

멀티모드 단말을 위한 가상 인터페이스 구현 연구

정희원 이경희*, 이성근*, 이은준*, 조경섭*, 종신회원 이현우*, 류원*, 홍승필**

Research on the Implementation of the Virtual Interface on Multi-mode Mobile Nodes

Kyounghee Lee*, Seongkeun Lee*, Eunjun Rhee*, Kyoung-seob Cho* *Regular Members*,
Hyun-woo Lee*, Won Ryu*, Seng-phil Hong** *Lifelong Members*

요약

본 논문에서는 유무선 통합 환경의 차세대 방통융합 서비스를 위한 기반 기술로서 다양한 액세스 망에 대한 다중 접속 인터페이스를 갖는 멀티모드 단말의 인터페이스 가상화 기법을 소개한다. 제안된 가상 인터페이스 기술은 멀티모드 단말의 각 물리 인터페이스에 대해 상이한 IP 주소를 할당하고 패킷 전달 시 이들 주소간의 변환을 수행하는 기존의 방식과 달리, 고정된 하나의 단일 IP 주소만을 이용하여 다수의 물리 인터페이스들을 가상화하는 기능을 제공한다. 따라서 패킷 전달 시 IP 주소 변환 등 단말 기능 변경을 요구하지 않으므로 Proxy MIP와 같은 네트워크 기반 IP 이동성 제어 기술과 연계하여 적용하기에 용이한 장점을 제공한다. 또한 기존의 OS가 단일 단말 내의 상이한 물리 인터페이스에 대해 동일 IP 주소를 할당할 수 없도록 하는 제약사항을 극복함으로써 MBB(Make-Before-Break) 방식의 핸드오버 제어뿐만 아니라 BBM(Break-Before-Make), 다중 접속(Multi-connection) 등과 같은 다양한 방식을 모두 지원할 수 있는 장점을 갖는다. 제안된 가상 인터페이스는 네트워크 장치 인터페이스 표준 규격으로 개발된 NDIS 라이브러리를 이용하여 구현되었으며, 테스트베드 구축과 시험 프로그램 개발을 통해 주요 기능의 시험 및 동작 결과를 확인할 수 있었다. 이러한 가상 인터페이스를 통해 멀티모드 단말이 이중 액세스 망간 핸드오버를 수행하는 도중에도 동일한 IP 주소를 이용하여 실시간 멀티미디어 서비스 세션을 끊임없이 유지하도록 지원하는 것이 가능하다.

Key Words : Virtual Interface, 멀티모드 단말, 인터페이스 가상화, NDIS(Network Driver Interface Specifications)

ABSTRACT

In this paper, we propose the virtual interface management scheme on the multi-mode mobile node for supporting multiple connections to various access networks in fixed mobile convergence (FMC) networks. The proposed scheme supports the virtualization of multiple physical network interfaces by presenting only the virtual interface to beyond IP layers and hiding physical network interfaces from them. In the proposed scheme, only one IP address is allocated to virtual interface without any IP allocations to physical network interfaces. Therefore, the proposed scheme does not change its IP address and keep it during the vertical handover, so that it can support the seamless handover of real-time multimedia services among heterogeneous access networks. The proposed scheme is implemented on the multi-mode mobile node with multiple network interfaces by using NDIS (Network Driver Interface Specifications) libraries. Through the mobility test-bed and the test application of virtual interface, we evaluate and analyze the performance of the proposed scheme.

* 본 연구는 지식경제부 및 한국산업기술평가관리원의 IT산업원천기술개발사업의 일환으로 수행하였습니다 [2008-S-006-03, 유무선 환경의 개방형 IPTV(IPTV2.0) 기술 개발].

* 한국전자통신연구원 방송통신융합연구부 IPTV연구부 융합서비스네트워크연구팀 (leekhe@etri.re.kr)

** 성신대학교 IT학부 (philhong@sungshin.ac.kr)

논문번호 : KICS2009-12-649, 접수일자 : 2009년 12월 30일, 최종논문접수일자 : 2010년 4월 6일

1. 서 론

방송, 통신, 인터넷을 유기적으로 결합하고자 하는 방통융합 서비스는 다양한 광대역 액세스 기술 및 이를 통합하는 IP 기반의 유무선 통합 환경을 기반으로 새로운 서비스 모델과 차세대 기반 기술 개발을 요하는 단계로 진화 중에 있다. “IPTV2.0”으로 대표되는 차세대 방통융합 서비스는 기존 1세대 IPTV의 특징 및 장점을 승계하면서 이동성, 지능성, 참여성 등 측면이 강조되는 미래형 TV 서비스로 정의될 수 있다^[1,2]. 특히 이동성 기술은 무선랜, WiBro, 3G/4G 등 지속적으로 발전되고 있는 다양한 무선 통신 인프라를 이용하여 가입자가 언제, 어디서나 끊김없는 고품질의 실시간 콘텐츠를 이용할 수 있도록 하는 Mobile IPTV 서비스의 바탕이 되며, 음성, 데이터, 방송 및 이동성 지원 서비스를 아우르는 QPS(Quadruple Play Service)와 이종 단말 간의 서비스 연속성을 지원하는 n-Screen 환경을 실현하기 위한 핵심 기술로 분류된다.

최근 들어 기존에 독립적으로 운영되던 다수의 유무선 액세스 망을 통합하여 단일한 서비스 환경 및 이종 망간 이동형 서비스 연속성을 지원하기 위한 시도가 출현하고 있으며, 이에 발맞추어 다수의 유무선 네트워크 인터페이스를 갖는 멀티모드(Multi-mode) 단말이 속속 등장하고 있다. 이종 망간의 이동성 지원을 위한 주요 요구사항 중 하나는 응용 계층에 대해 단말의 접속 망 변경 및 IP 주소 변경 상황을 감추어 서비스 세션이 끊기는 현상을 방지하는 기능을 제공하는 것이다. 이를 위해서는 IP 계층에서 동일한 주소를 유지하면서 이종 망간 핸드오버를 제어하는 IP 이동성 기술뿐만 아니라, 멀티모드 단말의 다중 네트워크 인터페이스간 전환을 효율적으로 제어하고 상위 계층에 대해 단일 인터페이스를 통한 통신이 유지되는 것처럼 보이게 하는 가상화(Virtualization) 기술의 개발이 필수적이라 할 수 있다. 즉, 다중 인터페이스간 핸드오버 수행 시 각 인터페이스에 직접적으로 연결되는 소켓(Socket) 기반의 서비스 세션이 끊어지는 문제는 종래의 IP 이동성 지원 기능만으로는 해결되지 않는 별도의 연구 주제로 대두되고 있다.

가상 인터페이스를 지원하기 위한 초기 기술은 각각의 물리적인 네트워크 인터페이스에 별도의 IP 주소를 할당하고, 물리 인터페이스와 TCP/UDP 계층간의 패킷 전달 과정에서 패킷 헤더 내의 IP 주소에 대해 물리 인터페이스에 할당된 주소 및 응용 계층이 사용하는 대표 IP 주소간 변환을 수행하는 방법 위주로 진행되어 왔다. 이러한 방법은 핸드오버 제어를 위해

IP 서브넷 별로 상이한 임시 IP 주소(예를 들어, Mobile IP의 Care-of-Address)를 할당하는 이동성 지원 방식과 연계하여 사용하기에 적합한 방식이다^[3]. 즉, MIP(Mobile IP)를 사용하는 단말은 통신 중인 상대 노드에게 알려지는 고정 IP 주소인 HoA(Home Address)를 유지하면서, 각 액세스 망(또는 IP 서브넷) 별로 할당되는 임시 IP 주소인 CoA(Care-of-Address)를 해당되는 물리 인터페이스에 직접 할당함으로써 위와 같은 가상화 방식을 자연스럽게 적용할 수 있다. 그러나 MIP와 같은 단말 기반의 IP 이동성 제어 방식은 특정 노드에 대한 트래픽 집중 및 단말의 IP 프로토콜 스택에 수정을 요구하는 문제 등으로 인해 실제 망에서 널리 활용되는 결과를 얻지는 못하고 있는 실정이다^[4].

현재 이동성 분야의 주요 추세 중 하나로 제어 기능을 망에 집중하여 단말에 요구되는 추가 기능을 최소화하는 네트워크 기반의 이동성 제어 방식을 들 수 있다^[4-7]. 네트워크 기반 이동성 제어 방식의 장점을 유지하기 위해서는 단말에서 다중 인터페이스 가상화를 위해 IP 주소를 변환하는 기존의 방식이 개선되어야 할 필요가 있다. 관련 연구로서 PMIP(Proxy MIP)과 연계하여 적용하기 위한 다중 네트워크 인터페이스 가상화 기술이 IETF의 주요 표준화 이슈로 대두되고 있으며^[8,9], 최근 PMIPv4/v6를 기반으로 하는 VI(Virtual Interface) 구조 및 구현 연구가 발표되고 있다^[10,11]. 그러나 현재까지의 VI 구현 연구는 OS에서 상이한 물리 인터페이스에 대해 동일 IP 주소 할당을 지원하지 않는 제약으로 인해 BBM(Break-Before-Make) 방식의 핸드오버 제어 기능만을 제공하는 단계에 머물러 있다^[11].

본 논문에서는 IETF의 PMIP과 같이 네트워크 기반의 이종 망간 IP 이동성 제어 기능을 지원하는 AIMS(Access Independent Mobile Service) 시스템^[7,8]을 기반으로 멀티모드 단말의 각 물리 인터페이스에 대해 상이한 IP 주소를 할당하지 않고 단일한 고정 IP 주소(HoA)만을 사용할 수 있게 해 주는 VI 설계 및 구현 내용을 제안한다. 제안되는 방식은 BBM 및 MBB(Make-Before-Break) 방식의 핸드오버를 모두 지원하며 다중 액세스 동시 접속(Multi-connection)을 통한 통신 효율의 향상 또한 지원할 수 있는 장점을 갖는다. 제안되는 VI 구조의 MBB 핸드오버 제어 방식은 이종 망간의 고품질 핸드오버(Seamless Handover) 지원에 적합한 기능을 제공하며 이동 단말이 고정된 IP 주소를 유지하면서 이종 망간 핸드오버 수행 후에도 지속적인 서비스 세션을 유지하는 환경을 구축하

는데 필수적인 기술이라 할 수 있다.

본 논문의 구성은 2장에서 관련 기술로서 AIMS, NDIS(Network Driver Interface Specifications) 등에 대해 소개하고, 3장에서 제안되는 VI의 구조 및 기능에 대해 상세히 설명한다. 4장에서는 구현 결과 및 향후 고려사항 등에 대해 논하고, 마지막으로 5장에서 결론을 맺는다.

II. 관련 기술

2.1 AIMS 시스템

AIMS(Access Independent Mobile Service) 기술은 다양한 액세스 기술을 포함하는 유무선 통합 네트워크 환경을 고려하여 단말의 이중 액세스 네트워크 간 핸드오버를 지원하는 네트워크 기반의 이동성 제어 기술이다^{4,5)}. AIMS 기술은 PMIP 기술과 유사하게 이동성 제어를 위한 단말 요구 기능을 최소화하면서 IEEE 802.21 MIH(Media Independent Handover) 기술을 이용하여 이동 단말의 고속 이동성을 지원할 수 있다. 이와 더불어, AIMS 기술은 특정 도메인 내의 지역적 이동성 관리 기술인 PMIP과 달리 이중 액세스 망들로 구성된 통합 환경에 직접 적용할 수 있는 제어 구조 및 프로토콜을 제공하며, 다양한 무선망이 중첩된 환경에서 사용자의 선호도 및 망 상태정보를 기반으로 품질 지향의 액세스 접속 제어를 수행하는 네트워크 선택 절차 기능을 제공할 수 있다. 그리고 AIMS는 안정된 이동성 제어 신호 전달을 위하여 제어 신호를 위한 경로와 데이터를 위한 경로가 분리된 구조를 가지며 이동성 지원을 위한 단말 위치 관리 기능과 핸드오버 제어 기능이 분리된 특징을 가진다^{4,5)}.

AIMS 시스템의 구성요소로는 코어 망 내에 위치하여 이동 노드(MN: Mobile Node)에 대한 위치 바인딩 정보(L2 ID, HoA, CoA 등)와 MN와 통신하고 있는 상대 노드(CN: Correspondent Node)에 대한 데이터 터널링 정보를 관리하는 이동성 제어 서버(MICS: Mobility Information Control Server), 각 액세스 망의 게이트웨이에 위치하여 MN에 대한 정보를 등록 및 관리하는 핸드오버 제어 에이전트(HCA: Handover Control Agent), MN에 대한 무선 링크 연결 및 접속 인지 기능을 제공하는 접속점(PoA: Point of Attachment) 그리고 액세스 망간을 이동하는 단말(MN)로 구성된다.

망 기반의 이동성 제어 기술인 AIMS 시스템에서 단말에 대해 이동성 지원을 위해 추가로 요구되는 기능은 없다. 그러나 단말이 다양한 액세스 망에 접속하

기 위해서는 기본적으로 다수의 유무선 물리 인터페이스들을 탑재한 멀티모드 단말을 가정하며, 단말은 초기 망 접속 시 망으로부터 할당 받은 하나의 고정 IP 주소(HoA)를 사용하면서 다수의 인터페이스를 사용하는 기술이 필요하다. 또한 멀티모드 단말이 접속 망을 변경하여 단말의 물리 인터페이스간 핸드오버를 수행하는 경우, 단말의 고정된 IP 주소를 유지하면서 단말이 통신 중인 세션을 끊김 없이 유지하는 기능이 제공되어야 한다. 그리고 다수의 액세스 망이 중첩된 환경에서 단말 또는 사용자 요구사항에 적합한 최적의 망에 접속하기 위해서는 MIH와 같은 서비스를 기반으로 주변 액세스 망을 감지 및 탐색하여 최적의 네트워크에 접속할 수 있는 기능이 필요하다.

2.2 NDIS

NDIS(Network Driver Interface Specification)는 마이크로소프트와 3COM이 공동으로 개발한 네트워크 장치 인터페이스 표준 규격으로, 윈도우 운영체제와 네트워크 드라이버 스택들 간의 통신 및 서비스를 제공하는 모듈들로 구성되어 있는 라이브러리 형태의 드라이버이다¹²⁻¹⁴⁾. NDIS는 계층화된 네트워크 드라이버들 간의 표준 인터페이스를 제공하고, 네트워크 드라이버 함수들에 대한 포인터, 핸들, 연결 블럭에 대한 파라미터, 시스템 값들에 대한 상태정보 및 파라미터를 유지하고 관리한다. 이를 위하여 NDIS는 미니 포트(Miniport) 드라이버, 인터미디엇(Intermediate) 드라이버 및 프로토콜(Protocol) 드라이버 등의 네트워크 드라이버들을 지원하며, NDIS와 네트워크 드라이버의 관계는 그림 1과 같다.

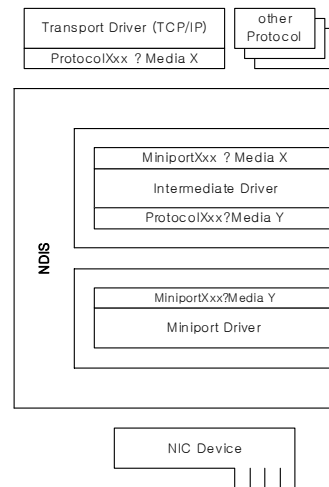


그림 1. NDIS와 네트워크 드라이버의 관계

2.2.1 NDIS 미니포트 드라이버

NDIS 미니포트 드라이버는 하위 레벨의 네트워크 인터페이스 카드(NIC: Network Interface Card)를 관리하는 기능을 수행하며, NIC을 통하여 데이터를 송수신하는 기능을 수행한다. 또한, NDIS 미니포트 드라이버는 상위 레벨의 드라이버들(즉, 인터미디엇 드라이버 또는 전송계층 드라이버)과의 인터페이스를 가진다. 미니포트 드라이버는 자신의 물리적 네트워크 카드 또는 상위레벨 드라이버와 NDIS 라이브러리를 통해 통신한다. NDIS 라이브러리는 미니포트 드라이버에게 NdisXxx 형태의 함수들에 대한 인터페이스를 제공하며, 이 함수들은 미니포트 드라이버가 호출해야 하는 운영체제 함수 전부를 캡슐화 한 것이다. 또한, 미니포트 드라이버는 NDIS 또는 미니포트로 접근하는 상위 레벨의 드라이버가 호출해야 하는 MiportXxx 형태를 띤 함수들에 대한 인터페이스를 제공한다.

2.2.2 NDIS 인터미디엇 드라이버

NDIS 인터미디엇 드라이버는 상위 레벨의 프로토콜 드라이버와 하위 레벨의 미니포트 드라이버의 사이에 위치하며, 상하위 레벨의 드라이버들과 통신을 수행하기 위한 두 가지 인터페이스를 가진다. 첫 번째 인터페이스는 인터미디엇 드라이버의 하단부에 위치하며, NDIS는 ProtocolXxx 함수를 호출하여 하위 레벨의 미니포트 드라이버가 요청하는 일을 처리한다. 즉, 인터미디엇 드라이버는 이 인터페이스를 통해서 미니포트 드라이버들에게 프로토콜 드라이버처럼 보이게 동작한다. 두 번째 인터페이스는 인터미디엇 드라이버의 상단에 위치하며, NDIS는 MiniportXxx 함수를 호출하여 상위 레벨의 프로토콜 드라이버가 요청하는 일을 처리한다. 즉, 인터미디엇 드라이버는 프로토콜 드라이버에게 하나 이상의 가상 어댑터를 노출함으로써 미니포트 드라이버처럼 보이게 동작한다. 인터미디엇 드라이버들은 일반적으로 패킷을 상이한 네트워크 미디어간에 변환하여 전송되거나 특정 패킷들을 필터링, 또는 하나 이상의 NIC과 관련하여 패킷 전송의 균형을 조절하는데 사용된다.

2.2.3 NDIS 프로토콜 드라이버

NDIS 드라이버 계층상에서 최상위에 존재하는 네트워크 프로토콜 드라이버는 TCP/IP 혹은 IPX/SPX 스택과 같은 전송 프로토콜 스택을 실행하는 전송계층 드라이버 내에서 가장 낮은 레벨 드라이버로서 사용된다. 프로토콜 드라이버는 어플리케이션으로부터 전송되는 데이터 패킷에 복사하고, 이러한 패킷들을

NDIS 함수들을 호출하여 하위 레벨의 드라이버에게 전송한다. 또한, 프로토콜 드라이버는 하위 레벨의 드라이버들(즉, 인터미디엇 또는 미니포트 드라이버)로부터 들어오는 패킷들을 수신하기 위해 프로토콜 인터페이스를 제공하며, 수신된 데이터 패킷들을 적당한 클라이언트 어플리케이션에게 전송한다.

Ⅲ. 멀티모드 단말의 가상 인터페이스 기술

본 장에서는 AIMS 시스템에서 다수의 네트워크 인터페이스를 가지는 멀티모드 단말의 가상 인터페이스를 NDIS 기술 기반으로 구현하기 위한 구조, 기능 및 동작 방식에 대하여 기술한다.

3.1 가상 인터페이스 구조 및 기능

AIMS 시스템에서 멀티모드 단말은 초기 접속 시에 망으로부터 할당 받은 하나의 HoA를 고정적으로 사용하면서 다수의 인터페이스를 사용한다. 또한 멀티모드 단말이 접속 망을 변경하여 상이한 물리 인터페이스간의 핸드오버를 수행하는 경우, 단말의 고정된 IP 주소를 유지하면서 통신 중인 세션을 끊김없이 유지하는 기능이 제공되어야 한다. 이를 위해 멀티모드 단말에서 다수의 물리 인터페이스들의 상위에 위치하는 가상 인터페이스(VI: Virtual Interface)를 생성하여 상위 계층 및 커널에 적용하는 방안을 제시한다. 멀티모드 단말의 VI는 NDIS 기술을 기반으로 하며, 그림 2와 같은 구조를 지닌다. VI 드라이버는 구조적으로 NDIS 네트워크 드라이버 스택 상에서 미니포트 드라이버와 프로토콜 드라이버 사이에 위치하는 인터미디엇 드라이버에 해당되며, 기능상으로 다수의 물리적 네트워크 인터페이스들과 1xN 관계를 유지하

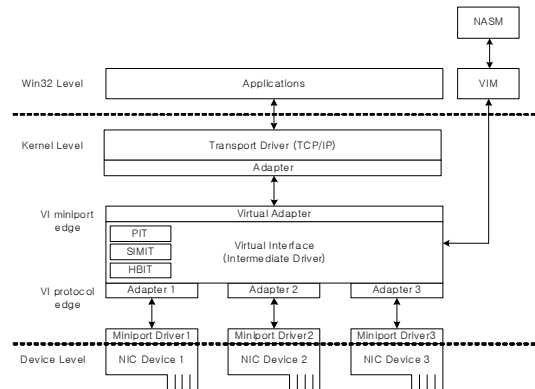


그림 2. 가상 인터페이스 구조도

는 분배기의 역할을 수행한다. 이에 따라 VI는 상위 레벨의 프로토콜 드라이버(TCP/IP)에 대해서는 하나의 어댑터로 보이며, 하위 레벨의 다수의 물리적 인터페이스들의 어댑터를 관리하는 역할을 수행한다. 즉, VI는 기존의 NDIS 드라이버 스택에 연결되기 위해서 상위 레벨의 프로토콜 드라이버에 대하여 가상으로 미니포트 드라이버를 생성하며, 미니포트 기능을 위하여 각종 핸들러를 구성하고 NDIS에서 제공하는 함수 NdisIMRegisterLayeredMiniport를 사용하여 등록한다. 또한 VI는 하위 레벨에 실제로 존재하는 다수의 미니포트 드라이버들과 인터페이스를 연결하는 프로토콜 역할을 수행하며, NDIS에서 제공하는 함수 NdisRegisterProtocol를 사용하여 자신의 기능을 등록한다. 그리고 자신이 인터미디어트 드라이버임을 NDIS에 알리기 위해 NdisIMAssociateMiniport 함수를 호출하여 인터미디어트 드라이버로 등록한다.

VI는 가상 인터페이스 관리(VIM: Virtual Interface Management) 모듈에 의해서 제어되며, VIM 모듈은 단말의 망 접속 및 선택 관리(NASM: Network Access and Selection Management) 모듈과 연계하여 물리 인터페이스 관리, 가상 인터페이스에 대한 초기화, 네트워크 연결 관리, 핸드오버 및 인터페이스 전환 기능 등을 제어 및 관리한다. 이를 위하여 VIM 모듈은 NASM 모듈이 VI에 대한 통신 및 명령을 처리하기 위한 인터페이스(API)를 제공한다.

VI는 하위 레벨의 다수의 물리 인터페이스들에 대한 정보를 관리하기 위해서 물리 인터페이스 테이블(PIT: Physical Interface Table)을 유지하며, 특정 세션이 사용하는 특정 인터페이스에 대한 매핑 정보를 관리하기 위해서 세션 및 인터페이스 매핑 정보 테이블(SIMIT: Session & Interface Mapping Information Table)을 가진다. 또한, VI는 NASM 모듈의 정책에 따라 핸드오버 및 패킷 버퍼링 동작 방식을 설정 및 저장하는 핸드오버 및 버퍼링 정보 테이블(HBIT: Handover & Buffer Information Table)을 관리한다.

3.2 가상 인터페이스 동작 방식

3.2.1 가상 인터페이스 초기화 및 망 접속 방식

단말의 VI 초기화 및 초기 망 접속 과정은 NASM 모듈의 요청을 통하여 VIM 모듈이 수행한다(그림 3). VIM 모듈은 단말의 VI 및 물리 네트워크 인터페이스들에 대한 정보를 VIM 모듈 내에 관리하는 인터페이스 정보 테이블(IIT: Interface Information Table)에 등록한다. 물리 인터페이스에 대한 정보는 VI 모듈이

제공하는 정보를 바탕으로 구성될 수 있다. 이 때 IIT에 저장되는 주요 정보는 인터페이스에 대한 기본 정보와 인터페이스를 통한 망 접속 가능 및 상태 정보 등으로 구성된다. VIM 모듈은 이러한 인터페이스 정보들을 단말의 NASM 모듈에게 전달함으로써 초기 접속 망 및 인터페이스 선택 정보를 요청한다. NASM 모듈은 단말의 이러한 인터페이스 정보들을 저장하고 이러한 정보를 바탕으로 초기 접속 인터페이스 우선순위 리스트를 구성하여 가상 인터페이스 관리 모듈에게 전달한다.

VIM 모듈은 인터페이스 우선순위 리스트에 따라 각각의 인터페이스들을 통하여 망 접속을 순차적으로 시도한다. 초기 망 접속이 완료되면 VIM 모듈은 초기 접속한 망에 대하여 세션 정보 테이블(SIT: Session Information Table) 내에 디폴트(Default) 인터페이스를 설정하고, VI에게 디폴트 물리 네트워크 인터페이스와의 연결 설정을 요청한다. VI는 VIM 모듈이 제공하는 초기 접속 인터페이스 정보에 따라 해당 물리 네트워크 인터페이스와 연결을 수행하고, 이 물리 인터페이스를 디폴트(Default) 인터페이스로 SIMIT 내에 설정한다. VI는 이에 대한 결과를 가상 인터페이스 관리 모듈로 전송하고, VIM 모듈은 NASM 모듈에게 이에 대한 결과를 보고함으로써 인터페이스 초기화 및 망 접속 절차를 완료한다.

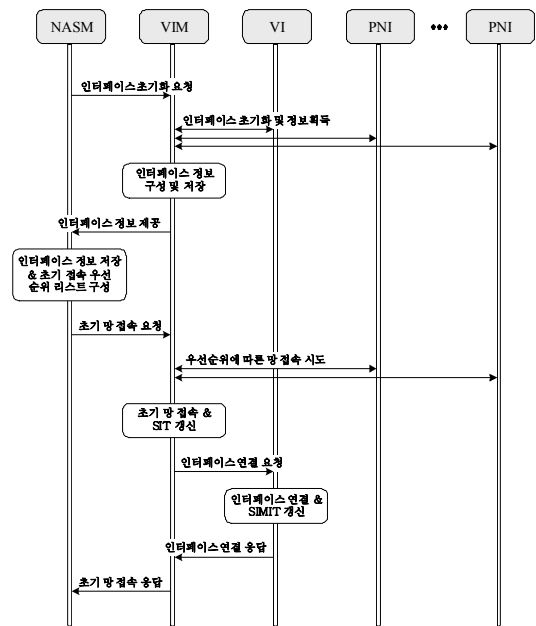


그림 3. 인터페이스 초기화 및 망 접속 절차

3.2.2 가상 인터페이스 기반 핸드오버 지원 방식

NASM 모듈은 각 세션에 적합한 인터페이스 우선 순위 리스트를 구성하여 VIM 모듈에게 전달함으로써 핸드오버를 요청한다(그림 4). 이 때 NASM 모듈은 사전 설정된 정책에 따라 MBB(Make-Before-Break), BBM(Break-Before-Make), 다중 접속(Multi-connect) 등과 같은 핸드오버 지원 모드와 버퍼링 모드를 설정할 수 있다. VIM 모듈은 각 세션에 대해 구성된 인터페이스 우선순위 리스트에 따라 각각의 인터페이스들을 통하여 망 접속을 순차적으로 시도한다. 인터페이스 우선순위 리스트에 따른 망 접속의 선택 및 접속 완료가 수행되면, VIM 모듈은 자신의 SIT 내에 해당 세션에 대한 인터페이스 변경 정보를 갱신한다. VIM 모듈은 VI에 이러한 변경 정보를 제공하고, VI는 해당 세션에 대하여 접속된 물리 네트워크 인터페이스와 VI간의 연결을 설정한 후 SIMIT의 갱신을 수행한다. 그림 4는 MBB 핸드오버 모드를 지원하는 방식을 보여준다. 이에 따라 VIM 모듈은 새로운 망에 접속을 수행한 이후에 이전 망에 대한 접속을 종료하게 된다.

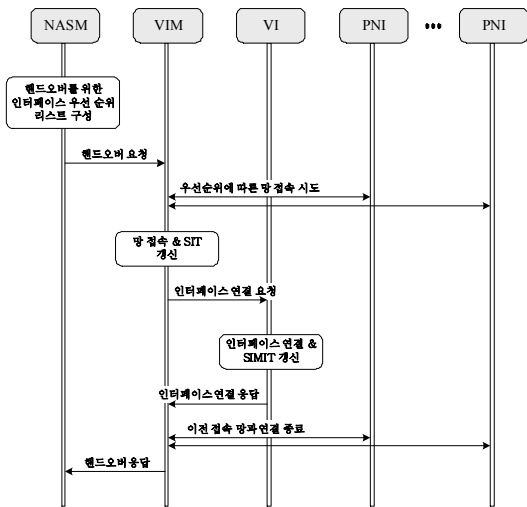


그림 4. 가상 인터페이스 기반의 MBB 방식 핸드오버 수행 절차

3.2.3 VI 기반의 데이터 전송 방식

NASM 모듈은 사전에 설정된 사용자의 정책에 따라 다수의 물리 인터페이스를 유동적으로 사용할 수 있다. 즉, 단말은 한 번에 하나의 물리 네트워크 인터페이스만을 사용하거나 또는 두 개 이상의 물리 네트워크 인터페이스를 동시에 사용할 수 있다. 단말이 한 번에 하나의 인터페이스만을 사용하는 경우에는 가상의 인터페이스에 연결된 물리 네트워크 인터페이스는

디폴트 인터페이스로 등록이 되어 모든 데이터들을 송수신하는데 사용된다. 만일 단말이 다수의 물리적 인터페이스를 동시에 사용하는 경우에는 NASM 모듈이 각각의 세션에 대한 특성을 고려하여 세션에 적합한 액세스 망을 선택하고 해당 물리 네트워크 인터페이스를 선택하여 해당 세션들에 대한 데이터들을 송수신한다. 이를 위해서 세션과 인터페이스에 대한 매핑 정보는 VIM 모듈의 SIT와 SIMIT에 등록된다.

IV. 기능 시험 및 분석

본 장에서는 AIMS 시스템의 테스트베드 환경에서 다수의 인터페이스를 가지는 멀티모드 단말의 VI에 대한 기능 시험 및 분석을 기술한다.

4.1 기능 시험 환경

그림 5는 AIMS 시스템의 테스트베드 구성도를 보여준다. AIMS 시스템 테스트베드에서는 다양한 무선 액세스 망을 실험실 내에 구축하는 것이 어려우므로, 무선랜(WLAN)의 AP(Access Point)를 이용하여 3G(HSDPA) 망의 기지국(BS: Base Station) 및 WiBro 망의 RAS(Radio Access Station)를 대체하는 에뮬레이션 시스템을 개발하여 사용한다. 이러한 핸드오버 에뮬레이션 시스템(HANE: Heterogeneous Access Network Emulator)은 각각의 액세스 망 특성들(무선 링크 대역폭, 패킷 손실을 및 접속 지연 시간 등)을 반영하여 실제 무선 통신 환경과 유사한 실험 환경을 제공할 수 있다(그림 6). 그리고 HANE 시스템은 사전에 준비된 이종 네트워크간의 핸드오버 시나리오에 따라 이동 단말들에 대하여 주변의 AP 시그널 파워를 조정함으로써 이동 단말이 실제로 핸드오버를 수행하는 것과 같은 효과를 제공하는 기능을 수행한다. 멀티모드 단말은 3개의 무선랜 카드를 탑재하여 각각의

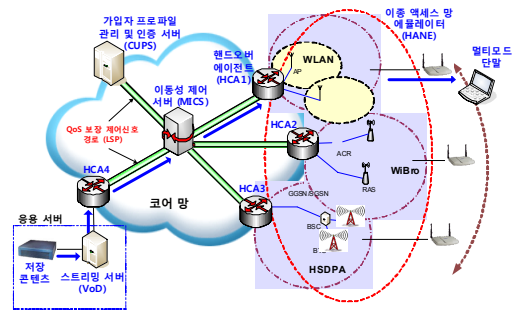


그림 5. AIMS 시스템의 테스트베드 구성도

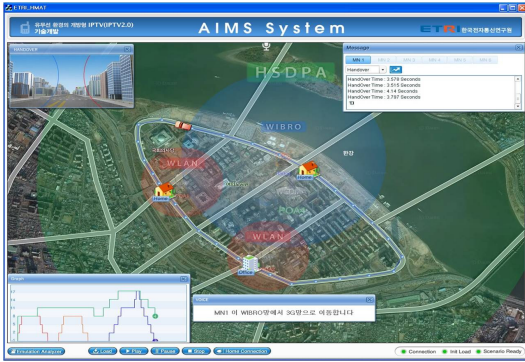


그림 6. 핸드오버 에뮬레이션 및 모니터링 시스템

무선 인터페이스가 WLAN, WiBro 그리고 HSDPA 인터페이스로 동작하는 것을 가정하였다.

그림 7은 VI의 성능 시험을 위해 멀티모드 단말에 구현된 VI 테스트 프로그램을 보여준다. 멀티모드 단말은 HANE 시스템이 제공하는 핸드오버 시나리오 또는 VI 테스트 프로그램을 통해 입력되는 사용자 명령에 따라 각각의 액세스 망에 대하여 물리 인터페이스간 핸드오버를 수행한다. VI 테스트 프로그램은 가상 인터페이스 및 물리 인터페이스의 동작 여부를 확인할 수 있도록 VI에 대한 GUID 정보, IP 주소, MAC 정보, 게이트웨이 IP 주소 등에 대한 정보를 설정 및 저장하는 기능을 제공한다. VI 테스트 프로그램은 VI의 하부에 존재하는 다수의 물리 인터페이스들에 대한 정보와 SIMIT 정보를 저장 및 설정하는 기능을 제공한다. 또한 NASM 모듈로부터 사전에 설정된 사용자의 정책에 따라 핸드오버 지원모드, 버퍼링 지원모드 그리고 인터페이스 지원모드 등을 설정 및 저장하는 기능을 제공하며, 사용자의 명령으로 멀티모드 단말의 핸드오버를 수행하도록 하는 물리 인터페이스간 핸드오버 기능을 지원한다.

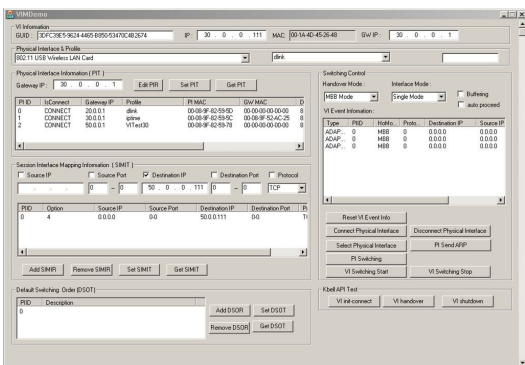


그림 7. 가상 인터페이스 테스트 프로그램

4.2 기능 시험

멀티모드 단말에 대한 VI 성능 실험을 위하여 AIMS 시스템 테스트베드 환경에서 멀티모드 단말에 구현된 VI 테스트 프로그램을 사용하여 VI 초기화 및 망 접속 연결, 물리 인터페이스간의 핸드오버 그리고 VI 종료 테스트를 수행하였다. VI 테스트 프로그램에서 지원되는 이러한 초기 접속, 망 접속, 핸드오버 등의 기능들에 대한 자세한 동작 방법은 3.2절을 참조하며, 이러한 기능에 대한 자세한 구현 방법들은 추후 발표될 논문 및 보고서에서 다룬다.

VI에서 NASM 모듈에게 제공하는 VI-Interface-Init API를 사용하여 VI를 초기화하는 시험을 수행하였다. VI는 NASM 모듈이 제공하는 정보(핸드오버 지원모드, 버퍼링 지원모드, 인터페이스 지원모드 등)에 따라 초기화 설정을 수행하고, VI-Init-Connect API를 사용하여 초기 망 접속 요청을 수행한다. 초기화 과정이 완료된 이후에 VI 테스트 프로그램을 이용하여 설정 결과를 확인하였고, 초기 접속된 망과 해당 물리 인터페이스를 통하여 Ping 테스트와 스트리밍 데이터 수신 테스트를 수행함으로써 가상 인터페이스의 기능 및 동작의 정확성을 확인할 수 있었다.

VI의 물리 인터페이스간 핸드오버 시험을 위해 VI-Handover API를 사용하여 인터페이스간의 전환 시험을 수행하였다. VI는 NASM 모듈이 제공하는 정보(물리 인터페이스 정보, 접속 망 정보 등)에 따라 물리 인터페이스 변경 및 VI와 물리 인터페이스에 대한 연결 설정 변경 등을 수행한다. 인터페이스간의 핸드오버 과정이 완료된 이후에 VI 테스트 프로그램을 통하여 VI 연결 설정 결과 및 물리 인터페이스 전환 결과를 확인하였고, 초기화 과정과 동일하게 Ping 테스트와 스트리밍 데이터 수신 테스트를 통해 전환된 물리 인터페이스로 지속적인 패킷을 송수신하는 것을 확인하였다. 멀티모드 단말이 인터페이스간의 핸드오버를 수행하더라도 현재 단말이 통신 중인 세션의 끊김없는 연결성을 확인할 수 있었고, 핸드오버 성능 역시 기존의 AIMS 시스템에서 VI를 사용하지 않는 동종 망간 핸드오버 성능과 유사함을 확인할 수 있었다.

마지막으로 VI-Shutdown API를 사용하여 VI의 종료 시험을 수행하였다. VI는 단말의 모든 물리적 인터페이스에 대한 연결 해제를 수행하며, VI와 물리 인터페이스간의 연결 설정을 해제한 후 VI가 초기화된다. 이러한 결과는 VI 테스트 프로그램으로 확인할 수 있었다.

4.3 고려 사항

멀티 모드 단말에서 가상 인터페이스를 설계하고 구현함에 있어서 고려해야 할 몇 가지 사항들은 다음과 같다. 먼저, 인터페이스 관리 모듈간의 충돌이 발생 할 수 있다. 즉, 본 논문에서 멀티모드 단말은 3개의 무선랜 카드를 사용하여 인터페이스간의 핸드오버를 수행한다. 이 때, 무선랜 AP에 대한 접속 및 해지를 위해 윈도우 OS가 제공하는 WZC(Wireless Zero Configuration) 클라이언트를 통한 API를 사용할 경우 망 접속 및 해지 시간의 지연, 다른 벤더가 제공하는 클라이언트와의 충돌, 그리고 WZC가 자동으로 망 접속/해지를 수행하는 것에 기인한 연결 불안정성 등 문제점을 유발할 수 있다. 이를 해결하기 위해 본 논문에서는 WZC 서비스를 사용하지 않은 상태에서 VI가 직접 망 접속/해지에 대한 기능을 수행하도록 수정하여 구현하였다.

두 번째 고려사항으로, 서로 다른 다수의 물리적 네트워크 인터페이스를 관리하는 통합 인터페이스 모듈 개발이 필요하다. 일반적으로 물리 인터페이스를 제어 및 관리하는 기능은 하드웨어 벤더들이 별도의 프로토콜 드라이버를 개발하고 커넥션 매니저(Connection Manager)를 만들어 수행한다. 현재 구현된 멀티모드 단말은 3개 무선랜 카드를 임의로 WLAN, WiBro 그리고 HSDPA로 간주하고 있지만, 실제의 멀티모드 단말의 경우 다수의 물리적 인터페이스의 하드웨어 구성이 모두 다르므로 WLAN, WiBro 그리고 HSDPA 등 서로 다른 모든 물리적 인터페이스들을 제어 및 관리하기 위한 통합 연결 관리자 기술 개발이 추가로 필요하다.

마지막으로, 다수의 인터페이스를 사용하는 멀티모드 단말에서 한정된 전원을 효율적으로 사용하는 방안이 필요하다. 이를 위해서는 3개의 물리 인터페이스에 대한 전원 관리 및 제어 기능이 필요하며, 이를 통해 물리 인터페이스의 전원 상태 처리 과정의 단계인 D0(완전히 켜짐), D1(거의 켜짐), D2(거의 꺼짐), D3(완전히 꺼짐) 등 상태를 원활히 제어해야 할 필요가 있다. 이러한 기능의 구현에는 우선적으로 하드웨어가 이러한 기능을 지원해야 하며, 다수의 물리 인터페이스 전원을 제어하는 통합 기능을 구현해야 할 필요가 있다. 이 기능은 커넥션 매니저 프로그램의 명령을 프로토콜 드라이버가 받아 미니포트를 제어해야 하는 과정으로 구성될 수 있다. 이때 프로토콜 드라이버는 이 과정을 수행하기 위한 DeviceIoControlCode를 구현해야 하므로, 실제 망 구성이 WLAN, Wibro, HSDPA 인 경우 해당 하드웨어 벤더가 제공하는 규

격과 전원 관련 처리정책 및 인터페이스를 제공 받아야 한다.

V. 결론 및 향후 연구

본 논문에서는 다양한 유무선 액세스 망을 수용하는 유무선 방통융합 서비스를 실현하기 위한 핵심 기술로서 멀티모드 단말의 이중 액세스 망간 끊임없는 이동성을 지원하는 네트워크 인터페이스 가상화 기술의 구현 내용을 제안하고 있다. 제안되는 가상 인터페이스(VI) 기술은 각 물리 인터페이스에 대해 상이한 IP 주소를 할당하고 패킷 전달 시 이들 주소간의 변환을 수행하는 기존의 방식과 달리, 고정된 하나의 단일 IP 주소만을 이용하여 다수의 물리 인터페이스들을 가상화하는 기능을 제공한다. 이는 PMIPv4/v6, AIMS 등 단말 기능 수정을 요구하지 않는 네트워크 기반의 IP 이동성 제어 기법에 적용하기가 용이한 장점을 제공하며, 특히 제안 기법은 기존의 OS가 단일 단말 내의 상이한 물리 인터페이스에 대해 동일 IP 주소를 할당할 수 없도록 하는 제약사항을 극복하여 MBB(Make-Before-Break), BBM(Break-Before-Make), 다중 접속(Multi-connection) 등과 같은 다양한 핸드오버 방식을 모두 지원할 수 있는 장점을 갖는다.

제안된 VI 구조 및 기능은 NDIS 라이브러리를 기반으로 TCP/UDP 프로토콜 계층과 하부 물리 인터페이스 계층을 연결하는 가상 드라이버 형태로 구현되었으며, 이중 망간 IP 이동성 지원을 위해 개발된 AIMS 시스템과 연동하여 이중 네트워크 인터페이스를 이용한 핸드오버(Vertical Handover) 제어를 위해 적용되고 있다. 기능 시험 및 분석을 위한 테스트베드를 구축하였으며, 테스트 프로그램의 구현 및 운영을 통해 VI 초기화, 물리 인터페이스 간 전환, 접속 종료 등 기능에 대한 동작의 정확성을 확인할 수 있었다. 향후 연구로는 다수의 물리 인터페이스 제어 시의 전력 소모 최적화 기술 및 이중 액세스 망 접속을 효과적으로 제어하기 위한 커넥션 매니저 개발을 수행하고자 한다.

참 고 문 헌

- [1] 이경희, 윤장우, 류원, 김봉태, "IPTV2.0 서비스를 위한 네트워킹 기술", 대한임베디드공학회 논문지, 3권 4호, pp.218-228, 2008년 12월.
- [2] 윤장우, 이현우, 류원, 김봉태, "IPTV 서비스 및 기술 진화 방향", 한국통신학회지(정보와통신),

- 25권 8호, pp.3-11, 2008년 8월.
- [3] C. Perkins, "IP Mobility Support for IPv4," RFC3344 on IETF, Aug. 2002.
 - [4] 류원, 이경희, 김봉태, 박성수, "유무선 통합망에서의 품질보장형 이동성 관리 기술 구현 연구", SK Telecommunications Review, 18권 4호, pp.624-638, 2008년 8월.
 - [5] Myoungju Yu et al., "A New Mechanism for Seamless Mobility Based on MPLS LSP in BcN," IEICE Transactions on Communications, Vol. E91-B, No.2, Feb. 2008.
 - [6] S. Gundavelli et al., "Proxy Mobile IPv6," RFC5213 on IETF, Aug. 2008.
 - [7] R. Wakikawa and S. Gundavelli, "IPv4 Support for Proxy Mobile IPv6," Internet Draft on IETF, Sept. 2009.
 - [8] V. Devarapalli et al., "Multiple Interface Support with Proxy Mobile IPv6," Internet Draft on IETF, Sept. 2009.
 - [9] H. Yokota et al., "Inter-Technology Handoff Support in Mobile Node for Proxy Mobile IPv6," Internet Draft on IETF, Oct. 2009.
 - [10] 최우진, 김성진, 유홍렬, 이성춘, "PMIPv4 기반 WiBro-HSDPA 이중망간 Seamless 핸드오버기술 개발", 한국통신학회지(정보와통신), 26권 2호, pp.3-8, 2009년 2월.
 - [11] 김경아, 임재강, 한연희, "PMIPv4 기반의 WiBro-HSDPA간 이동성 지원방법", KNOM Review, Vol. 11, No.1, July 2008.
 - [12] Walter Oney, "Programming the Microsoft Windows Driver Model," 2nd edition, Microsoft Press, Redmond, WA, 2002.
 - [13] Anthony Jones and Jim Ohlund, "Network Programming for Microsoft Windows," 2nd edition, Microsoft Press, Redmond, WA, 2002.
 - [14] Paul Goransson and Raymond Greenlaw, "Secure Roaming in 802.11 Networks," 1st edition, Newnes, 2007.

이 경 희 (Kyounghee Lee)

정회원



1999년 광운대학교 전자계산학과 학사

2000년 한국정보통신대학교 공학부 석사

2006년 한국정보통신대학교 공학부 박사

2006년~현재 한국전자통신연

구원 방송통신융합연구부 IPTV연구부 융합서비스네트워크연구팀 선임연구원

<관심분야> 이동성 관리, IPTV, 인터넷 QoS, NGN, 미래 네트워크

이 성 근 (Seongkeun Lee)

정회원



2004년 고려대학교 전기전자전공과공학부 학사

2009년 고려대학교 전자컴퓨터공학과 박사

2009년~현재 한국전자통신연

구원 방송통신융합연구부

IPTV연구부 융합서비스네트

워크연구팀 연구원

<관심분야> 유무선 IP 네트워크, 광전송 네트워크, IPTV, 멀티미디어 서비스

이 은 준 (Eunjun Rhee)

정회원



1992년 한국항공대학교 항공통신정보공학과 학사

1995년 한국항공대학교 정보통신공학과 석사

1995년~2000년 모다정보통신과장

2000년~현재 한국전자통신연

구원 방송통신융합연구부 IPTV연구부 융합서비스네트워크연구팀 선임연구원

<관심분야> 데이터 통신, 이동성, IPTV, NGN

조 경 섭 (Kyoung-seob Cho)

정회원



1983년 서울대학교 수학교육과 학사
1985년 서울대학교 수학과 석사
2000년~현재 한국전자통신연구원 방송통신융합연구부문 IPTV연구부 융합서비스네트워킹연구팀 책임연구원

<관심분야> IPTV, 서비스 제어기술, 통신망 연동, NGN

류 원 (Ryu Won)

중신회원



1983년 부산대학교 계산통계학과 학사
1988년 서울대학교 계산통계학과 석사
2002년 성균관대학교 정보공학과 박사
1989년~현재 한국전자통신연구원 방송통신융합연구부문 IPTV연구부 부장

<관심분야> IPTV, 이동성, 이종망간 핸드오버, 유무선 연동, BcN

이 현 우 (Hyun-woo Lee)

중신회원



1993년 한국항공대학교 항공전자공학과 학사
1995년 한국항공대학교 정보통신공학과 석사
2005년 한국항공대학교 정보통신공학과 박사
1995년~현재 한국전자통신연구원 방송통신융합연구부문 IPTV연구부 융합서비스네트워킹연구팀 팀장

<관심분야> 서비스 제어기술, 통신망 연동, 트래픽 혼잡제어

홍 승 필 (Seng-phil Hong)

중신회원



1993년 Indiana State Univ. 학사
1994년 Ball State Univ. 석사
1997년 Illinois Institute of Technology 박사수료
2002년 한국정보통신대학교 박사

1997년~2005년 LG CNS
2005년~현재 성신여자대학교 IT학부 조교수
<관심분야> 접근제어, 통합인증, 정보보호 아키텍처, 유비쿼터스 보안, 프라이버시 보호