



Developing a Wireless Interworking Platform using MIH in SDN Environment

Wan-Jik Lee, Jung Yun Pil, Seok-Yeol Heo*

Department of IT Engineering & Application, Pusan National University

A B S T R A C T

SDN-based network control technology has attracted much attention on wired core networks, and therefore all control and resource management on the network is expected to be integrated into the SDN technology in the future. On the contrary, the IEEE 802.21 standard, Media Independent Handover (MIH), includes a technique for managing wireless resource information to support media independent handover services. By combining these two technologies, SDN controller, managing only the resource and control information of the existing wired network, also has the advantage of being able to manage information and mobility of wired/wireless network resources in an integrated manner by gathering information on MIH that manages wireless network resources and mobility related information together. In this paper, we propose an interworking scheme of MIH standard protocol in SDN environment and build an SDN-based MIH interworking platform. We also developed a simple SDN-based wireless resource management application on the platform and tested the performance of the platform constructed in this paper. The wireless interworking platform of the SDN environment developed in this paper can be used to test the wireless bandwidth management of SDN based APs and the load management function of APs. And it can also be utilized for developing terminal mobility technology based on the allocated resource information.

© 2017 KKITS All rights reserved

KEYWORDS : SDN, Openflow, IEEE 802.21, MIH, MIS, Interworking, Platform, Wireless resource management

ARTICLE INFO: Received 24 January 2017, Revised 3 February 2017, Accepted 10 February 2017.

*Corresponding author is with the Department of IT Engineering & Application, Pusan National University, Cheonghak-ri, Samlanggin-eup Miryang, Gyungnam,

50463, KOREA.

E-mail address: syheo@pusan.ac.kr

1. 서론

현재의 인터넷 구조는 스위치와 라우터를 기반으로 하는 계층적인 구조를 가지고 있기 때문에 ‘클라이언트-서버’ 컴퓨팅을 기반으로 하는 작업들을 효율적으로 처리하는 데는 적합하지만, 최근 폭발적으로 증가하고 있는 모바일 서비스와 클라우드 기반의 가상화 서비스와 같은 환경에는 적합하지 않다. 이러한 문제를 해결하기 위한 기술로서 등장한 SDN(Software Defined Networking)은 네트워크 제어 기능(control plane)이 물리적 네트워크와 분리되어 있어서 네트워크 장치에 대해서 독립적이고 효율적인 관리가 가능하다[1][2].

최근 들어 전통적인 PC 시장이 포화상태에 이른 것과는 달리 태블릿 PC나 스마트폰을 비롯한 스마트 디바이스 시장이 지속적으로 증가하고 있다. 이러한 스마트 디바이스의 증가와 함께 이들이 사용하는 무선 데이터 사용량도 함께 폭발적으로 증가하고 있다. 많은 양으로 증가하는 무선 데이터를 효과적으로 담당하기 위해서는 3G/4G 셀룰러 망의 확충과 무선 랜의 안정성 및 협소한 서비스 영역 문제가 해결되어야 한다. 그러나 제한적인 주파수 대역 및 서비스 영역의 중첩 문제가 발생하면서 서로 다른 특성을 갖는 다양한 무선 접속망간에 끊임 없는 통신을 지원하는 개방형 무선 이종 망 연동 기술에 대한 관심이 증가하고 있다[3][4].

따라서 사용자의 요구나 네트워크의 상황에 따라 다양한 접속 망(셀룰러, 무선 랜, WPAN(Wireless Personal Area Network) 등)들을 사용자의 멀티모드 단말과 끊임 없이 최적으로 연결하는 연동기술이 필요하다. 이를 위해서는 이미 존재하는 다양한 무선 인터페이스뿐만 아니라 새로 출현할 무선 인터페이스까지도 쉽게 수용할 수 있는 개방형 미디어 독립적인 연동 기술의 표준화가 요구된다.

IEEE 802.21 WG에서는 2008년 미디어 독립적인 핸드오버(Media Independent Handover:MIH) 지원을 위한 프레임워크와 프로토콜 표준을 완료하였으며, 2013년부터는 미디어 독립적인 무선 자원 관리 기능을 강화한 802.21 Revision (Media Independent Service Framework)과 IEEE 802.21.1 Media Independent Services(MIS) 표준화를 진행하고 있다 [5]. 또한 유선 코어 네트워크상에서는 SDN 기반의 네트워크 제어 기술이 표준화되고, 성숙화 단계에 놓여 있음에 따라, SDN 기반 환경에서 MIH 기반의 무선 이종 망의 연동 및 제어 기술을 융합하는 연구의 필요성이 높아지고 있다.

이러한 두 가지 기술을 결합하면, 기존의 유선 네트워크의 자원 및 제어 정보들만 관리하는 SDN Controller가 무선 네트워크의 자원 및 이동성 관련 내용을 관리하는 MIH의 정보를 같이 수집하여 유/무선 네트워크 자원 정보나 이동성을 통합적으로 관리할 수 있는 장점을 가진다.

이를 위해, 본 논문에서는 SDN 프로토콜 상에서 IEEE 802.11 MIH 표준과 연동 방안을 구현하여 SDN 기반의 MIH 연동 플랫폼을 구축하였다. 또한 플랫폼 상에서 간단한 SDN 기반의 무선 자원 관리 응용 프로그램을 작성하여, 본 논문에서 구축한 플랫폼의 수행 기능을 시험하였다.

본 논문의 구성은 다음과 같다. 우선 2장에서는 본 논문의 주제인 SDN, MIH 연동 테스트베드 구축에 관련한 연구들에 대해 소개하고, 3장에서는 SDN 기반 망에서 MIH 정보를 전달하는 무선 연동 플랫폼 구축에 대해서 설명한다. 그리고 4장에서는 무선 자원 응용 프로그램 구현과 시험에 대해 기술하며, 마지막으로 5장에서는 결론과 향후 계획에 대해 설명한다.

2. 관련연구

SDN은 네트워크 제어 기능이 물리적 네트워크와 분리되어 있는 네트워크 구조를 말한다. SDN은 기본적으로 네트워크 제어 기능이 기존의 스위치나 라우터 등의 하드웨어와 별도로 분리되며, 데이터 전달 기능과도 분리되어 개발 및 실행될 수 있는 네트워크 구조를 가진다. 분리된 SDN의 제어 기능은 네트워크 스위치(하드웨어) 상의 데이터 경로와 상호 작용할 수 있는 기능이 필수적인데, 이러한 상호 작용 혹은 통신 구조 중의 하나가 바로 오픈플로우(Openflow) 기술이다.

오픈플로우 기술은 네트워크 스위치(혹은 라우터)의 패킷 포워딩 기능과 제어 기능을 분리하고 이들 두 기능 간의 통신을 위한 프로토콜을 제공한다. 이는 외부 제어장치(서버)에 의하여 구동되는 소프트웨어가 장비 벤더에 무관하게 스위치 내의 패킷 경로 결정을 가능하게 한다. 이러한 패킷 포워딩과 제어 영역의 분리는 기존의 네트워크 장치에서 사용하는 ACL이나 라우팅 프로토콜보다 더욱 정밀한 트래픽 관리를 가능하게 한다. 따라서 오픈플로우 기술은 기존의 폐쇄적인 네트워킹 기술을 개방형 네트워킹 기술로 변화시켜서 새로운 네트워킹 기술을 발전시킬 수 있는 미래인터넷 기술로 등장하였다[6][7].

이러한 SDN 기술에 비해, IEEE 802.21 표준[8]인 MIH의 목적은 다양한 이기종 네트워크 환경에서 단절 없는 이동 서비스를 제공함으로써 모바일 단말을 사용하는 사용자 환경을 개선하는 데 있다. 모바일 단말의 네트워크 연결 지속성을 지원하기 위해서는 각 미디어마다 서로 다른 기술의 지원이 필요한데, IEEE 802.21 표준에서는 상위 레벨에서 미디어에 독립적으로 네트워크 연결 지속성에 관한 접근/제어를 할 수 있도록 하위 계층의 표준화하는 프레임워크를 제공하고 있다.

IEEE 802.21 표준에서는 미디어 독립적인 개체(Media-independent entity)를 정의하여 서로 다른

링크 계층 기술과 상위 계층 간에 공통의 인터페이스를 제공한다. 또한, 각각의 미디어 의존적인 서비스 액세스 포인트(Media-dependent service access points)를 미디어 독립적인 개체와 매핑함으로써 각 특정 기술을 제어를 할 수 있도록 하였다. IEEE 802.21에서 제공하는 기능은 이벤트(Event), 명령(Command)와 정보(Information)의 서비스의 형태로 제공됨으로써 하위 링크 계층과 상위 계층 간의 통신 모델을 제공한다. 이와 같이 MIH 기술의 목적은 이종 무선 네트워크상에서 단말에게 단절 없는 핸드오버 서비스를 제공하는 것이지만, 이러한 핸드오버를 위하여 MIH는 이동 단말과 AP의 무선 자원정보를 교환하게 된다.

Salsano외 5명[9]은 SDN을 적용한 ICN(Information Centric Networking) 프레임워크를 제안하고 이를 실증하기 위한 OFELIA 테스트베드를 구축하였고 Kim외 2명[10]은 소규모 SDN 네트워크 상에서 저렴한 비용으로 적용할 수 있는 오픈플로우 테스트베드를 구축하였다. SDN에 이동성을 지원하기 위한 연구로서 Guimarães외 4명[11]은 단일 SDN 환경에서 MIH 서비스를 제공하는 단일 프레임워크를 제안하였고 Raza외 3명[12]은 무선 이기종 네트워크 환경에서 IP 이동성을 위한 SDN 기반 아키텍처를 제안하였으며 Daniel외 3명[13]은 오픈소스를 이용한 MIH 기능을 구현하였다.

한편, IEEE 802.21 WG에서는 기존의 이종 망간 핸드오버 개념 이외에 다양한 이종 망간 연동 이슈를 발굴하고 표준화하기 위해 기존의 IEEE 802.21 MIH 표준을 프레임워크 표준(IEEE 802.21 Revision)[14]과 활용 사례에 따른 서비스 표준(IEEE 802.21.1)[15]으로 분리하고 새로운 활용 사례들을 발굴하여 표준화하고 있다.

3. SDN 기반의 무선 연동 테스트 플랫폼 구축

본 논문에서 구축한 무선 연동 플랫폼은 다음 <그림 1>과 같이 구성된다. <그림 1>의 H/W 시스템들은 크게 SDN Controller, SDN AP(Switch), MN(Mobile Node)으로 구분된다.

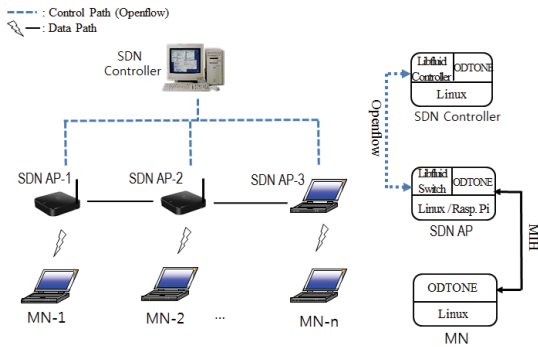


그림 1. SDN 기반 무선 연동 플랫폼
Figure 1. SDN based wireless inter-working platform

<표 1>에는 <그림 1>의 연동플랫폼 구성 시스템들에 대한 설명을 요약하였다.

표 1. 연동 플랫폼 구성 시스템
Table 1. Component systems of Inter-working platform

System	H/W	O/S	S/W
SDN Controller	노트북 (Intel i5)	Linux (Ubuntu 12.04)	Libfluid Controller S/W, ODTONE
SDN AP-1/2	Raspberry Pi B	Ubuntu-Mate 15.04	Libfluid Switch S/W, ODTONE
SDN AP-3	노트북 (Intel i5)	Linux (Ubuntu 12.04)	Libfluid Switch S/W, ODTONE
MN	노트북 (Intel i5)	Linux (Ubuntu 12.04)	ODTONE

<표 1>에서 SDN Controller와 SDN AP에는 각각 Libfluid Controller와 Libfluid Switch S/W 모듈이 탑재된다. Libfluid Controller와 Libfluid Switch는 오픈소스인 Libfluid 패키지[16]를 이용하여 본 논문에서 구현한 SDN Controller와 Switch 소프트웨어로

써 MIH 메시지를 SDN 오픈플로우 메시지로 연동하는 핵심 기능을 수행하게 된다. 또한 모든 시스템들에 공통적으로 포함되는 ODTONE[17]은 MIH 프로토콜을 구현한 오픈 소프트웨어인데, 다음의 3.1절에서 Libfluid와 ODTONE에 대해 좀 더 상세히 소개한다.

<그림 1>의 SDN Controller와 SDN AP들 간에는 오픈플로우 프로토콜에 의한 네트워크 제어 정보가 전달되는 Control Path(<그림 1>에서 점선으로 표현)가 설정되는데, 이 Control Path에는 SDN 규격에 의해 일반적인 사용자 데이터는 전달되지 않고, 오픈플로우 프로토콜에 의한 제어 정보만 전달된다. SDN AP들과 MN들 사이에는 실제 데이터가 전달되는 Data Path가 생성된다. 이 Data Path에는 MIH 메시지에 의해 AP들과 MN에 대한 무선 자원 정보들이 교환된다.

이에 따라, 본 논문의 목적인 SDN Controller에 의한 유/무선 자원 정보의 통합 관리를 위해서는 Data Path에서 전송되는 MIH 정보를 오픈플로우 프로토콜 메시지에 캡슐화하여 SDN Controller로 전달하고, 반대로 SDN Controller에서 전달되는 MIH 메시지를 AP나 MN으로 전달하는 연동 기법이 필요하다. 이 연동기법 구현에 대한 내용은 3.2절에서 설명한다.

3.1 Libfluid와 ODTONE 오픈 기술

Libfluid는 SDN Controller와 Switch 구현을 위해 필요한 기능을 제공하는 C++ 라이브러리와 예제 구현 소스를 제공하는 오픈 S/W 패키지이다. Libfluid 라이브러리는 libfluid_base 라이브러리와 lib_msg 라이브러리로 구성된다. 이 중 libfluid_base 라이브러리는 SDN Controller 구현을 위해 필요한 Switch의 연결과 연결에서 발생하는 각종 이벤트 처리를 위한 각종 함수들을 제공하고, lib_msg 라

이브리리는 SDN의 오픈플로우 프로토콜 메시지를 쉽게 생성하고 파싱(parsing)할 수 있는 기능을 제공한다. 또한 Libfluid 패키지는 이들 라이브러리 이외, SDN 핵심 기능 위주로 구현한 Controller, Switch 예제 프로그램도 함께 제공한다.

현재 SDN Controller 기능 구현 오픈 S/W로는 ONOS[18], OpenDayLight[19] 등이 있으며, SDN Switch 기능 구현 오픈 S/W로는 OpenVSwitch[20]가 있지만, 이들 오픈 S/W들은 다양한 SDN 부가 기능들을 제공하기 위해 너무 복잡하고 구현 소스도 방대하다는 단점을 가진다. 따라서, 본 논문의 연동 플랫폼의 Controller와 Switch는 Libfluid 패키지에 포함된 SDN Controller와 Switch 예제 프로그램을 기반으로 본 논문의 MIH 연동 기능을 확장하여 구현하였다.

ODTONE(Open Dot Twenty ONE)은 2008년 표준화 된 IEEE 802.21 미디어 독립적인 핸드오버 프레임워크의 C++ 언어 기반의 오픈 소스 구현이다.

ODTONE은 IEEE 802.21 표준 규격에 따라 크게 MIH-user 모듈, MIHF(MIH Functions) 모듈, LINK-SAP 모듈로 나누어 볼 수 있다. MIH user 모듈은 MIHF 모듈에서 제공하는 MIH 기능을 사용하는 응용으로써, 실제로는 Mobile IP 등의 단말의 이동성을 지원하는 네트워크 계층 프로토콜이나 본 논문에서 구현한 자원 관리 응용 등이 될 수 있다. 반면, LINK-SAP 모듈은 물리적인 링크와 네트워크 인터페이스에 관련된 작업을 담당하는 모듈으로써, 물리적인 링크와 네트워크 인터페이스에서 발생하는 이벤트들(예를 들면 링크 탐지, 링크 단절 등)을 MIHF 모듈에게 전달하거나 MIHF 모듈이 요구하는 명령들을 처리하는 역할을 수행한다. LINK-SAP 모듈은 시스템에 존재하는 물리적인 인터페이스 종류 개수만큼의 모듈이 존재해야 한다.

ODTONE 구현에서 MIH 모듈 간의 MIH 프리미티브 전달은 로컬 시스템 내부의 모듈 간이나 원

격 시스템의 모듈 간이나 모두 UDP socket을 사용하는 특징을 가진다. ODTONE은 MIH 프로토콜의 대부분의 기능들을 구현하였다는 것과 리눅스, 윈도우즈, 안드로이드 등의 다중 플랫폼에 설치 가능하다는 장점이 있지만, 현재 ODTONE에서 공식적으로 제공하는 LINK-SAP은 802.3 이더넷 인터페이스와 802.11 무선랜(WLAN) 인터페이스만 지원하는 단점도 존재한다.

3.2 SDN 기반의 MIH 연동 기능

MIH 프로토콜에서 자원 관리나 핸드오버를 위한 프리미티브 전달은 이동 단말(MN)과 AP들 간에서 수행된다. 하지만 본 논문에서는 SDN 기반의 MIH 기능을 연동하기 때문에, SDN 설계 원칙에 따라 모든 MIH 메시지 전달은 MN과 SDN Controller 사이, 또는 AP와 SDN Controller 사이에서 발생하는 구조로 플랫폼을 설계하였다. 즉 모든 무선 자원 정보들은 Controller에 의해 수집되어 통합 관리된다.

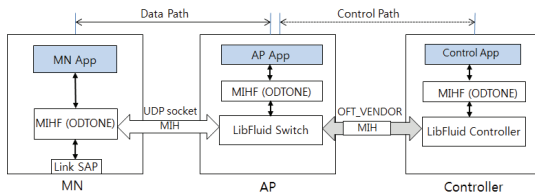


그림 2 플랫폼 상의 시스템 내부 모듈
Figure 2. Inner modules of system on the platform

앞서 언급한 바와 같이, 본 논문의 플랫폼은 MN, AP, Controller로의 세 가지 시스템으로 구성된다. 각 시스템에서 동작하는 내부 S/W 모듈과 이들 간의 프로토콜 메시지 전달은 다음의 <그림 2>와 같이 나타낼 수 있다.

<그림 2>의 MN 시스템 내부에는 “MN APP” 응용 프로그램과 ODTONE에서 구현한 MIH 메인

모듈(MIHF: MIH Functions)인 “MIHF”, ODTONE에서 구현한 “Link_SAP”의 세 가지 모듈이 존재한다. “MN APP”는 MIH user 모듈 형태로 구현되어 MIH 프로토콜 프리미티브를 이용하여 자신의 무선 자원정보 등을 최종적으로 <그림 2>의 Controller 시스템의 “Control App”으로 전달한다. 이때 AP 시스템의 “Libfluid Switch” 모듈은 MIH와 SDN 연동을 위한 MIH 메시지/오픈플로우 메시지 변환 기능을 가진 게이트웨이 역할을 수행한다.

MIH 메시지를 오픈플로우 메시지로 변환하여 전달하는 역할을 수행하기 위해, AP의 “Libfluid Switch” 모듈은 먼저 Controller로 전송되는 MIH 패킷을 탐지하는 기능을 수행해야 한다. 다음의 <그림 3>은 MIH 패킷을 탐지하는 시스템 구성과 이를 위한 AP의 플로우 테이블 구성을 보여준다.

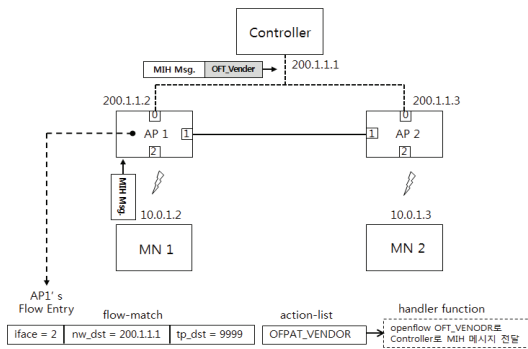


그림 3. AP 상에서 MIH 메시지 검색을 위한 플로우 테이블 구성 예

Figure 3. Flow table example in AP for finding MIH message

<그림 3>과 같은 시스템 구성에서 MN으로부터 Controller로의 MIH 메시지 전송은 AP의 “Libfluid Switch” 모듈에서 설정된 플로우 테이블에 의해 검색된다. 이러한 MIH 메시지 검색을 위한 플로우 엔트리는 AP가 Controller에 접속되는 순간에 Controller의 오픈플로우 메시지(플로우 생성)로 접

속된 모두 AP들의 플로우 테이블에 생성하게 된다.

<그림 3>의 AP 1의 플로우 엔트리 예에서는 링크 인터페이스 2번으로 목적지 네트워크(IP) 주소(flow-match의 nw_dst 항목)가 Controller의 IP 주소인 “200.1.1.1”이고 목적지 포트(flow-match의 tp_dst 항목) 주소가 “9999”를 가진 프레임은 MIH 메시지로 판단된다. ODTONE 구현에서 MIH 메시지 전송을 위한 포트 값은 임의로 할당할 수 있도록 하였기 때문에, 본 논문에서는 MIH 메시지 전송의 목적지 포트 값을 “9999”로 정하였다.

<그림 3>의 AP 1의 플로우 엔트리에 의해 검색된 MIH 메시지는 action-list에 의해 지정된 핸들러 함수에서 UDP 헤더를 제거한 순수한 MIH 메시지만 오픈플로우 프로토콜 메시지로 캡슐화(encapsulation)되어 Controller에 전달된다. 현재 SDN 규격에는 MIH 메시지 전달을 위한 오픈플로우 패킷 형식(type)이 존재하지 않기 때문에, 본 논문에서는 오픈플로우 규격의 벤더 메시지들을 전송할 수 있도록 정의한 “OFPT_VENDOR” 형식을 사용한다.

Controller의 내부 “Libfluid Controller” 모듈은 “OFPT_VENDOR” 형식으로 된 오픈플로우 메시지에서 페이로드의 MIH 메시지를 다시 UDP 로컬 전송(루프백 IP 주소)으로 Controller 내부의 “MIHF” 모듈로 전달하고, 최종적으로 “Control App” 응용 프로그램으로 전달된다(<그림 2> 참조). 이렇게 Controller 내부에서 다시 UDP 로컬 전송을 하는 이유는, 앞서 밝힌 바와 같이, ODTONE 구현의 로컬 모듈 간의 MIH 프리미티브도 UDP 소켓 전송으로 구현되어 있기 때문이다.

AP에서 Controller로의 MIH 메시지는 AP 내부의 “AP App” 응용에서 “MIHF” 모듈을 통해 UDP 로컬 전송(루프백 IP주소와 “9999” 값의 목적지 포트 값)으로 수행된다. 본 논문의 “Libfluid Switch” 모듈은 이 UDP 로컬 전송을 수신하여 이

전의 MN의 MIH 메시지를 Controller로 전달하는 동작과 동일하게 오픈플로우 메시지 내에 캡슐화하여 전달한다. <그림 4>는 AP로부터 Controller로의 MIH 메시지 전달 과정을 도식적으로 보여준다.

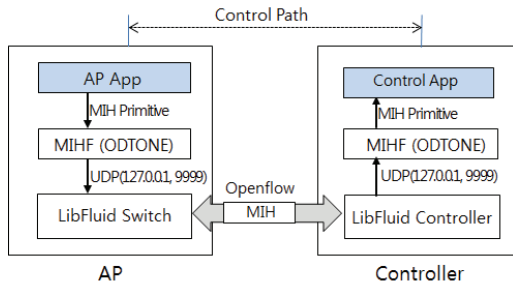


그림 4. AP에서 Controller로의 MIH 메시지 전달 과정
Figure 4. MIH message delivery from AP to Controller

4. 무선 자원관리 기능 구현 및 시험

본 장에서는 앞서 설명한 무선 연동 플랫폼의 테스트를 위한 무선 자원관리 응용의 구현과 시험에 대해 기술한다.

4.1 무선 자원관리 응용 시나리오

무선 자원관리의 응용은 플랫폼의 각 구성 시스템에 따라 MN 응용, AP 응용, Control 응용 이렇게 총 세 가지로 구성된다. SDN의 설계 구조에 맞추어 MN 응용과 AP 응용은 각 응용에서 이벤트 발생 시와 주기적인 시점에 자신들의 무선 자원에 대한 정보를 모두 Controller 시스템의 Control 응용에게 전달한다.

<그림 5>는 MN의 자원관리 응용의 동작 시나리오 예를 보여준다. <그림 5>는 MN 응용(MN App)이 MIH Link_Connect 요청에 의해 특정 AP에 접속한 경우와 MN 응용이 MIH Link_Get_Parameters 요청이 발생한 경우, MN의 무선 자원정보가

Controller의 Control 응용(Control App)에게 전달되는 과정을 보여준다. <그림 5>에서는 플랫폼에서 발생하는 프로토콜 동작을 간략하게 보여주기 위해 실제 MN의 링크 동작을 담당하는 MIH Link_SAP 모듈의 동작은 생략하였다.

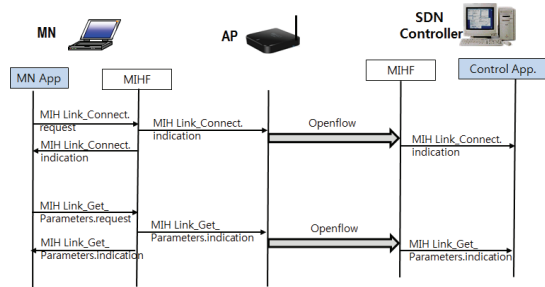


그림 5. MN 무선 자원관리 응용의 시나리오 예제
Figure 5. Example scenario of MN wireless resource management application

<그림 5>에서 MN 응용이 인접한 AP와의 접속을 요구(MIH Link_Connect.request)하면, MIH의 메인 모듈(MIHF)은 이 요청 프리미티브를 MN의 Link_SAP 모듈(그림에서는 생략)에 다시 전달하여 AP와의 링크 접속을 수행한다. 접속이 실제 수행 되면 MIHF 모듈은 접속 실행 결과(MIH Link_Connect.indication)를 MN App와 원격으로 연결된 Controller의 MIHF 모듈에 전달한다. 이 Controller로의 MIH 프리미티브는 앞 3 장에서 설명한 것과 같이 오픈플로우 메시지에 캡슐화되어 전송되며(그림 5의 음영으로 나타낸 화살표), 최종적으로 Control 응용에게 전달된다. 접속이 완료되면 MN App는 자신이 연결된 무선 링크 품질을 점검할 수 있는데, 이 때 사용하는 MIH의 프리미티브가 Link_Get_Parameters.request이다. <그림 5>에서는 이 프리미티브에 의해 획득한 무선 자원정보(링크 품질)가 Controller에 전달되는 과정도 보여준다. MIH에서 활용하는 링크품질 정보는 할당된 무선대역폭과 수신 신호 세기로 구성된다. MIH에서는 자

신의 링크품질 정보를 응용의 요청에 의해 전달받을 수도 있으며, 링크 수신 신호 세기의 임계값을 지정하여 링크의 수신 신호 세기가 임계값 이하로 떨어지는 경우에 이벤트 방식으로 전달받을 수도 있다.

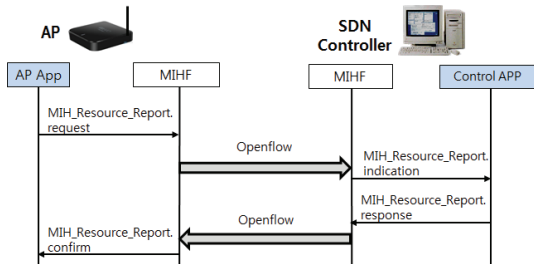


그림 6. AP의 자원관리 응용의 시나리오 예제
Figure 6. Example scenario of AP resource management application.

<그림 6>은 AP의 자원관리 응용의 동작 시나리오 일부를 보여준다. 본 논문의 AP 응용은 주기(2초 간격)적으로 현재 AP에서 사용하는 자원정보를 MIH Resource Report 프리미티브로 Controller의 Control 응용(Control App)에게 전달한다. 이때 전달되는 무선 자원정보는 주파수 할당 대역, 전송 파워, 접속된 MN 수, 각 접속의 MN 전송 속도 등으로 구성된다.

이와 같이, 본 논문의 무선 자원관리 응용은 자신의 관리하는 모든 AP의 자원 상황과 AP에 접속된 각 스테이션(MN)들의 자원정보를 모두 수집하여 데이터베이스화 할 수 있다. 이렇게 수집된 정보들은 향후 AP들의 로드 밸런싱(Load Balancing)이나 이동 단말에게 효율적인 핸드오버 지원을 위한 AP 선택을 위한 정보로 활용될 수 있다.

4.2 무선 자원관리 응용 구현 및 시험

<그림 7>은 무선 자원관리 응용을 개발하고 시

험한 플랫폼의 네트워크 구성을 도식적으로 나타낸 것이다. 전체 플랫폼은 하나의 Controller와 3개의 AP 그리고 여러 개의 MN들로 구성되며, 모든 AP들과 MN들의 무선 자원정보는 Controller로 수집된다.

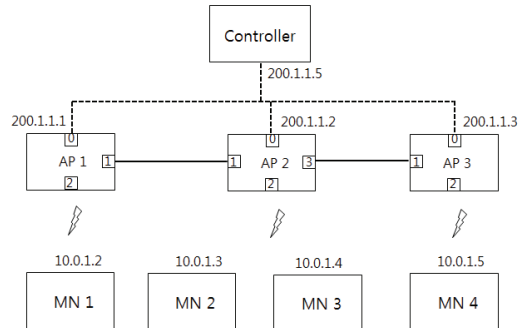


그림 7. 무선 자원관리 응용 네트워크 구성도
Figure 7. Network configuration of wireless resource management application

<그림 8>은 <그림 7>의 네트워크 구성도에 따른 무선 자원관리 응용의 시연 모습을 보여준다.



그림 8. 무선 자원관리 응용 시연 환경
Figure 8. Test environment of wireless resource management application

<그림 9>는 MN에서 동작하는 자원관리 응용의 동작 모습을 보여준다. MN 응용은 간단한 형태의 명령어 입력 방식 인터페이스로 동작한다. MN 응용을 통해 사용자는 기본적인 무선 인터페이스 정

보와 링크 정보 출력부터 접속 가능한 AP들의 스캐닝, AP에 대한 접속/해제 명령을 수행할 수 있다. 또한 이러한 수행 결과에 의해 생성된 자원정보는 <그림 5>의 시나리오에 의해 모두 Control 응용으로 전달된다.

```

=====
MIH MN App program
Usage: info ifname || link ifname || scan ifname ||
       connect ifname ssid || disconnect ifname || quit
=====
cmd> connect wlan0 "SDN-AP2"
=>> Connected to "SDN-AP2"

=====
MIH MN App program
Usage: info ifname || link ifname || scan ifname ||
       connect ifname ssid || disconnect ifname || quit
=====
cmd> info wlan0
[wlan0]
Supported Mode:IEEE 802.11abgn MAC Addr:[a0:88:b4:8f:da:dc]
Connected:yes SSID:"SDN-AP2" AP Addr:[2c:33:7a:69:b0:9f]
TX Power:15 dBm Bit Rate: 1 Mb/s

=====
MIH MN App program
Usage: info ifname || link ifname || scan ifname ||
       connect ifname ssid || disconnect ifname || quit
=====
cmd> link wlan0
Freq: 2.437[Channel:6] Sig Level: -26 dBm

=====
MIH MN App program
Usage: info ifname || link ifname || scan ifname ||
       connect ifname ssid || disconnect ifname || quit
=====
cmd>
    
```

그림 9. MN 응용 동작 예
Figure 9. Example operation of MN application

<그림 10>은 Controller 시스템에서 동작하는 Control 응용의 동작 모습을 보여준다. <그림 10>에서 볼 수 있는 바와 같이, 현재 Control 응용에 의해 모니터링 되는 모든 AP들의 목록 정보도 볼 수 있으며 각 AP에 접속된 단말들의 무선 접속 정보도 볼 수 있다. <그림 10>에서 보인 것 외에, 각 AP들의 상세한 무선 링크 정보도 모니터링 할 수 있다.

이와 같이, 본 논문에서는 플랫폼 상에서 작성한 무선 자원관리 응용 동작의 상세한 기능 시험을 수행하였으며, 이러한 시험에 따라 논문에서 구축한 SDN 환경의 MIH 연동 플랫폼이 정확히 동작하고 있음을 확인할 수 있었다.

```

=====
MIH Control App program
Usage: show ap list || show ap [#ap] || show sta [#ap] ||
       del sta [#ap] [#sta] || quit
=====
cmd> show ap list

AP[0] IPAddr: 200.1.1.2 iface: wlan0 mac: 2c:33:7a:69:af:ff
      SSID: "SDN-AP3 sta_count: 1
AP[1] IPAddr: 200.1.1.3 iface: wlan1 mac: 10:f6:3f:29:13:0d
      SSID: "SDN-AP3 sta_count: 0
AP[2] IPAddr: 200.1.1.1 iface: wlan1 mac: 10:f6:3f:29:15:35
      SSID: "SDN-AP3 sta_count: 2

=====
MIH Control App program
Usage: show ap list || show ap [#ap] || show sta [#ap] ||
       del sta [#ap] [#sta] || quit
=====
cmd> show sta 0

sta[0] Addr: d0:57:85:97:02:d5
      Inactive time: 160 ns
      Connectd time: 8641 seconds
      Rx Packets: 4382 pkts
      Tx Packets: 477 pkts
      Signal Strength: -55 dBm
      Tx_BitRate: 1.0 Mbps
      Rx_BitRate: 1.0 Mbps

=====
MIH Control App program
Usage: show ap list || show ap [#ap] || show sta [#ap] ||
       del sta [#ap] [#sta] || quit
=====
cmd>
    
```

그림 10. Control 응용 동작 예
Figure 10. Example operation of Control application

5. 결론

스마트 폰을 비롯한 다양한 형태의 스마트 디바이스들의 사용이 급증하면서 무선 데이터 사용량이 폭발적으로 증가하고 있다. 하지만 많은 양의 데이터를 전송하기 위한 3G/4G 셀룰러 망의 확충 및 무선랜의 안정성과 협소한 서비스 영역 문제가 대두되면서 서로 다른 특성을 갖는 무선 통신망 간에 끊임없는 통신을 지원하는 개방형 미디어 독립적인 연동기술이 요구되고 있다.

이를 위해 본 논문에서는 SDN 환경의 무선 연동 테스트베드 플랫폼을 구축하고 무선 자원 관리를 위해 제시된 예제 시나리오에 따라 SDN 프로토콜과 IEEE 802.21 MIH 표준의 연동 방법을 설계하고 구현하였다. 또한 SDN과 MIH 연동에 따른 이동성 및 네트워크 자원 관리 응용 프로그램을 작성하여 연동 플랫폼 상에서 동작 시험을 수행하였다. 이러한 동작 시험 결과에 따라, 본 과제에서 구현

한 SDN 환경의 무선 연동 플랫폼과 플랫폼 상의 무선 자원 관리 응용이 정상적으로 동작함을 확인할 수 있었다.

본 논문의 연동 테스트 플랫폼과 이동성 및 자원 제어 관리 기술은 앞으로 TTA, IEEE 802 WG 등의 국내/외 표준화 기구에서 표준 문서 개발에 활용할 수 있을 것으로 여겨지며, 다양한 형태의 SDN 기반의 무선망의 이동성 및 자원 제어/관리 기술의 토대가 될 수 있으리라 생각된다. 또한 현재 많은 관심을 받고 있는 SDN 기반의 이동 단말의 미디어 독립적인 연동 서비스 기술 개발의 중요한 자료로 활용될 수 있으리라 기대된다.

본 논문의 자원관리 응용은 단말과 AP들의 자원 정보를 수집하여 모니터링 할 수 있는 기능만 제공하는데, 향후 수집된 자원정보를 바탕으로 AP들의 효율적인 대역폭 할당과 로드 밸런싱 정책의 수립과 구현 방안에 대한 연구가 필요하다. 또한 MIH 프로토콜의 목적이 이동 단말의 미디어 독립적인 이동성 지원이므로, 이에 따른 SDN 기반의 이기종 무선망에서 이동성 지원 기능 설계와 구현에 대한 연구도 계속 진행되어야 할 것으로 생각된다.

References

- [1] F. Hu, Q. Hao, and K. Bao, *A survey on software-defined network and openflow: from concept to implementation*, IEEE Communications Surveys & Tutorials, Vol. 16, Issue 4, pp. 2181-2206, 2014.
- [2] K-H. Lee, *Mobility management framework in software defined networks*, International Journal of Software Engineering and Applications Vol. 8, pp 1-10, Aug. 2014.
- [3] K. Pentikousis, Y. Wang, and W. Hu, *Mobileflow: Toward software-defined mobile networks*, Communication Magazine, Vol. 51, No. 7, pp. 44-53, 2013.
- [4] M. Arslan, K. Sundaresan, and S. Rangarajan, *Software-defined networking in cellular radio access networks: Potential and Challenges*, IEEE Communication Magazine, Vol. 53, No. 1 pp. 150-56, 2015.
- [5] B. S. Ghahfarokhi, and N. Movahhedinia, *A survey on applications of IEEE 802.21 Media Independent Handover framework in next generation wireless networks*, Computer Communications, Vol. 36, pp. 1101-1119, Jun. 2013.
- [6] P. Dely, A. Kassler, and N. Bayer, *Openflow for wireless mesh networks*, in Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on, pp. 1-6, 2011.
- [7] Open Network Foundation, *Conformance test specification for openflow switch specification*, v1.0.1, Jun. 2013.
- [8] IEEE Standard for Local and metropolitan networks-Part 21: Media Independent Handover Services, IEEE Std. 802.21TM-2008, Jan. 2009.
- [9] S. Salsano, N. Blefari-Melazzi, A. Detti, G. Morabito, and L. Veltri, *Information centric networking over SDN and openflow: Architectural aspects and experiments on the OFELIA testbed*, Computer Networks, Vol. 57, Issue 16, pp. 3207-3221, 2013.
- [10] H. Kim, J. Kim, and Y. B. Ko, *Developing a cost-effective openflow testbed for small-scale software defined networking*, In 16th International Conference on Advanced Communication Technology, pp. 758-761, Feb. 2014.

[11] C. Guimarães, D. Corujo, R. L. Aguiar, F. Silva, and P. Frosi, *Empowering software defined wireless networks through media independent handover management*, Proceeding of IEEE Global Communications Conference (GLOBECOM), pp. 2204-2209, 2013.

[12] S. M. Raza, S. Yeoum, D. Kim, and H. Choo, *SDN based architecture for IP mobility in wireless heterogeneous networks*, in Proceedings of the Spring Conference on KSII, pp. 5-6, 2014.

[13] D. Corujo, C. Guimaraes, B. Santos, and R. L. Aguiar, *Using an open-source IEEE 802.21 implementation for network-based localized mobility management*, IEEE Communication Magazine, Sep. 2011

[14] IEEE 802 Draft Standard, *Local and metropolitan area networks-Part 21: Media independent services framework*, IEEE P802.21/D03, May 2016.

[15] IEEE 802 Draft Standard, *Local and metropolitan area networks-Part 21.1: Media independent services*, IEEE P802.21.1/D03, May 2016.

[16] Libfluid: The ONF OpenFlow driver, <http://opennetworkingfoundation.github.io/libfluid/>, Aug. 2016.

[17] ODTONE: An Open Source Multi-Platform Implementation of the IEEE 802.21 Protocol, <http://atnog.github.io/ODTONE/>, Oct. 2016.

[18] ONOS: Open Network Operating System, <http://onosproject.org/>, Oct. 2016.

[19] OpenDaylight: Open Source SDN Platform, <https://www.opendaylight.org/>, Sept. 2016.

[20] OpenvSwitch: Production Quality, Multilayer Open Virtual Switch, <http://openvswitch.org/>, Oct. 2016.

SDN 환경에서 MIH를 이용한 무선 연동 플랫폼 개발

이완직, 정윤필, 허석렬

부산대학교 IT응용공학과

요 약

유선 코어 네트워크상에서는 SDN 기반의 네트워크 제어 기술이 크게 주목받고 있으며, 향후 네트워크 상의 모든 제어와 자원 관리가 SDN 기술로 통합될 것으로 기대되고 있다. 반면, IEEE 802.21 표준인 MIH(Media Independent Handover)는 미디어 독립적인 핸드오버 서비스를 지원하기 위해 무선 자원정보를 관리하는 기술을 포함하고 있다. 이러한 두 가지 기술을 결합하면, 기존의 유선 네트워크의 자원 및 제어 정보들만 관리하는 SDN Controller가 무선 네트워크의 자원 및 이동성 관련 내용을 관리하는 MIH의 정보를 같이 수집하여 유/무선 네트워크 자원 정보나 이동성을 통합적으로 관리할 수 있는 장점을 가진다. 이를 위해, 본 논문에서는 SDN 환경에서 MIH 표준프로토콜의 연동 방안을 제안하여 SDN 기반의 MIH 연동 플랫폼을 구축하였다. 또한 플랫폼 상에서 간단한 SDN 기반의 무선 자원 관리 응용 프로그램을 작성하여, 본 논문에서 구축한 플랫폼의 수행 기능을 시험하였다. 본 논문에서 구축한 SDN 환경의 무선 연동 플랫폼은 SDN 기반 AP들의 무선 대역폭 관리나 AP들의 부하 관리 기능을 테스트하기 위해 사용될 수 있으며, 할당된 자원정보에 의한 단말 이동성 기술 개발용으로도 활용될 수 있다.

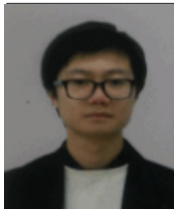
감사의 글

이 논문은 부산대학교 기본연구지원사업(2년)에 의하여 연구되었음.



Wan-jik Lee received the bachelor's degree in the Department of Statistics from the Kyungbook National university in 1992. He received the M.S. degree and the Ph.D. degree in the Department of Computer Engineering from Kyungbook National University in 1994 and 2007, respectively. From 1997 to 2006, he was a professor in the Department of Information & Communication Engineering at Miryang National University. He has been a professor in the Department of Applied IT & Engineering at Pusan National University since 2006. His current research interests include network communication protocol, SDN and IoT communication technology. He is a regular member of the KKITS.

E-mail address: wjlee@pusan.ac.kr



Jung Yun Pil received bachelor's degree in the Department of Cyber Police from the Youngsan university in 2015. He is currently a graduate student in the Department of Applied IT & Engineering at Pusan National University. His interests are SDN and IoT communication technology.

E-mail address: whitecrist@naver.com



Seok-Yeol Heo received bachelor's degree in the Department of Electronics from the Kyungbook National university in 1986. He received the M.S. degree and the Ph.D. degree in the Department of Computer Engineering from Kyungbook National

University in 1991 and 2008, respectively. From 1992 to 2006, he was a professor in the Department of Computer Engineering at Miryang National University. He has been a professor in the Department of Applied IT & Engineering at Pusan National University since 2006. From 2012 Aug. to 2013 Jul., he was a visiting scholar at University of Dallas. His current research interests include IoT and USN technology. He is a regular member of the KKITS.

E-mail address: syheo@pusan.ac.kr