

# Implementation of an OpenFlow-based Access Point Virtual Switch for Monitoring and Virtualization of Legacy Wireless LAN

Hyung-Bong Lee \*, Ji-Hoon Park \*\*

## Abstract

Network virtualization is an emerging technology for solving the rigidity of the physical network infrastructure. The innovative technique virtualizes all resources in the network, including the network links and nodes, and provides a number of virtual networks on a single network infrastructure. In order to realize a virtual network, a thorough and complete monitoring of all resources in the network should be performed firstly. OpenFlow is an open source stack for network virtualization. However, it is impossible to apply OpenFlow to AP-based legacy wireless LAN environment because OpenFlow targets ethernet-based LAN environment. In this paper, we implement an adaptor-styled virtual switch for AP-based wireless LAN through customizing the Open vSwitch which is a virtual switch of OpenFlow. The evaluation test results show that the implemented OpenFlow stack operates successfully. The implemented OpenFlow stack can now be plugged immediately in existing AP-based wireless LAN environment and plays network resource monitoring. In the future, we can develop wireless LAN virtualization applications on the wireless OpenFlow stack.

▶ Keyword : Network Virtualization, SDN, OpenFlow, Open vSwitch, Wireless LAN

## I. Introduction

전통적인 무선랜 환경에서는 스마트폰, 노트북 등의 무선 단말들이 그림 1과 같이 유선 백본(스위치)에 연결된 AP(Access Point)에 Wi-Fi 채널을 통하여 인터넷에 접속한다. 최근 5~6년 간 Wi-Fi를 통신 수단으로 하는 스마트 디바이스의 폭발적인 확산과 사물 인터넷(IoT: Internet of Things) 단말의 등장으로 무선 트래픽이 급속도로 증가하고 있다. 잘 알려진 바와 같이 무선랜의 전송률은 적절한 트래픽 범위를 벗어나는 순간 급격히 저하되는 특성을 가지고 있다[1]. 학교나 대규모 행사장과 같이 무선 단말들이 특정 지역에 편중하여 집중될 경우 이를 감당하는 AP들의 성능이 급격하게 저하되어 사실상 통신이 마비되어 쓸모가 없게 된다. 이와 같은 과열 상황을 예방하기 위해서는 AP들의 상태를 모니터링하여 장기적으로는 AP를 적절하게 재배치해야 하고, 실시간적으로는 특정 AP에 대한 접

속 제한 조치 등의 컨트롤 메커니즘이 필요하다. 이러한 시스템을 위한 가장 쉬운 접근 방법은 특화된 AP와 컨트롤 프로토콜을 새롭게 구현하여 중앙에서 제어하도록 하는 것이다. 실제로 네트워크 장비 시장의 많은 부분을 점유하고 있는 시스코는 LWAPP/CAPWAP 프로토콜[2, 3]에 기반한 전용 AP를 개발하여 공급하고 있다. 그러나 이 시스템을 도입하기 위해서는 기존 시스템을 모두 교체해야 할 뿐 아니라, AP 당 3~4백만 원대의 높은 비용을 지불해야 한다. 또 다른 한편으로, 기존 물리적 네트워크의 경직성 한계를 극복하기 위한 방안으로 네트워크 가상화 기술이 떠오르고 있다. 네트워크 가상화는 노드와 링크를 포함한 하나의 물리적 네트워크 인프라를 가상화하여 요구 사항이 상이한 응용 및 서비스별로 독립된 여러 개의 가상 네트워크 인프라를 지원한다. 가상 네트워크는 통상 SDN(Software Defined Network)이라 부르고, 주로 이더넷 기반 유

• First Author: Hyung-Bong Lee, Corresponding Author: Ji-Hoon Park

\*Hyung-Bong Lee (hblee@gwnu.ac.kr), Dept. of Computer Science & Engineering, Gangneung-Wonju National University

\*\*Ji-Hoon Park (if93014@gmail.com), e-LibraryCenter

• Received: 2015. 11. 03, Revised: 2015. 12. 22, Accepted: 2016. 01. 11.

선 LAN 환경에서 연구되고 있으며, 그 대표적인 오픈 소스 기술로 OpenFlow를 들 수 있다[4].

이 연구에서는 OpenFlow를 기존의 전통적 AP 기반 무선랜 인프라에 적용하여 무선랜 가상화 환경을 마련하고, 이를 이용하여 무선랜 모니터링 시스템을 구현한다. 이를 위하여 유선랜 스위치 환경을 목표로 구현되어 있는 가상 스위치인 Open



Fig. 1. The Legacy Wireless LAN Environment

vSwitch(OpenFlow 용 스위치)를 어댑터 스타일로 개발하여 AP 전단에 배치한다. 이 방안은 기존에 구축된 무선랜 인프라를 그대로 활용할 수 있을 뿐 아니라, 특정 벤더 중심의 고가 AP 대신 중저가의 일반 AP를 활용할 수 있으므로 무선랜 구축 비용을 크게 절감할 수 있다. 이 논문의 2장에서는 네트워크 가상화 관련 연구를 살펴보고, 3장에서는 일반 무선 AP를 OpenFlow와 연동시키기 위한 어댑터 스타일의 가상 스위치를 설계·구현하며, 4장에서 개발된 어댑터의 기능을 평가한다. 그리고 마지막 5장에서 결론으로 맺는다.

## II. Related Works

### 1. Virtual Networks

#### 1.1 VLAN, VPN

VLAN(Virtual LAN)은 VLAN id를 이용하여 물리적으로 구분된 LAN의 여러 세그먼트를 하나의 가상 세그먼트로 묶어주거나 하나의 세그먼트 내에서 다른 트래픽과 독립된 논리적 세그먼트를 제공한다. VPN(Virtual Private Network)는 라우터에서 특정 패킷을 식별하여 지정된 경로를 따라 전송함으로써 하나의 전용 가상 통신 선로를 제공해 주는 기능으로 터널링이라 부르기도 한다. VLAN과 VPN이 가상 네트워크 개념의 초기 모습이라 할 수 있다[5].

#### 1.2 SDN

SDN은 1995년경부터 사용된 오래된 용어이지만 그 개념 및 내용은 많은 변화를 겪어왔다. SDN이 추구하는 방향은 그림 2와 같이 데이터 평면(패킷 라우팅)과 제어 평면(라우팅 테이블 설정) 등 두 가지 기능을 동시에 가지고 있는 기존의 라우터나

스위치에서 제어 평면을 분리하여 라우팅 정책을 중앙에서 유연하게 컨트롤함으로써 하부의 물리적 네트워크 인프라를 논리적 인프라로 관리하자는 것이다[6]. 즉, 중앙 컨트롤러의 소프트웨어가 라우팅 정책을 수립하여 라우터 및 스위치에 전달하면 이들은 수신한 패킷을 지정된 정책에 따른 전달 역할만 담당한다. SDN의 장점이 네트워크 관리자 및 사용자의 의도에 따른 복잡하고 다양한 라우팅 정책의 실시간적 생성 및 변경이 가능하다는 점에서 오늘날의 클라우드 네트워크를 지원하기 위한 필수 기술로 평가받고 있다[4].

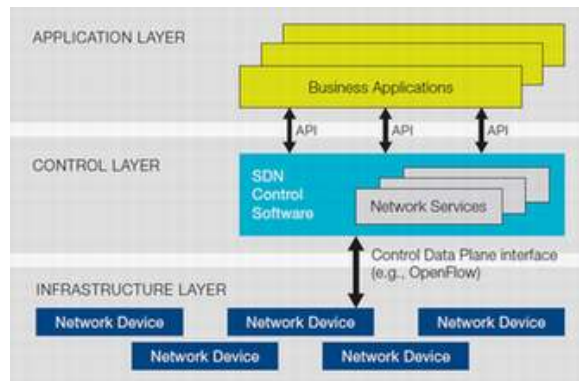


Fig. 2. Software Defined Network Architecture

### 1.3 OpenFlow

SDN 개념을 실현하기 위해서는 특정 벤더나 장비에 관계없이 공통으로 통신할 수 있는 표준화된 컨트롤 프로토콜이 필요하다. 이런 필요성에 의해 스탠포드 대학을 중심으로 SDN을 구현하는 OpenFlow 프로젝트[7]가 시작되었고, 현재는 굴지의 벤더들이 참여하는 ONF(Open Networking Foundation)가 설립되어 연구가 진행되고 있다[8]. OpenFlow는 한 마디로 south-bound(컨트롤러에서 장비 방향) 통신 규약이라 말할 수 있고, 이 규약에 따른 컨트롤러와 스위치에 대한 프로토타입 오픈 소스가 제공되고 있다. 그림 3에 보인 OpenFlow의 전체적인 개념에서 스위치(라우터)는 입력된 패킷에 대응되는 플로우 테이블의 내용에 따라 전달 역할만 담당한다. 만약 플로우 테이블이 발견되지 않으면 그림 4와 같이 컨트롤러에게 질의한 결과에 따른다[9].

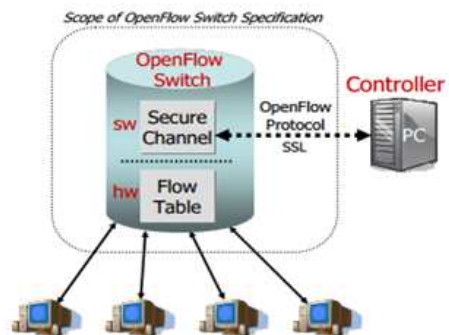


Fig. 3. Overview of OpenFlow

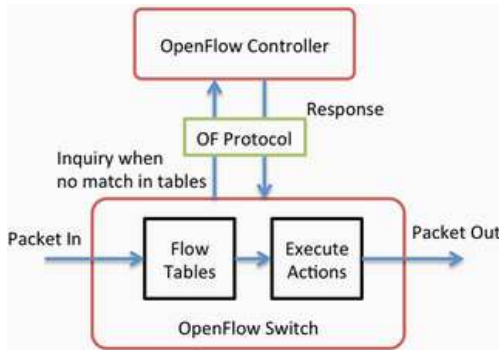


Fig. 4. Packet Flow in an OpenFlow Switch

## 2. OpenFlow in Wireless Network

이더넷 기반의 유선랜 가상화에 중점을 둔 OpenFlow를 무선 네트워크에 적용하고자 하는 시도들이 있는데, 이 분야의 연구는 주로 IEEE 802.11s, IEEE 802.15 기반의 무선 메쉬 네트워크 즉, 무선 백본 환경에 OpenFlow를 적용하는 방안을 다룬다. [10]은 무선 메쉬 네트워크에서 메쉬 클라이언트(무선 단말)에 OpenFlow 프로토콜을 구현하고, 이를 관리하는 OpenFlow 컨트롤러를 추가하여 이원화시키는 방법으로 클라이언트의 이동성을 지원하는 방안을 제시하였다. [8]은 OpenFlow를 이용하여 무선 메시 네트워크에 플로우 기반 라우팅을 적용하여 기존 표준 라우팅 프로토콜 중의 하나인 HWMP(Hybrid Wireless Mesh Protocol)보다 크게 개선된 결과를 보였고, [11]에서는 [8]의 결과를 확장하여 콘텐츠를 거점 노드에 보관하고 이를 라우팅에 활용하는 콘텐츠 라우팅 기법으로 무선 백본 내 트래픽 감소 방안을 제시하였다. 이런 연구들은 이미 설치되어 있는 AP 기반 무선랜 환경에서의 OpenFlow 적용 방안과는 거리가 멀다. 한편, SDN 및 OpenFlow와는 별개로 무선 트래픽의 증가와 함께 AP 기능에 대한 상황 파악 필요성 증대에 따라 SNMP 프로토콜 기반으로 AP를 모니터링하는 몇몇 시도들이 있었는데 [12-14], 이들은 모니터링 자체를 위한 트래픽 혼잡을 초래하고 모니터링 어플리케이션을 각각의 설계에 따라 비표준으로 구현하는 한계를 지니고 있다. 이 연구에서는 OpenFlow의 가상화 기능을 기구축된 기존 AP 기반 무선랜에 적용하여 AP 모니터링 및 무선랜 가상화 기반 마련 방안을 제시한다.

## III. OpenFlow-based Access Point(AP) Virtual Switch

### 1. Open vSwitch

Open vSwitch는 그림 4의 OpenFlow 스위치의 구현 프로토타입으로 OSI L2(Layer 2)에서 동작하는 리눅스 브리지 기반 소프트웨어 스위치의 일종으로, 하드웨어 스위치와 분리되어 가상 네트워크를 펼치는 하이퍼바이저(Hypervisor)와 하부

의 물리적 네트워크 사이에서 가상 네트워크별 트래픽 분리·모니터링·QoS 관리 등 논리적 스위칭 기능을 제공한다. 이 가상 스위치에는 데이터 평면만 유지되고 제어 평면은 중앙의 OpenFlow 컨트롤러로 이관되어 그 자리에는 컨트롤러와 통신하기 위한 OpenFlow 에이전트가 대신한다. Open vSwitch의 개략적인 구성 요소는 그림 5와 같고 그 주요 내용은 아래와 같다.

- 소프트웨어는 크게 커널 영역과 사용자 영역(어플리케이션)으로 분리

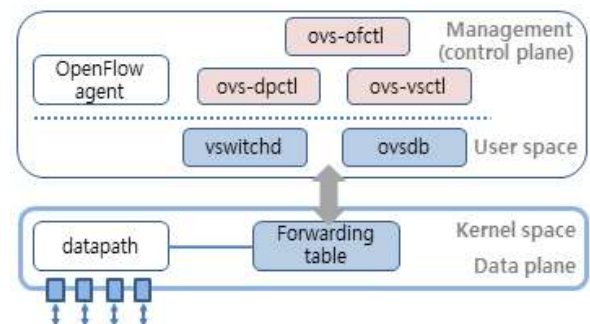


Fig. 5. Software Modules of Open vSwitch

- vswitchd: 가상 스위치 컨트롤을 위한 전반적인 외부 통신연결 디몬으로 OpenFlow 프로토콜에 따라 커널, DB 서버(ovsdb), 기타 외부 시스템과 통신
- datapath: 가상 스위치의 커널 모듈로 패킷 전달 역할 수행
- ovs-dpctl: 커널의 datapath 모듈을 설정하는 유틸리티
- ovs-vsctl: 디몬인 vswitchd의 설정 및 갱신 유틸리티
- ovs-ofctl: 가상 스위치와 컨트롤러 사이의 질의 및 제어 유틸리티

### 2. AP Virtual Switch Design Approach

일반적으로 L2(Data link layer) 스위치는 브로드캐스팅 방식인 이더넷 프로토콜에 따라 프레임 헤더의 MAC 주소에 기반하여 프레임 단위로 전송하고, L3(Network layer) 라우터는 IP 프로토콜에 따라 패킷 헤더의 IP주소에 기반하여 분리된 통신망사이에서 중계 역할을 수행한다. 일반적인 L2 스위치에서는 전송이 하드웨어적으로 이루어지고, L3 라우터에서는 중계(전달) 기능이 소프트웨어적으로 이루어진다. 반면에 L3 스위치는 하드웨어의 발전으로 전달 경로 결정을 하드웨어적(FPGA: Field Programmable Gate Array)으로 수행하고, L2 스위치를 확장하여 구현한다. 이 연구에서는 L3 스위치 모델을 도입하여, AP 가상 스위치를 그림 6의 프로토콜 스택과 같이 AP 상단에 물리적으로 분리된 L2 계층에 Open vSwitch를 구현하고, 우선은 모니터링기능에 집중하기 위해 L3에 별도의 라우팅 프로토콜을 탑재한다. 또한, 가상 스위치 하단에는 AP의 트래픽 중 데이터 패킷과 제어 패킷을 필터링하고 가상 스위치의 입력 포트 연결을 위한 스위칭 실리콘(switching silicon)을 배치하되 이 또한 물리적으로 분리된 플랫폼을 사용한다.

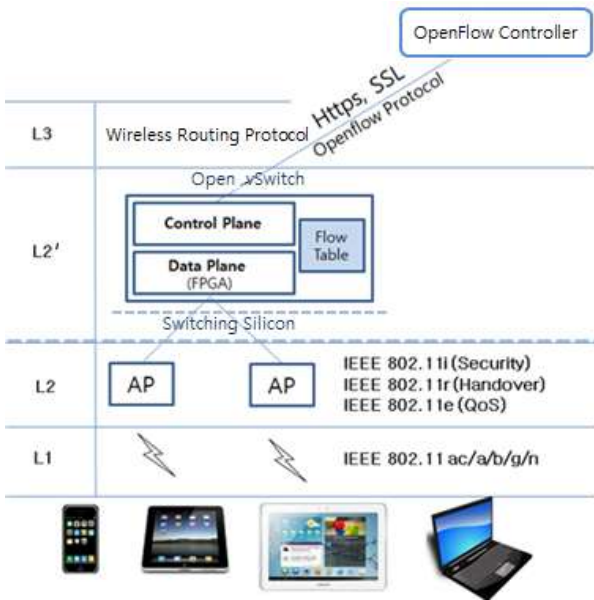


Fig. 6. Protocol Stack of the Proposed AP Virtual Switch

### 3. Implementation of the Proposed AP Virtual Switch

#### 3.1 Hardware Platform

시중에서 구입 가능한 상용 인터넷 스위치나 라우터들은 일반 사용자에게 의한 펌웨어 수정을 허용하지 않기 때문에 이 연구에서는 AP 가상 스위치를 구현할 하드웨어 플랫폼으로 임베디드 리눅스 다운로드가 가능한 MikroTik 사[15]의 Atheros SoC를 사용하는 상용 라우터 보드인 RB450G를 이용하고, 그 주요 사양은 표 1과 같다. 스위칭 실리콘 플랫폼으로는 기존의 일반 AP를 활용하였다. 그림 7은 RB450G 라우터 보드의 실제 모습이다.

Table 1. RB450G Specification(Virtual Switch)

Items	Specification
Bootloader	RouterBoot
SOC	Atheros AR7161 MIPS 24K
CPU Speed	680Mhz (Max 800Mhz)
Flash-Chip	Hynix NAND 512MiB
Flash size	512M
RAM	256M
Ethernet	Atheros AR8316
USB	None
Serial	1 Port

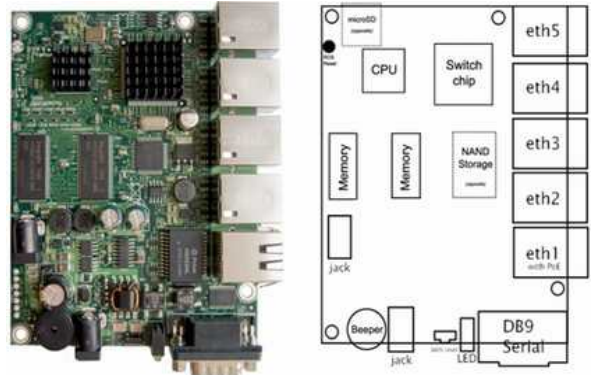


Fig. 7. Photo of MikroTik RB450G Board

#### 3.2 Software Implementation

□ OpenFlow 가상 스위치

가상 스위치를 탑재할 운영체제로는 무선랜 라우팅 정책을 플러그인할 수 있는 임베디드 리눅스 기반 OpenWrt(Open Wireless Router)를 RB450G 플랫폼에 이식·수정·구현한다. 그 주요 과정은 아래와 같다.

- 교차 컴파일 호스트 컴퓨터 설치

윈도우 7용 VirtualBox 내에 Ubuntu ver. 12.04 설치 후 필수 모듈들 삽입하여 호스트 운영체제 이미지를 재생성한다.

```
> sudo apt-get install build-essential subversion git-core libncurses5-dev zlib1g-dev gawk flex quilt libssl-dev xsltproc libxml-parser-perl unzip
```

- OpenWrt 소스 코드 다운로드·설정·빌드

OpenWrt 커뮤니티에서 패키지를 다운받아 설정파일을 업데이트하고, 필요 패키지 선택 후 빌드한다.

```
> git clone git://git.openwrt.org/openwrt.git && cd openwrt
> mv feeds.conf.default feeds.conf
> echo 'src-git openvswitch git://github.com/pichuang/openvswitch.git' >> feeds.conf
> ./scripts/feeds update -a
> ./scripts/feeds install -a
> make menuconfig
> make prereq
> make
```

- Open vSwitch(OVS) 소스 코드 다운로드·설정·빌드

OVS 커뮤니티에서 패키지를 다운받아 설정파일을 업데이트하고, 필요 패키지 선택 후 가상스위치 전체 펌웨어를 빌드한다.

```
> ./scripts/feeds update -a
> ./scripts/feeds install -a
> echo 'src-git openvswitch git://github.com/pichuang/open
vswrt.git' >> feeds.conf
> ./scripts/feeds update openvswitch
> ./scripts/feeds install -a -p openvswitch
> make menuconfig
> make kernel_menuconfig
> make
```

- 가상 스위치 펌웨어 업로드  
TFTP를 이용하여 빌드된 가상 스위치 펌웨어를 RB450G 보드에 업로드한다.
- OpenWrt 네트워크 설정  
그림 8의 내용으로 스위칭 실리콘에서의 인터넷 라우팅을 위한 네트워크 환경을 설정한다.

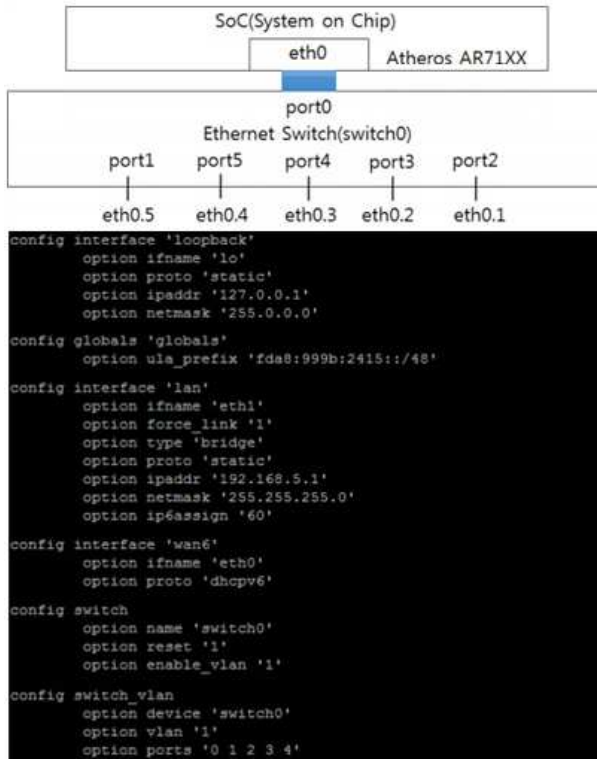


Fig. 8. Network Configuration of OpenWrt

- 가상 스위치 브리지, 브리지 인터페이스, 컨트롤러 설정  
가상 스위치와 스위칭 실리콘 이더넷과의 연결을 위한 가상 브리지를 설정하고, OpenFlow 컨트롤러와의 통신을 위한 포트를 설정한다.

```
# ovs-vsctl add-br ovsbr0
# ovs-vsctl add-port ovsbr0 eth1
# ovs-vsctl set-controller ovsbr0 tcp:1.2.3.4:6633 vswrt.git'
>> feeds.conf
```

그림 9는 AP 모니터링 및 가상화를 위해 구현된 AP 가상 스위치 전체의 블록 다이어그램이다.

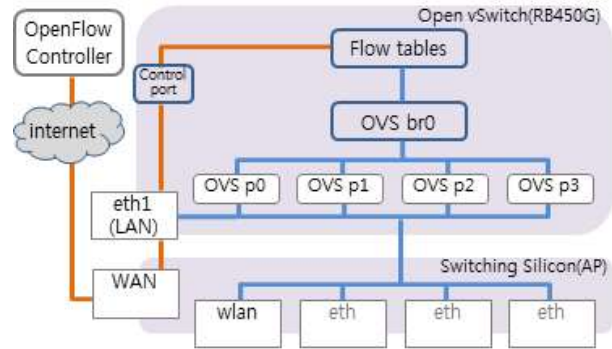


Fig. 9. Block Diagram of the Implemented Virtual Switch and a General AP

□ OpenFlow 컨트롤러

OpenFlow 컨트롤러는 운영자의 목적과 환경에 따라 수정과 확장이 가능하도록 프로토타입으로 제공되는 Java 기반 FloodLight를 설치하여 활용한다. 이를 위하여 JDK, Eclipse, Ant 패키지 등과 함께 FloodLight를 펌웨어 개발 시스템 (Ubuntu 12.04)에 설치한다.

```
> sudo apt-get install build-essential default-jdk ant
python-dev eclipse
> git clone git://github.com/floodlight/floodlight.git
> cd floodlight
> sudo mkdir /var/lib/floodlight
> sudo chmod 777 /var/lib/floodlight
> java -jar target/floodlight.jar
> ant eclipse
```

4. Evaluation of the Implemented AP Virtual Switch

4.1 Test Environment

구현된 AP 가상 스위치 기능 평가를 위해 그림 10과 같이 가상 스위치와 AP를 하나의 set으로 구성하고 OpenFlow 컨트롤러인 FloodLight와 연결되도록 무선 환경에서의 가상 네트워크 환경을 구축하여, 가상 스위치와 실시간 접속 스마트 디바이스들의 토폴로지, AP들의 트래픽, 무선랜에 대한 실시간 모니터링 기능을 검증한다.

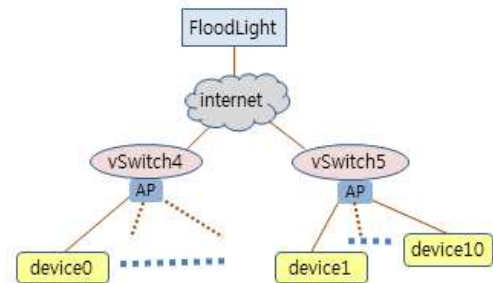


Fig. 10. Test Environment for Wireless LAN Monitoring

4.2 Network Topology

무선 네트워크 가상화 환경에서 가상스위치가 현재 접속된 단말들에 대한 데이터를 콘트롤러인 FloodLight에게 실시간으로 전송한 결과를 웹에서 그래프화한 토폴로지가 그림 11과 같이 나타나는데 그림 10의 네트워크 구성도와 일치함을 확인할 수 있다. AP와 가상 스위치가 하나로 구성되므로 두 개의 가상 스위치에 접속된 무선 단말(스마트 디바이스)를 실시간으로 나타내고 있으며 접속된 디바이스의 MAC 주소도 표현되어 있다. AP1에는 하나의 무선 단말이 연결되어 있고, AP2에는 10개의 무선 단말이 연결되어 있다.

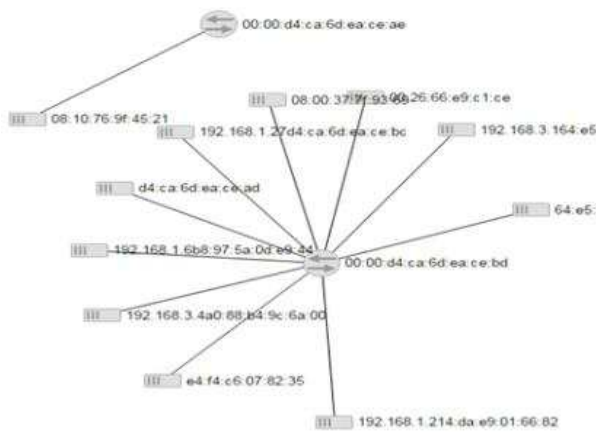


Fig. 11. Wireless Network Topology Report Generated by FloodLight for Fig. 12

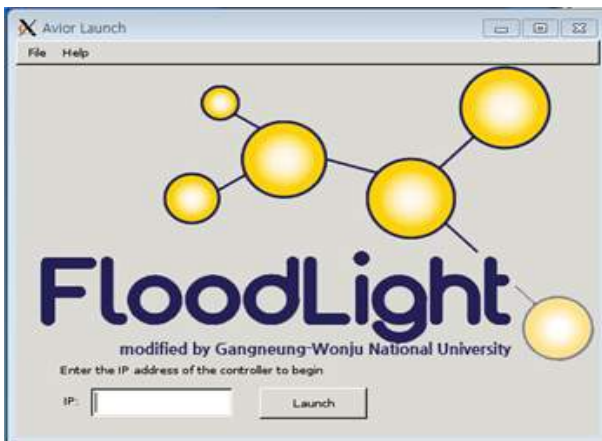


Fig. 12. Execution of Avior Program Connected with FloodLight

4.3 Traffic Monitoring of Access Points

FloodLight의 모니터링 데이터를 GUI 형태로 접근하기에는 Avior[16]가 적합하다. Avior는 자바에 기반한 SDK 일종으로, 테스트 환경에 맞추어 수정하고 실행하면 그림 12와 같이 기 설치된 OpenFlow 콘트롤러(FloodLight)에 접속하여 기동한다. Avior로 확인한 가상 스위치(AP)별 트래픽 통계 현황은 그림 13과 같이 나타난다. 이 그림으로부터 가상 스위치의 포트 별 트래픽들이 정확하게 식별되고 있음을 확인할 수 있다.

Ports					
#	Link Status	Tx Bytes	Rx Bytes	Tx Pkts	Rx Pkts
3	UP - 100 Mbps FDX	1001 MB	164 MB	921 K	847 K
2	DOWN	14 MB	7522 KB	14 K	14 K
24	UP - 100 Mbps FDX	129 MB	194 MB	437 K	672 K
5	UP - 100 Mbps FDX	121 MB	986 MB	822 K	888 K
4	DOWN	45 MB	2504 KB	33 K	30 K
65534	UP	0	0	0	0
1	DOWN	11 MB	38 MB	84 K	33 K

Flows			
#	Priority	Match	Action
1	0	src-mac:08:10:76:9f:45:20, dst-mac:00:21:6b:c8:d2:fc, port:5	output=3
2	0	src-mac:00:21:6b:c8:d2:fc, dst-mac:08:10:76:9f:45:20, port:3	output=5

Fig. 13. Traffic Report Generated by FloodLight for APs (virtual Switches) of Fig. 10

4.4 Realtime Wireless LAN Monitoring

가상 스위치와 마찬가지로 Avior 창에서 무선 단말들을 모니터링하는 모습은 그림 14와 같이 나타나는데, 각 무선 단말들의 MAC 주소, 연결 스위치 및 포트 정보, 실시간 송·수신 패킷 등을 확인할 수 있다.

#	MAC	IP	Attached Switch	Switch Port
1	7a:b3:64:c0:d1:68	None	None	None
2	00:00:00:00:00:01	10.0.0.1	00:00:00:00:00:04	1
3	00:00:00:00:00:03	10.0.0.3	00:00:00:00:00:05	2
4	a6:88:7c:b1:bc:4a	None	None	None
5	62:b9:2d:ab:c5:3d	None	None	None
6	aa:8b:6c:c0:d9:6a	None	None	None
7	3a:70:78:ae:3c:51	None	None	None
8	00:00:00:00:00:02	10.0.0.2	00:00:00:00:00:04	2

Fig. 14. Monitoring Wireless Devices in Avior Window

그림 15와 그림 16은 구현 가상 스위치 기반 무선랜의 올바른 동작을 검증하기 위해 모바일 디바이스에서의 실시간 무선 패킷 흐름을 확인하는 모습으로, 무선단말 테스트 장비에 리눅스 운영체제와 Wireshark[17]을 설치한 후 무선 패킷을 캡처하기 위해 Aircrack-ng 패키지를 이용하였다. Aircrack-ng는 IEEE 802.11 무선랜에 대해 패킷을 모니터링하거나, WEP, WPA-PSK 암호화 방식의 키를 복호화할 수 있는 패키지이다. Device0이 AP에 연결되어 DHCP로 IP 주소 192.168.3.11를 할당 받는 과정과 ICMP 패킷을 보내 서로의 네트워크 상태를 확인하는 과정은 그림 15와 같이 나타나는데 콘트롤 패킷에 대한 모니터링이 정확하게 이루어짐을 볼 수 있다. 그림 16은 device0(ip 173.194.72.125)에서 device1(ip 192.168.3.11)로 스트리밍 패킷이 전송되는 상황을 정확하게 캡처하는 모습을 보이고 있다.

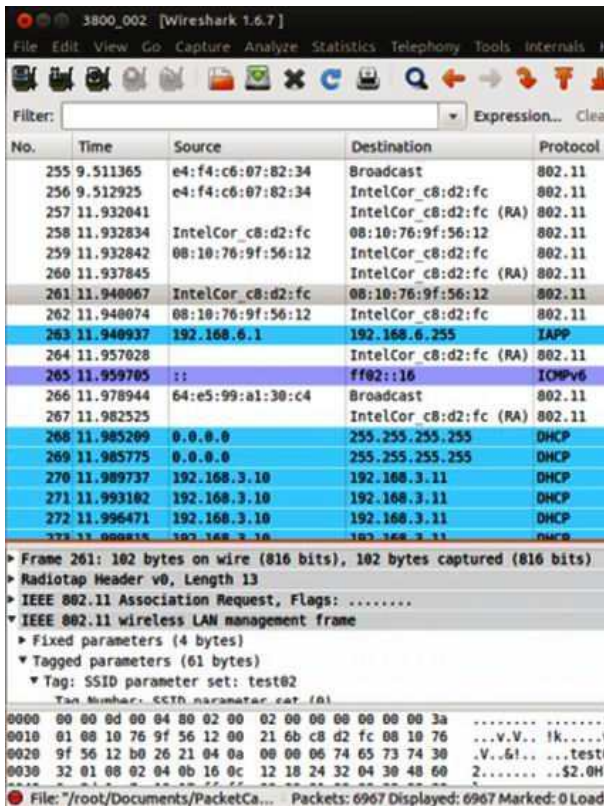


Fig. 15. Capturing DHCP Server Packets Assigning an IP Address to device0 in Wireshark

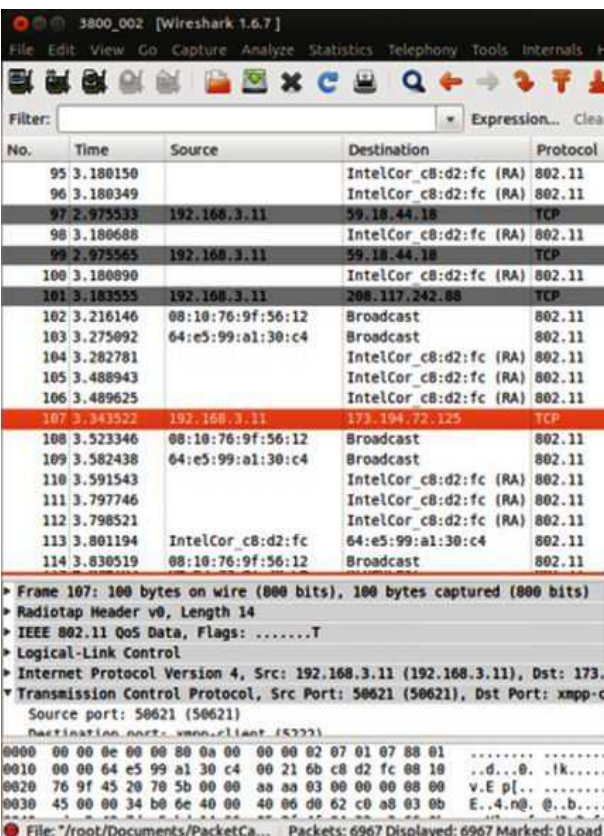


Fig. 16. Capturing Streaming Packets Between device0 and device1 in Wireshark

## IV. Conclusions

네트워크 가상화는 물리적 네트워크 인프라 위에 여러 개의 독립된 논리적 네트워크를 관리하는 개념으로 외부 환경 변화나 내부 고장 등에 신속하게 대응할 수 있다. 이러한 가상화 기법은 네트워크를 구성하는 모든 구성 요소에 대한 완벽한 모니터링을 전제로 이루어진다. 이 논문에서는 데이터 센터 등 유선 랜 환경의 가상화를 목표로 구현된 오픈 소스 스택인 OpenFlow를 기존의 AP 기반 무선랜 환경에 현지화시키고, 실험을 통하여 성공적인 기능을 검증하였으며, 그 결과로부터 구현된 AP용 가상 스위치가 지금 즉시 무선 AP 모니터링에 활용될 수 있음을 확인하였다. 저비용으로 제작 가능한 AP 용 OpenFlow 가상 스위치는 AP의 종류와 무관하게 AP 전단에 위치하여 동작하므로, 학교나 병원 등 기존의 AP 기반 무선랜 환경에 Plug & Play 개념으로 쉽게 적용될 수 있고, 그 모니터링 결과는 AP의 효율적인 운용 정책 수립에 매우 유용한 데이터로 활용될 수 있다.

이번에 구현된 무선네트워크 모니터링 체계 위에 무선랜 용 가상화 프로비전(provision) 어플리케이션을 개발하여 OpenFlow 컨트롤러에 탑재하면 AP 기반 무선랜에 대한 가상화 기능을 갖추게 된다. 향후 연구에서는 기존 무선 네트워크 시스템과 본 논문에서 구축된 OpenFlow 기반 무선 네트워크 시스템 간의 성능 비교 실험을 통하여 객관적인 기능적·시간적 성능 평가 결과를 도출하고 차별화 포인트를 개발함으로써 시장성을 확보할 계획이다.

## REFERENCES

- [1] W. Choi, "The Performance Characteristics of 802.11 Wireless LAN," The Journal of The Korean Institute of Communication Sciences, Vol. 24, No. 6, pp. 50-57, 2007.
- [2] LWAPP, <http://tools.ietf.org/html/rfc5412>
- [3] CAPWAP, <http://tools.ietf.org/html/rfc5415>
- [4] J. Yoo, W. Kim, C. Youn, "A Technical Trend and Prospect of Software Defined Network and OpenFlow," KNOM Review, Vol. 15, No. 2, pp. 1-24, December 2012.
- [5] N. Kabir, R. Boutaba, "Network Virtualization: State of the Art and Research Challenges," IEEE Communications Magazine, Vol. 47, Issue 7, pp. 20-26, July 2009.
- [6] ONF, "Software-Defined Networking: The New Norm for Networks," ONF(Open Networking Foundation) White Paper, pp. 1-12, April 2012. (<https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-newnorm.pdf>).

- [7] J. Pettit, J. Gross, B. Pfaff, and M. Casado, "Virtual Switching in an Era of Advanced Edges," Proc. of the 2nd Workshop on Data Center-Converged and Virtual Ethernet Switching (DC CAVES), pp. 1-7, September 2010.
- [8] ONF, "OpenFlow Switch Specification, ver 1.5.0", ONF(Open Networking Foundation), December 2014 (<https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.0.pdf>).
- [9] S. Lee, S. Chung, "Performance Improvement via Flow-based Routing on OpenFlow-based Wireless Mesh Networks," Journal of KIISE: Information Networking, Vol. 40, No. 4, pp. 187-199, August 2013.
- [10] P. Dely, A. Kassler, and N. Bayer, "OpenFlow for Wireless Networks," Proc. of IEEE International Conference on Computer Communications and Networks, pp. 1-6, 2011.
- [11] W. Kim, S. Chung, H. Choi, and M. Do, "Contents Routing in the OpenFlow-based Wireless Mesh Network Environment," Journal of KIISE: Information Networking, Vol. 41, No. 10, pp. 810-823, October 2014.
- [12] A. Balachandran, G. Voelker, P. Bahl, and P. Rangan, "Characterizing user behavior and network performance in a public wireless LAN," Proc. of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, pp. Vol. 30, Issue 1, pp. 195-205, June 2002.
- [13] H. Kim, N. Choi, S. Baek, Y. Choi, G. Lee, S. Kim, and H. Jung, "Design and Implementation for Measurement System to Analyze Wireless LAN Traffic," Proceedings of The Korean Institute of Communication Sciences Conference, pp. 4-7, November 2004.
- [14] H. Kim, G. Lee, "Implementation and Analysis of Wireless AP Management Scheme," Proc. of KIISE Symposium, Vol. 2, No. 1, pp. 53-59, June 2008.
- [15] MikroTik, <http://www.mikrotik.com>
- [16] Avior, <http://openflow.marist.edu/avior>
- [17] Wireshark, <https://www.wireshark.org>

## Authors



Hyung-Bong Lee was born in Korea on Oct. 22, 1961. He received the B.S. and M.S. degrees in computer science from Seoul National University, Seoul, Korea, in 1984 and 1986 respectively.

He received his Ph.D. degree in computer science from Kangwon National University, Chuncheon, Korea, in 2002. From 1986 to 1994, he was a senior engineer in Computer R&D Division of LG Electronics, and developed a UNIX-based parallel operating system for a symmetric multi-processor machine sponsored by Korea government. From 1995 to 1998, he was with DEC(Digital Equipment Corporation) as a UNIX consultant. From 1999 to 2003, he was an Associate Professor at Honam University, Gwangju, Korea. Since 2004, he has been a Professor in the Department of Computer Science & Engineering at Gangneung-Wonju National University, Wonju, Korea. His current research interests include embedded systems, wireless sensor networks, and data mining algorithms.



Ji-Hoon Park received the M.S. and Ph.D. degrees in Computer Science from Kangwon National University, Korea, in 2003 and 2008 respectively. Dr. Park joined the R&D department of

eLC(e-LibraryCenter), Chuncheon, Korea, in 2011. He is currently a director of R&D department, eLC. He is interested in Software Define Networking, Embedded and cloud computing.