위해서 사용된다.

2. 다중 CoA 등록 및 부하 분산 기능의 HA 구현

이동노드가 다른 망으로 이동하였을 경우 해당 망의 AR과 접속절차를 통해서 CoA를 할당 받는다. 그 후 이동노드는 HA(Home Agent)에게 CoA를 등록하고 주기적으로 갱신하기 위해서 메시지를 보내게 된다. 그림 9는 이동노드가 HA에게 자신의 CoA를 등록하는 절차에 대한 함수 호출관계를 나타낸다.

이 때 다중 CoA를 등록하기 위해서 기존의 바인딩업데이트 메시지에 구분자를 두어 갱신과 추가를 구분할 수 있도록 하였으며 HA는 자신이 관리하는 이동 노드가 외부의 BS로 이동하였을 경우 이동 노드가 정상적으로 데이터를 수신하기 위해서 외부로 이동한 노드로

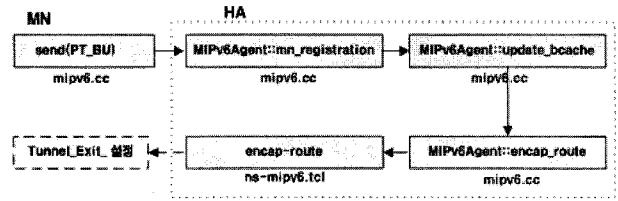


그림 9. 이동 노드가 HA에 CoA를 등록하는 절차
Fig. 9. Procedure of CoA registration to HA by a mobile node.

```

1 class Entry {
2     ...
3     int MultiCoACount;
4     int M_caddr[Max_interface];
5     double weight[Max_interface];
6
7 }
    
```

그림 10. 다중 CoA 지원을 위한 HA entry 구조체
Fig. 10. HA entry structure for supporting multiple CoAs.

```

1 int BSAgent::command(int argc, ...) {
2     if (strcmp(argv[1], "get-mcoa-by-haddr") ==
3         0){
4         double r = Random::uniform(0, 1.0);
5         double total_weight = 0;
6         for(int i = 0 ; i < MultiCoACount; i++) {
7             total_weight += weight[i];
8             if(total_weight < r){
9                 tcl.resultf("%d", n->M_caddr[i]);
10                return TCL_OK;
11            }
12            tcl.resultf("%d", n->caddr);
13            return TCL_OK;
14        ...}
15 }
    
```

그림 11. HA에서 다중 CoA를 이용한 load 분산
Fig. 11. Load Distribution using multiple CoAs in HA.

전송되는 데이터가 수신되면 터널링을 통해서 이동 노드에게 데이터를 전송한다. 이를 위해서 이동 노드들에 대한 정보를 관리하는 리스트가 필요하며, NS-2에서는 이동 노드에 대한 다양한 정보를 `Entry` 클래스를 이용하여 저장한다. 그림 10에서 3~5 라인은 다중 CoA 등록을 지원하기 위해서 `Entry` 클래스에 추가된 코드를 나타낸다. 3 라인의 `MultiCoACount`는 다중 CoA의 사용 여부와 함께 CoA의 개수를 나타내며, 4라인의 `M_caddr`은 다중 CoA 주소를 나타내고, 5라인의 `weight`는 각각의 CoA에 전송 시 데이터를 부하 분산하여 보내게 되는데 이때 사용하는 부하분산 비율을 나타낸다.

그림 11는 HA에서 이동노드의 `haddr`을 통해 다중의 CoA로 Encapsulation할 CoA를 얻기 위해 `get-mcoa-by-haddr` 명령어를 호출하였을 경우 수행되는 코드를

나타낸다. 구현된 내용을 살펴보면, 4라인에서 균일한 랜덤값을 발생 시키는 함수를 통해서 0에서 1사이의 임의의 랜덤한 값을 취하고, 5~13라인은 각 CoA에 할당된 가중치^[7]에 맞게 트래픽을 각 인터페이스에 할당하기 위해서 계산되는 과정을 나타낸다.

V. 시뮬레이션 및 성능 평가

본 장에서는 제안하는 방안에 대해 NS-2를 이용하여 구현된 시뮬레이터에 UDP 트래픽과 TCP 트래픽을 이용하여 시뮬레이션을 수행하고 그 결과를 분석하여 성능을 평가한다. 먼저 UDP 트래픽을 이용해서는 제안하는 MFBU 메시지와 기존의 FBU 메시지에 대해서 트래픽이 어떻게 전달되는지 프로토콜의 정상동작 여부를 검증하였고, TCP를 이용하여 핸드오버 성능을 평가하였다.

1. 시뮬레이션 환경

본 시뮬레이션의 토폴로지 구성과 링크환경은 그림 12와 같이 구성하였는데, 멀리 떨어져 있는 HA와 CN의 조건을 시뮬레이션하기 위해 각각의 링크에는 큰 값의 지연을 설정하였으며, 하위 BS와 라우터는 비교적 가까이 있으므로 작은 값의 지연을 각각 설정하였다. 그리고 5m/s의 움직임을 갖는 이동 노드를 bs0에서 bs3까지 이동 시키며 시뮬레이션을 수행하였다. 시뮬레이션 동안 다중 인터페이스의 동작은 희망인 bs0에서는 한 개의 인터페이스만 사용하다가 bs1으로 이동하면서 핸드오버가 시작 되면 다중 인터페이스를 사용하도록 설정하였다.

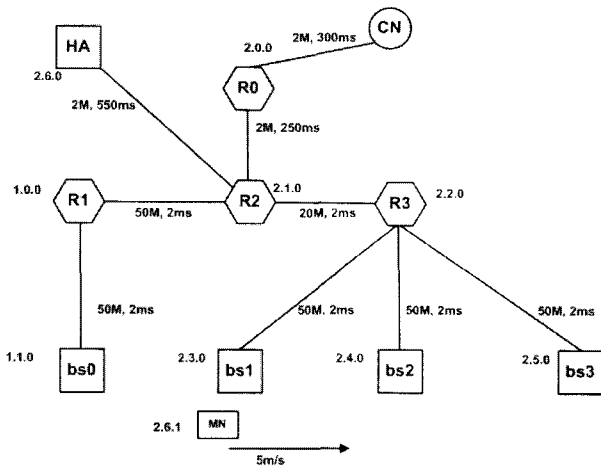


그림 12. 시뮬레이션 토폴로지
Fig. 12. Simulation Topology.

다음은 시뮬레이션 시 핸드오버에 영향을 주는 파라미터와 사용한 트래픽이다.

- 무선 환경에서의 BS의 전송 반경 : 250m
- L2 trigger : BS로부터 244m의 거리에서 Down Trigger
- Beacon interval : 0.08초
- 트래픽 설정 : UDP 및 FTP/TCP New Reno

2. UDP Traffic을 이용한 패킷 흐름 확인

시뮬레이션에서 패킷의 흐름이 알고리즘대로 전송되는지 검증할 때는 패킷의 수가 가변적인 TCP 트래픽 보다는 일정한 간격으로 패킷이 전송되는 UDP 트래픽을 이용하면 쉽게 검증을 할 수 있다. 그림 13과 14는 NS-2의 NAM^[8] 파일을 통해서 기존 FBU 메시지와 MFBU 메시지를 이용하였을 때 PAR에서 NAR로 HI 메시지가 전달되면서 각각 어떻게 동작하는지 동작 상태를 UDP 트래픽을 이용하여 검증한 것이다. 그림 13과 14에서 동작하는 모습을 통해 확인 되듯이 기존의 FBU 메시지를 이용하는 경우에는 그림 13의 PAR(7)에서 NAR(8)로 HI 메시지를 전달되고, 핸드 오버하는

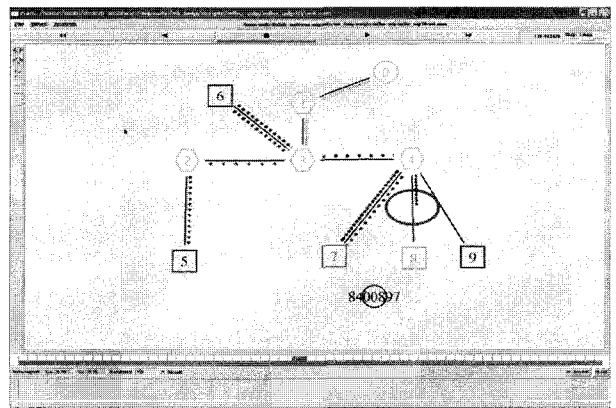


그림 13. FBU 메시지를 받고 난 후 동작 절차
Fig. 13. Operation when FBU message is received.

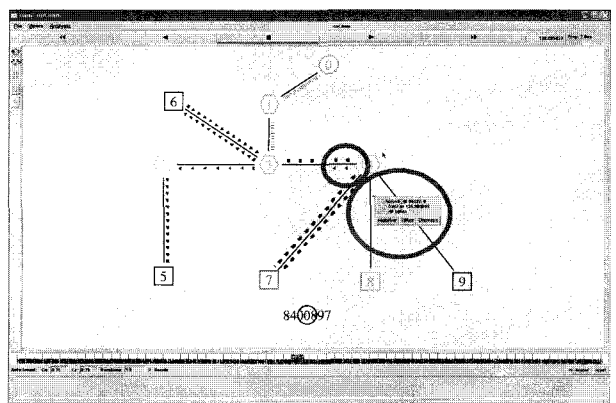


그림 14. MFBU 메시지를 받고 난 후 동작 절차
Fig. 14. Operation when MFBU message is received.

동안 PAR에 버퍼링 되어 있던 패킷이 NAR로 터널링 되는 모습을 볼 수 있다. 하지만 본 논문에서 제안하는 새로 정의된 MFBU 메시지를 이용하는 경우에는, 그림 14와 같이 HI 메시지를 전달함과 동시에 이동 노드에서 MFBU 메시지를 통해 전달된 또 다른 특정 CoA를 갖는 AR(5)로 패킷을 터널링 하게 된다.

3. TCP Traffic을 이용한 시뮬레이션 성능 평가

TCP 트래픽은 망의 혼잡과 관계없이 일정하게 데이터를 전송하는 UDP 트래픽과 달리 송신노드는 수신 노드에서 보내진 ACK 패킷을 조사하여 망의 혼잡에 따라 혼잡 윈도우(congestion window, CWND)를 조절하게 된다. 이때 수신 노드로부터 전달 받은 ACK 중 연속해서 3개가 중복 되면 지금 사용하는 망이 혼잡하다고 여기고 자신의 혼잡윈도우를 줄이게 된다. 이와 같이 동작하는 TCP 특성 때문에 TCP 트래픽에서 패킷의 순서 뒤집어짐은 TCP 성능에 큰 영향을 주게 된다.

만약 다중 인터페이스를 이용하는 이동 노드에 TCP를 사용한다고 하고, HA가 If1과 If2에 균등하게 TCP 트래픽을 분산한다고 하면 그림 15와 같이 각각의 인터페이스를 통해 전달될 것이다. 이동 노드가 핸드오버를 하지 않을 때 패킷의 도착순서는 뒤바뀌지 않을 것이다. 하지만 인터페이스 If2가 핸드오버를 시작하고 이동 노드가 원래의 fast 핸드오버 절차를 따를 때 그리고, 이동 노드의 핸드오버 이벤트가 그림 15와 같이 발생한다면, 패킷의 도착순서는 "1 2 3 4 5 7 9 11 6 8 10 12 13 14 15" 와 같이 될 것이며, 이동 노드가 11번 패킷을 받았을 때 순서 뒤바뀔 문제는 3 Duplication ACKs를 발생 시킨다.

본 장에서는 TCP New Reno를 이용하여 240초 동안 트래픽을 발생시키며 제안한 MFBU와 FBU 메시지 수신 시 TCP의 성능 비교를 각각 시뮬레이션하고 그 결과를 분석하였다.

TCP 성능은 혼잡윈도우가 높게 나타날 때 전송량이 많음을 나타낸다.^[9] 따라서 FBU 메시지를 사용하는 기

```

If1 (CoA1) : 1 3 5 7 9 11 13 15 .....
If2 (CoA2) : 2 4 6 8 10 12 14 .....
|
|<- handover instant
|      (FBU or MFBU message arrival to AR2)
|
|<- FNA message arrival to AR3
    
```

그림 15. FMIPv6 또는 MFMIPv6에서의 핸드오버 이벤트
Fig. 15. Handover events in FMIPv6 or MFMIPv6.

존의 FMIPv6와, MFBU 메시지를 사용하는 다중 인터페이스 MFMIPv6에 대해서 각각 CWND를 분석함으로써 전체적인 전송량 분석이 가능하다.

시뮬레이션 결과 그림 16과 같이 CWND가 각각 변했는데, 희망을 떠나 첫 번째 BS에 도착하여 핸드오버하기 전까지의 CWND의 변화 모습은 서로 같다. 그러나 FBU 메시지와 MFBU 메시지를 이용해 각각 핸드오버가 발생한 58.48초, 138.48초, 218.48초시에는 각각 다른 CWND의 변화 모습을 보인다. 그 이유는, 다중 인터페이스를 사용하지만, 기존의 FMIPv6의 FBU 메시지를 사용하는 경우, NAR 쪽에서 들어오는 TCP 패킷은 PAR과 NAR 사이에서 큐잉 되었다가 포워딩 되어 bs0로 전달되는 패킷들과 패킷의 순서가 크게 뒤바뀌기 때문에 3개의 중복 ACK가 발생하게 되어 TCP 혼잡제어 메커니즘에 의해 TCP 성능이 감소됨을 볼 수 있다. 그러나 MFMIPv6의 MFBU 메시지로 처리하는 경우에는, 이동 노드가 보낸 MFBU 메시지를 AR이 수신하는 즉시 MFBU를 통해 요청한 다른 인터페이스 쪽

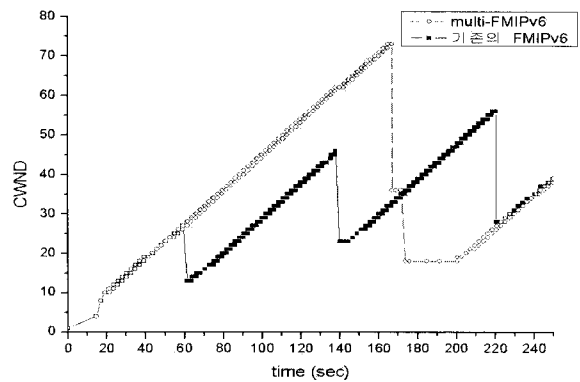


그림 16. 두 핸드오버 방식의 혼잡윈도우 변화
Fig. 16. congestion window variation in two handover mechanisms.

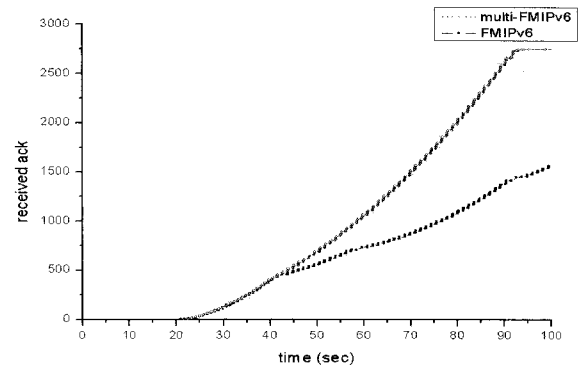


그림 17. 핸드오버 시 수신한 ACK 시퀀스 번호
Fig. 17. Received ACK sequence number at handover.

으로 패킷이 재 전달 되게 되므로 TCP 서비스에 FMIPv6 보다 덜 영향을 주어 핸드오버와 상관없이 마치 하나의 인터페이스를 이용하는 것과 같은 TCP의 CWND 모습을 보여준다.

그림 17은 핸드오버 시에 수신한 ACK 번호를 비교하였다. TCP는 패킷을 전송한 후 수신자로부터 ACK를 받음으로써 정상적인 패킷 전송을 확인하게 된다. 따라서 동일 시간 단위에서 TCP 시퀀스 번호가 크게 증가했을 경우 전송량이 큰 것을 의미한다. 그림 17에서 볼 수 있듯이 MFBU 메시지를 사용하는 MFIPv6 경우 TCP의 시퀀스 번호가 더 빠르게 증가함을 볼 수 있으며, 기존의 FBU 메시지를 사용하는 FMIPv6 경우에는 전송한 데이터에 대한 ACK 수신이 되지 않아, 다시 재전송함으로써 TCP 시퀀스 번호가 잘 증가하지 못함을 볼 수 있다. 이와 같은 분석 결과 이동 노드가 다중 인터페이스를 지원하는 경우에는 FBU 메시지를 쓰는 기존의 FMIPv6 보다, 'T' 플래그와 tunnel destination" 이동성 옵션을 포함한 MFBU 메시지를 이용하여 트래픽을 재전달하는 제안하는 MFIPv6을 이용하면 더 좋은 품질의 핸드오버를 수행 할 수 있음을 알 수 있다.

VI. 결론 및 향후 연구 과제

본 논문에서는 서로 다른 여러 가지의 무선인터페이스를 가지는 multi-homing^[10] 이동 단말이 다중 인터페이스를 동시에 사용하면서 이중 액세스 망간을 핸드오버 할 때 끊김 없는 서비스를 제공 받을 수 있도록 기존의 FBU 메시지를 확장한 MFBU 메시지를 이용한 MFIPv6 프로토콜을 제안하였다. 그리고 이를 성능 분석하기 위해 다중 CoA를 등록하는 기능과, 가중치에 따라 부하를 분산 할 수 있는 HA^[12]를 비롯하여, 다중 인터페이스를 갖는 이동노드를 포함하는 시뮬레이터를 구현하고 UDP 트래픽을 이용하여 제안하는 방안의 동작 상태를 검증하고 TCP 트래픽을 이용하여 시뮬레이션을 수행 및 결과를 분석하여 순서 어긋남을 줄임으로써 얻는 성능 개선을 검증하였다. 결과로서 다중 인터페이스 MIPv6 환경에서의 핸드오버 시에는 제안하는 MFIPv6 프로토콜이 기존의 FMIPv6에 비해, 핸드오버 시 TCP 트래픽에 대해서 핸드오버 시 패킷순서 오류의 영향이 적음을 알 수 있었으며, 그 결과 더 좋은 성능 향상 결과를 얻었다.

향후 연구 과제로써는 MIPv6 환경이 아닌 와이브로 망, 3G 망 또는 다양한 망이 혼재된 망에서 이동 노드

가 고속으로 수직 또는 수평 핸드오버를 수행할 때 패킷 유실이 없는 고품질의 핸드오버를 지원하기 위한 방안을 확장 연구하는 것이다.

참고 문헌

- [1] Jukka Ylitalo, Tony Jokikyyny, Tero Kauppinen, Antti J. Tuominen, Jaakko Laine, "Dynamic Network Interface Selection in Multihomed Mobile Hosts," hicss, p. 315, 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 9, 2003.
- [2] D. Johnson, C. Perkins, and J. Arkko, "Mobility Support in IPv6", RFC 3775, June 2004.
- [3] R. Koodli, "Fast Handovers for Mobile IPv6" RFC 4068, July 2005.
- [4] R. Wakikawa, T. Ernst, and K. Nagami, "Multiple Care-of Addresses Registration", draft-ietf-monami6-multiplecoa-00 (work in progress), June 2006.
- [5] N. Montavont, T. Noel, "Home Agent Filtering for Mobile IPv6", draft-montavont-mobileip-ha-filtering-v6-00.txt, July 23, 2003.
- [6] Ns-2, <http://www.isi.edu/nsnam/ns/>
- [7] H. Huang, J. Cai, A.J. Kessler, and C. Fu, "Load-sharing in Wireless Multi-homed Systems", pp. 3489-3493, IEEE International Conference on Communications ICC, 2005.
- [8] NAM : <http://www.isi.edu/nsnam/nam/>
- [9] An Kyu Hwang Jae Yong Lee Hyun Duk Kim Whoi Jin Jung, and Byung-Chul Kim., "New TCP Congestion Control Algorithm Based on Bandwidth Measurement for Seamless Vertical Handoffs in Heterogeneous Mobile Networks", pp. 75-83, WWIC 2006, LNCS 3970
- [10] S. Kashihara, T. Nishiyama and K. Iida, "Path Selection Using Active Measurement in Multi-homed Wireless Networks", Proceedings of the 2004 International Symposium on Applications and the Internet, IEEE 2004.
- [11] H. Soliman, N. Montavont, N. Fikouras, K. Kuladinithi, "Flow binding in Mobile IPv6", draft-soliman-monami6-flow-binding-00(work in progress), February 2006.
- [12] S. Cho, J. Na and C. Kim, "A Dynamic Load Sharing Mechanism in Multihomed Mobile Networks", pp. 1459-1463, IEEE International Conference on Communications ICC, 2005.

저 자 소 개



박 만 규(학생회원)
 1999년 공주대학교 물리학과
 학사
 2001년 공주대학교 전기전자정보
 통신공학과 석사
 2001년~2004년 시스윌 기업
 부설 통신기술 연구소
 전임연구원

2004년~2006년 케이디넷 통신사업본부 기술팀
 전임연구원/대리

2007년~현재 충남대학교 전자전파정보통신
 공학과 박사과정

<주관심분야 : 이동통신 네트워크, 데이터 통신
 센서네트워크>



이 재 용(종신회원)
 1988년 서울대학교 전자공학과
 학사
 1990년 한국과학기술원 전기 및
 전자공학과 석사
 1995년 한국과학기술원 전기 및
 전자공학과 박사

1990년~1995년 디지콤 정보통신 연구소
 선임연구원

1995년~현재 충남대학교 정보통신공학부 교수

<주관심분야 : 초고속통신, 인터넷, 네트워크 성
 능분석>



황 안 규(학생회원)
 1999년 한남대학교 물리학과
 학사
 2004년 충남대학교 정보통신
 공학과 석사
 2005년~현재 충남대학교
 전자전파정보통신공학과
 박사과정

<주관심분야 : 이동통신 네트워크, 데이터 통신,
 초고속 통신>



김 병 철(종신회원)
 1988년 서울대학교 전자공학과
 학사
 1990년 한국과학기술원 전기 및
 전자공학과 석사
 1996년 한국과학기술원 전기 및
 전자공학과 박사

1993년~1999년 삼성전자 CDMA 개발팀

1999년~현재 충남대학교 정보통신공학부 부교수

<주관심 분야 : 이동인터넷, 이동통신 네트워크,
 데이터통신>